Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

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

Applications of Embedded - Microprocessors

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

Details	
Product Status	Obsolete
Core Processor	M68000
Number of Cores/Bus Width	1 Core, 8/16-Bit
Speed	20MHz
Co-Processors/DSP	Communications; RISC CPM
RAM Controllers	DRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	-
SATA	-
USB	-
Voltage - I/O	5.0V
Operating Temperature	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/mc68302pv20c

DNS—Done Not Synchronized

This bit is set if operand packing is performed between 16-bit memory and an 8-bit peripheral and the DONE signal is asserted as an input to the IDMA (i.e., by the peripheral) during the first access of the 8-bit peripheral. In such a case, the IDMA will still attempt to finish the second access of the 8-bit peripheral even though DONE has been asserted (the access could be blocked with external logic); however, the DNS bit will be set to signify this condition. DNS will not be set if the transfer is terminated by an odd byte count, since, in this case, the exact number of requested bytes will be transferred by the IDMA.

BES—Bus Error Source

This bit indicates that the IDMA channel terminated with an error returned during the read cycle. The channel terminates the IDMA operation without setting DONE. BES is cleared by writing a one or by setting RST in the CMR. Writing a zero has no effect on BES.

BED—Bus Error Destination

This bit indicates that the IDMA channel terminated with an error during the write cycle. The channel terminates the IDMA operation without setting DONE. BED is cleared by writing a one or by setting RST in the CMR. Writing a zero has no effect on BED.

DONE—Normal Channel Transfer Done

This bit indicates that the IDMA channel has terminated normally. Normal channel termination is defined as 1) having decremented the BCR to zero with no errors occurring during any IDMA transfer bus cycle or 2) by the external peripheral asserting DONE with no errors occurring during any IDMA transfer bus cycle. DONE will not be set if the channel terminates due to an error. DONE is cleared by writing a one or by a software RST in the CMR. Writing a zero has no effect on this bit.

3.1.3 Interface Signals

The IDMA channel has three dedicated control signals: DMA request (DREQ), DMA acknowledge (DACK), and end of IDMA transfer (DONE). The IDMA's use of the bus arbitration signals is described in 3.1.6 DMA Bus Arbitration. The peripheral used with these signals may be either a source or a destination of the transfers.

3.1.3.1 DREQ and DACK

These are handshake signals between the peripheral requiring service and the IMP. When the peripheral requires IDMA service, it asserts DREQ, and the IMP begins the IDMA process. When the IDMA service is in progress, DACK is asserted during accesses to the device. These signals are not used when the IDMA is programmed to internal request modes.

3.1.3.2 DONE

This bidirectional signal is used to indicate the last IDMA transfer. With internal request modes, the IDMA activates DONE as an output during the last IDMA bus cycle. If DONE is externally asserted during internal request modes, the IDMA transfer is terminated. With external request modes, DONE may be used as an input to the IDMA controller indicating that the device being serviced requires no more transfers and that the transmission is to be terminated. DONE is an output if the transfer count is exhausted.

$\overline{\text{DTACK}}$ generation occurs under the same constraints as the chip-select signal—if the chip-select signal does not activate, then neither will the $\overline{\text{DTACK}}$ signal.

Chip select 0 has the special property of being enabled upon system reset to the address range from 0 to 8K bytes. This property allows chip select 0 to function as the “boot ROM” select on system start-up. $\overline{\text{DTACK}}$ is initially enabled for six wait states on this chip select.

External masters may use the chip-select logic on the IMP during an external master access to external memory/peripherals. In this case, the external master chip-select timing diagram (see Figure 6-15) must be used. Since the chip-select logic is slightly slower when using external masters, an optional provision can be made to add an additional wait state to an external access by an external master. See the EMWS bit in the SCR for more details (3.8.3 System Control Bits).

A priority structure exists within the chip-select block. For a given address, the priority is as follows:

1. Access to any IMP internal address (BAR, dual-port RAM, etc.)
   No chip select asserted.
2. Chip Select 0
3. Chip Select 1
4. Chip Select 2
5. Chip Select 3

1. Calculate what the mask should be. For a 1 Megabyte block, the address lines A0 through A19 are used to address bytes within the block, so they need to be masked out.
2. Write $3E00 to OR2 (DTACK=1 for 1 wait state, M23-M20 = 1 to use these bits in the comparison, M19-M13 = 0 to mask these address bits, MRW = 0 to enable the chip select for both read and write, and CFC = 0 to mask off function code comparison).
3. Write $0401 to BR2 (FC2-FC0 = 0 don't care, A23-A13 = base address, RW = 0 don't care, and EN =1 to enable the chip select.

NOTE

The mask bits in the OR are used to mask the individual address bits, so in the previous example, if bit 12 (M23) was changed to a zero, then CS2 would assert for a 1 Megabyte block beginning at $200000 and a 1 Megabyte block at $A00000.

3.7 ON-CHIP CLOCK GENERATOR

The IMP has an on-chip clock generator that supplies clocks to both the internal M68000 core and peripherals and to an external pin. The clock circuitry uses three dedicated pins: EXTAL, XTAL, and CLKO.

The external clock/crystal (EXTAL) input provides two clock generation options. EXTAL may be used to interface the internal generator to an external crystal (see Figure 3-10). Typical circuit parameters are C1 = C2 = 25 pF and R = 700 kΩ using a parallel resonant crystal. Typical crystal parameters are Co < 10 pF and Rx = 50 Ω. The equivalent load capacitance (C_L) of this circuit is 20 pF, calculated as (C1 + C_{in})/2, where C1 = C2 = 25 pF and C_{in} = 15 pF maximum on the EXTAL pin.

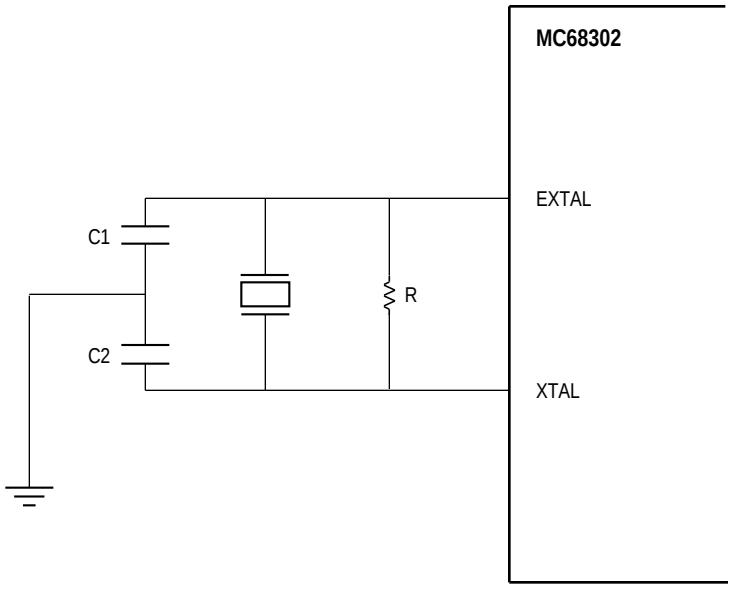

Figure 3-10. Using an External Crystal

2. $\overline{BG}$ will be an input to the IDMA and SDMA from the external M68000 bus, rather than being an output from the MC68302. When BG is sampled as low by the MC68302, it waits for $\overline{AS}$, $\overline{BERR}$, $\overline{HALT}$, and $\overline{BGACK}$ to be negated, and then asserts $\overline{BGACK}$ and performs one or more bus cycles. See Section 6 for timing diagrams.
3. $\overline{BCLR}$ will be an input to the IDMA, but will remain an output from the SDMA.
4. The interrupt controller will output its interrupt request lines ($\overline{IPL0}$, $\overline{IPL1}$, $\overline{IPL2}$) normally sent to the M68000 core on pins $\overline{IOUT0}$, $\overline{IOUT1}$, and $\overline{IOUT2}$, respectively. $\overline{AVEC}$, $\overline{RMC}$, and $\overline{CS0}$, which share pins with $\overline{IOUT0}$, $\overline{IOUT1}$, and $\overline{IOUT2}$, respectively, are not available in this mode.

DISCPU should remain continuously high during disable CPU mode operation. Although the $\overline{CS0}$ pin is not available as an output from the device in disable CPU mode, it may be enabled to provide $\overline{DTACK}$ generation. In disable CPU mode, BR0 is initially $C000.

Accesses by an external master to the MC68302 RAM and registers may be asynchronous or synchronous to the MC68302 clock. (This feature is actually available regardless of disable CPU mode). See the SAM and EMWS bits in the SCR for details.

In disable CPU mode, the interrupt controller may be programmed to generate or not generate interrupt vectors during interrupt acknowledge cycles. When multiple MC68302 devices share a single M68000 bus, vector generation at level 4 should be prevented on all but one MC68302. When using disable CPU mode to implement an interface, such as between the MC68020 and a single MC68302, vector generation can be enabled. For this purpose, the VGE bit is defined.

VGE—Vector Generation Enable

0 = In disable CPU mode, the MC68302 will not output interrupt vectors during interrupt acknowledge cycles.
1 = In disable CPU mode, the MC68302 will output interrupt vectors for internal level 4 interrupts (and for levels 1, 6, and/or 7 as enabled in the interrupt controller) during interrupt acknowledge cycles.

NOTE

Do not use the function code value “111” during external accesses to the IMP, except during interrupt acknowledge cycles.

In disable CPU mode, the low-power modes will be entered immediately upon the setting of the LPEN bit in the SCR by an external master. In this case, low-power mode will continue until the LPEN bit is cleared. Users may wish to use a low-power mode in conjunction with disable CPU mode to save power consumed by the disabled M68000 core.

All MC68302 functionality not expressly mentioned in this section is retained in disable CPU mode and operates identically as before.

NOTE

Even without the use of the disable CPU logic, another processor can be granted access to the IMP on-chip peripherals by re-

pendix A SCC Performance). Also, the minimum 1:2.5 serial to CLKO clock ratio must be maintained at all times.

The following list gives a step-by-step example of how to achieve the lowest possible power using an external clock. For this example, an external wakeup signal is issued to the PB11 pin to exit the lowest power mode.

1. Set the lower byte of the SCR (location $F7) to $A0. This sets the LPREC bit and the LPEN bits only.
2. Disable all interrupts except PB11 in the IMR.
3. Turn off any unneeded peripherals, such as the SCCs, by clearing the ENR and ENT bits. Also, turn off any unneeded baud rate generators by setting the EXTC bits in the SCON registers. This procedure can save as much as 4 mA per SCC at 16.67 MHz. (EXTC is cleared by default on after reset.)
4. Start off a timer now to toggle a $\overline{\text{TOUT}}$ pin in approximately 20 clocks. Do not wait for this to occur, but continue on to the next step.
5. Execute the STOP instruction. The IMP is now safely in the lowest power mode.
6. Use the toggled $\overline{\text{TOUT}}$ pin to switch the EXTAL clock rate to approximately 50 kHz. Ensure no glitches occur on the EXTAL signal which exceed the maximum clock frequency.
7. Power consumption is now the lowest.
8. A wakeup signal comes from the system.
9. The wakeup signal switches the clock frequency back to the 8–16.67-MHz range and pulls the PB11 pin low. These two events can happen simultaneously.
10. The IMP generates the PB11 interrupt, and a M68000 core reset is generated.
11. After the IMP is reset, software processing continues from the exception vector table reset vector address. The M68000 is reset, but the rest of the IMP retains its state.

The low-power logic uses eight bits in the SCR.

LPCD4–LPCD0—Low-power Clock Divider Selects

The low-power clock divider select bits (LPCD4—LPCD0) specify the divide ratio of the low-power clock divider equal to LPCD4—LPCD0 + 1. The system clock is divided by 2, then divided by the clock divider value (1 to 32). Thus, a divide ratio of 2 to 64 (LPCD4—LPCD0 0 to 31) can be selected. After a system reset, these bits default to zero.

LPEN—Low-power Enable

0 = The low-power modes are disabled.
1 = The low-power modes are enabled.

After a system reset, this bit defaults to zero to disable the low-power modes.

counter (NOSEC).

6. IDLE Sequence. Receive IDLE (preamble) is detected by the UART controller when a character with 9 to 13 consecutive ones (depending on the UM1–UM0, SL, PEN, and CL bits in the UART mode register) is received. When an IDLE sequence is received, the channel starts to count the number of IDLE sequences received. If it reaches the MAX_IDL value, the buffer is closed and an RX interrupt is generated (if enabled). The counter is reset every time a character is received.
7. BREAK Sequence. A BREAK sequence is detected by the UART receiver when a character with zero value and framing error is received. When a BREAK sequence is received, the channel will increment the BRKEC counter, close the buffer, set the BR bit (if a buffer was currently open), and generate a BRK interrupt (if enabled). Also, if the channel was in the middle of buffer processing, the buffer is closed and an RX is generated (if enabled). A long break sequence only increments the counter once.

Error Counters

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

PAREC—Parity Error Counter
FRMEC—Framing Error Counter
NOSEC—Noise Error Counter
BRKEC—BREAK Error Counter

4.5.11.12 Fractional Stop Bits

The UART transmitter can be programmed to transmit fractional stop bits. Three bits in the SCC data synchronization register (DSR) are used to program the length of the last stop bit transmitted. These DSR bits may be modified at any time. If two stop bits are transmitted, only the second one is affected. Idle characters are always transmitted as full-length characters. In UART mode, bits 14–12 in the DSR are now decoded as follows:

14–12 of DSR			
111	Last Transmit Stop Bit	16/16	(the default value after reset)
110	Last Transmit Stop Bit	15/16	
. . .			
001	Last Transmit Stop Bit	10/16	
000	Last Transmit Stop Bit	9/16	

The setting of the DSR in combination with the setting of the CL bit in the UART mode register causes the number of stop bits transmitted to be either 9/16 to 1 or 1-9/16 to 2 stop bits.

The UART receiver can always receive fractional stop bits. The next character's start bit may begin anytime after the 11th internal clock of the previous character's first stop bit (the UART uses a 16x clock).

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 8- or 16-bit control field provides a flow control number and defines the frame type (control or data). The exact use and structure of this field depends upon the protocol using the frame.

Data is transmitted in the data field, which can vary in length depending upon the protocol using the frame. Layer 3 frames are carried in the data field.

Error control is implemented by appending a cyclic redundancy check (CRC) to the frame, which is 16-bits long in most protocols, but may be 32-bits long in some.

When the MODE1–MODE0 bits of an SCC mode register (SCM) select the HDLC mode, then that SCC functions as an HDLC controller. The HDLC controller handles the basic functions of the HDLC/SDLC protocol on either the D channel, a B channel, or from a multiplexed serial interface (IDL, GCI (IOM-2), or PCM highway). When the HDLC controller is used to support the B or D channel of the ISDN, the SCC outputs are internally connected to the physical layer serial interface.

NOTE

SDLC is fully supported, but the SDLC loop mode (ring configuration) is not supported.

When an SCC in HDLC mode is used with a nonmultiplexed modem interface, then the SCC outputs are connected directly to the external pins. In this case, the serial interface uses seven dedicated pins: transmit data (TXD), receive data (RXD), receive clock (RCLK), transmit clock (TCLK), carrier detect ($\overline{CD}$), clear to send ($\overline{CTS}$), and request to send ($\overline{RTS}$). Other modem signals may be supported through the parallel I/O pins.

The HDLC controller consists of separate transmit and receive sections whose operations are asynchronous with the M68000 core and may be either synchronous or asynchronous with respect to the other SCCs. Up to eight frames may be transmitted or received without M68000 core intervention. When the HDLC controller is connected to one of the multiplexed physical interface options (IDL, GCI, or PCM highway), the receive and transmit clocks are identical and are supplied externally by the physical layer. In non-ISDN applications, each clock can be supplied either from the baud rate generator or externally. The baud rate generator is discussed more fully in 4.5.2 SCC Configuration Register (SCON).

The HDLC controller key features are as follows:

- Flexible Data Buffers with Multiple Buffers per Frame Allowed
- Separate Interrupts for Frames and Buffers (Receive and Transmit)
- Four Address Comparison Registers with Mask
- Maintenance of Five 16-Bit Error Counters
- Flag/Abort/Idle Generation/Detection
- Zero Insertion/Deletion
- NRZ/NRZI Data Encoding
- 16-Bit or 32-Bit CRC-CCITT Generation/Checking

NON TRANSPARENT WITH HEADER

SYN1	SYN2	SOH	HEADER	STX	TEXT	ETX	BCC

NON TRANSPARENT WITHOUT HEADER

SYN1	SYN2	STX	TEXT	ETX	BCC

TRANSPARENT

SYN1	SYN2	DLE	STX	TRANSPARENT TEXT	DLE	ETX	BCC

Figure 4-30. Typical BISYNC Frames

The bulk of the frame is divided into fields whose meaning depends on the frame type. The BCC is either a 16-bit CRC (CRC-16) format if 8-bit characters are used or a longitudinal check (a sum check) in combination with vertical redundancy check (parity) if 7-bit characters are used. In transparent operation, to allow the BISYNC control characters to be present in the frame as valid text data, a special character (DLE) is defined, which informs the receiver that the character following the DLE is a text character, not a control character (from the control character table). If a DLE is transmitted as valid data, it must be preceded by a DLE character. This procedure is sometimes called byte-stuffing.

The physical layer of the BISYNC communications link must provide a means of synchronizing the receiver and transmitter, which is usually accomplished by sending at least one pair of synchronization characters prior to every frame.

BISYNC has the unusual property that a transmit underrun need not be an error. If an underrun occurs, the synchronization pattern is transmitted until data is once again ready to transmit. The receiver discards the additional synchronization characters as they are received, provided the V bit is set in the BISYNC-BISYNC SYNC register. In non-transparent operation, all synchronization characters (SYNCs) are discarded if the V bit is set in the BISYNC-BISYNC SYNC register. In transparent operation, all DLE-SYNC pairs are discarded. (Note that correct operation in this case assumes that, on the transmit side, the underrun does not occur between the DLE and its following character, a failure mode prevented in the MC68302.)

By appropriately setting the SCC mode register, any of the SCC channels may be configured to function as a BISYNC controller. The BISYNC controller handles the basic functions of the BISYNC protocol in normal mode and in transparent mode.

The SCC in BISYNC mode can work with IDL, GCI (IOM2), PCM highway, or NMSI interfaces. When the SCC in BISYNC mode is used with a modem interface (NMSI), the SCC outputs are connected directly to the external pins. The modem interface uses seven dedicated pins: transmit data (TXD), receive data (RXD) receive clock (RCLK), transmit clock (TCLK), carrier detect ($\overline{CD}$), clear to send ($\overline{CTS}$), and request to send ($\overline{RTS}$). Other modem lines can be supported using the parallel I/O pins.

The BISYNC controller consists of separate transmit and receive sections whose operations are asynchronous with the M68000 core and may be either synchronous or asynchronous with respect to the other SCCs. Each clock can be supplied from either the internal baud

R—Ready

0 = This buffer is not currently ready for transmission. The user is free to manipulate this BD (or its associated buffer). The CP clears this bit after the buffer has been fully transmitted or after an error condition has been encountered.
1 = The data buffer has been prepared for transmission by the user (but not yet transmitted). No fields of this BD may be written by the user once this bit is set.

X—External Buffer

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

W—Wrap (Final BD in Table)

0 = This is not the last BD in the Tx BD table.
1 = This is the last BD in the Tx BD table. After this buffer has been used, the CP will transmit data from the first BD in the table.

NOTE

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

I—Interrupt

0 = No interrupt is generated after this buffer has been serviced.
1 = When this buffer is serviced by the CP, the TX or TXE bit in the transparent event register will be set, which can cause an interrupt.

L—Last in Message

0 = The last byte in the buffer is not the last byte in the transmitted block. Data from the next transmit buffer (if ready) will be transmitted immediately following the last byte of this buffer.
1 = The last byte in the buffer is the last byte in the transmitted block. After this buffer is transmitted, the transmitter will require synchronization before the next buffer can be transmitted.

Bits 10—2 are reserved for future use; they should be written with zero.

The following status bits are written by the CP after it has finished transmitting the associated data buffer.

UN—Underrun

The transparent controller encountered a transmitter underrun condition while transmitting the associated data buffer.

CT—CTS Lost

CTS in NMSI mode or L1GR in IDL/GCI mode was lost during frame transmission.

Data Length

The data length is the number of octets that the CP should transmit from this BD's data buffer. The data length, which should be greater than zero, may be even or odd. This value is never modified by the CP.

Tx Buffer Pointer

The transmit buffer pointer, which always points to the first byte 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.

4.5.16.10 Transparent Event Register

The SCC event register (SCCE) is referred to as the transparent event register when the SCC is programmed as a transparent controller. It is an 8-bit register used to report events recognized by the transparent channel and to generate interrupts. On recognition of an event, the transparent controller sets the corresponding bit in the transparent event register. Interrupts generated by this register may be masked in the transparent mask register.

The transparent event register is a memory-mapped register that may be read at any time. A bit is cleared by writing a one (writing a zero does not affect a bit's value). More than one bit may be cleared at a time. All unmasked bits must be cleared before the CP will negate the internal interrupt request signal. This register is cleared at reset.

7	6	5	4	3	2	1	0
CTS	CD	—	TXE	RCH	BSY	TX	RX

CTS—Clear-To-Send Status Changed

A change in the status of the serial line was detected on the transparent channel. The SCC status register may be read to determine the current status.

CD—Carrier Detect Status Changed

A change in the status of the serial line was detected on the transparent channel. The SCC status register may be read to determine the current status.

Bit 5—Reserved for future use.

TXE—Tx Error

An error (CTS lost or underrun) occurred on the transmitter channel.

RCH—Receive Character

A word has been received and written to the receive buffer.

BSY—Busy Condition

A word was received and discarded due to lack of buffers. The receiver will resume reception after an ENTER HUNT MODE command.

triggers. TCLK2 acts as the SCC2 baud rate generator output if SCC2 is in one of the multiplexed modes.

- RXD2/PA0
- TXD2/PA1
- RCLK2/PA2
- TCLK2/PA3
- $\overline{\text{CTS2}}$/PA4
- $\overline{\text{RTS2}}$/PA5
- $\overline{\text{CD2}}$/PA6
- SDS2/PA7/BRG2

Table 5-9. Baud Rate Generator Outputs

Source	NMSI	GCI	IDL	PCM
SCC1	BRG1	BRG1	BRG1	BRG1
SCC2	BRG2	TCLK2	TCLK2	TCLK2
SCC3	BRG3	TCLK3	TCLK3	TCLK3

NOTE: In NMSI mode, the baud rate generator outputs can also appear on the RCLK and TCLK pins as programmed in the SCON register.

5.15 NMSI3 PORT OR PORT A PINS OR SCP PINS

The NMSI3 port or port A pins or SCP pins are shown in Figure 5-12.

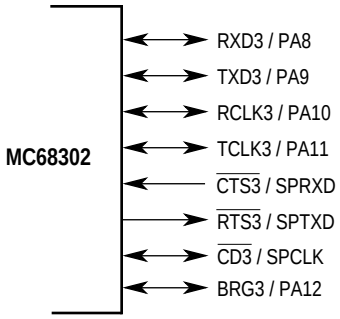

Figure 5-12. NMSI3 Port or Port A Pins or SCP Pins

These eight pins can be used either as the NMSI3 port or as the NMSI3 port (less three modem lines) and the SCP port. If the SCP is enabled (EN bit in SPMODE register is set), then the three lines are connected to the SCP port. Otherwise, they are connected to the SCC3 port.

Each of the port A I/O pins can be configured individually to be general-purpose I/O pins or a dedicated function in NMSI3. When they are used as the NMSI3 pins, they function exactly

6.18 AC ELECTRICAL SPECIFICATIONS—SERIAL COMMUNICATIONS PORT (see Figure 6-19).

		16.67 MHz		20 MHz		25 MHz		
Num.	Characteristic	Min	Max	Min	Max	Min	Max	Unit
250	SPCLK Clock Output Period	4	64	4	64	4	64	clks
251	SPCLK Clock Output Rise/Fall Time	0	15	0	10	0	8	ns
252	Delay from SPCLK to Transmit (see Note 1)	0	40	0	30	0	24	ns
253	SCP Receive Setup Time (see Note 1)	40	—	30	—	24	—	ns
254	SCP Receive Hold Time (see Note 1)	10	—	8	—	7	—	ns

NOTES:
1. This also applies when SPCLK is inverted by CI in the SPMODE register.
2. The enable signals for the slaves may be implemented by the parallel I/O pins.

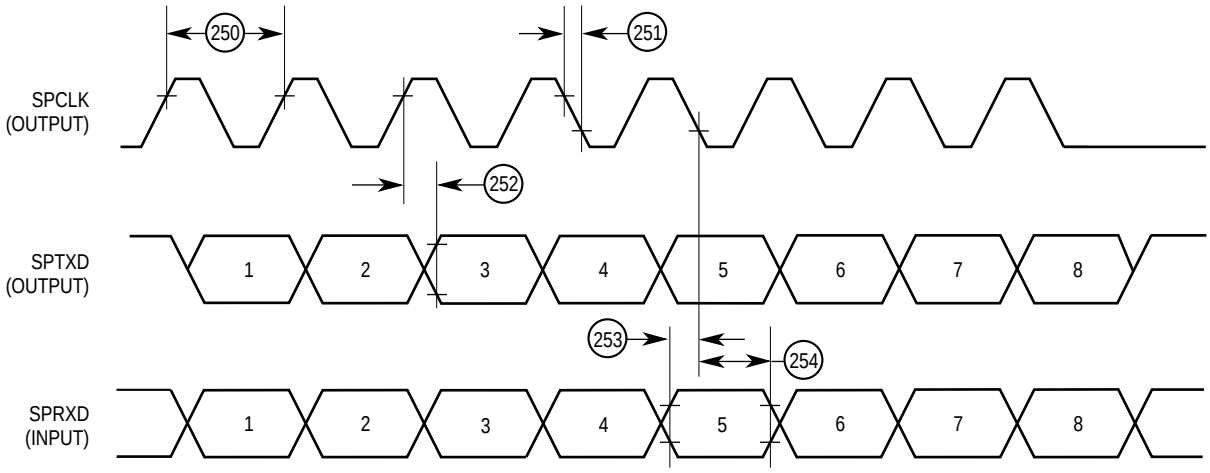

Figure 6-19. Serial Communication Port Timing Diagram

APPENDIX C
RISC MICROCODE FROM RAM

The MC68302 RISC processor has an option to execute microcode from the 576-byte user RAM in the on-chip dual-port RAM. In this mode, the 576-byte user RAM cannot be accessed by the M68000 core or other M68000 bus masters. Also in this mode, port A pins are optionally available to the RISC processor as well as the M68000 core. Figure C-1 shows these resulting changes.

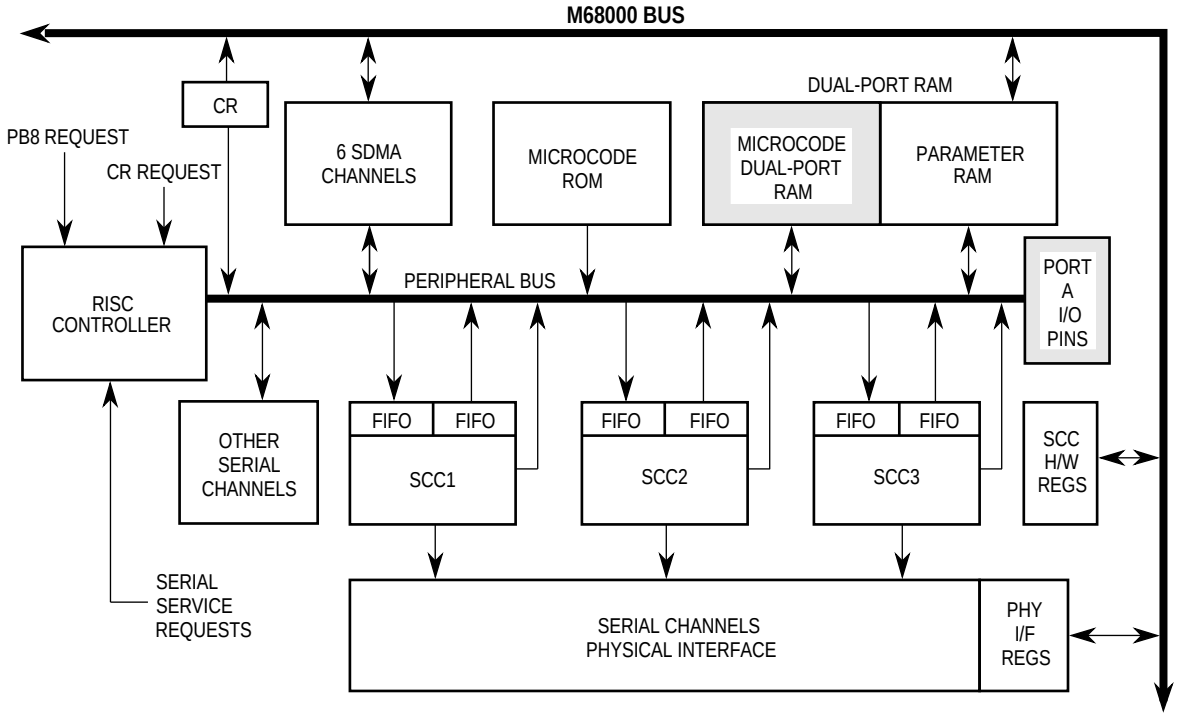

Figure C-1. CP Architecture Running RAM Microcode

Once the microcode has been loaded into the dual-port RAM by the M68000 core or other bus master, the microcode from RAM option is enabled in the reserved register at location $0F8. When the user writes $0001 to location $0F8, the RISC processor will execute microcode from RAM once the CP is reset in the command register. Hereafter, the RISC processor can freely address both the dual-port RAM and its own private ROM.

APPENDIX D
MC68302 APPLICATIONS

This appendix describes different applications for the MC68302.

D.1 MINIMUM SYSTEM CONFIGURATION

The following paragraphs describe a minimum 16-bit MC68302 system. As Figure D-1 shows, this system can be easily built with very few components.

D.1.1 System Configuration

The crystal circuit shown in Figure D-1 is a typical configuration. The values used are not required by Motorola. Some deviation of the capacitance or resistance values is allowed. Crystal parameters need not be anything special—Co < 10 pF and Rx = 50 Ω is used. Of course, an oscillator could be used in place of the crystal circuit.

$\overline{\text{AVEC}}$ is pulled high since autovectoring for external interrupts is not needed. If external devices were added (not shown) the MC68302 interrupt controller could handle the interrupt vector generation for up to seven external sources using $\overline{\text{IRQ7}}$, $\overline{\text{IRQ6}}$, $\overline{\text{1RQ1}}$, PB11, PB10, PB9, and PB8. The $\overline{\text{IPL2}}$-$\overline{\text{IPL0}}$ lines are also pulled high (inactive) since no external interrupts are required.

BUSW is pulled high for 16-bit operation and may not be modified dynamically. Choice of 8-bit operation could be used to eliminate two of the memory chips.

$\overline{\text{BERR}}$ is pulled high since it is an open-drain signal. It will be asserted low by the MC68302 if the hardware watchdog terminates a stalled bus cycle.

$\overline{\text{BR}}$ is tied high since no external bus masters exist in this design.

$\overline{\text{BGACK}}$ is pulled high (inactive). It is asserted low during an IDMA or SDMA bus cycle.

$\overline{\text{FRZ}}$ is tied high since the MC68302 freeze debugging logic is not used in this design.

DISCPU is tied low to allow the M68000 core to function normally. Tying this pin high causes the part to enter the disable CPU mode when the core is disabled and the part is an intelligent peripheral.

All unused lines should be terminated.

porarily halt any IDMA accesses. The SDMAs only use the M68000 bus for a single bus cycle before giving up the bus, so priority between SDMAs is not an issue. In this system, if an SDMA channel from the slave requests service at the same time as an SDMA channel from the master, the slave SDMA will go first.

With multiple slaves (i.e., multiple external bus masters from the standpoint of the master MC68302), external logic must prioritize the various $\overline{BR}$ signals. The $\overline{BGACK}$ and $\overline{BCLR}$ signals can be connected as shown in Figure D-20, but the $\overline{BR}$ and $\overline{BG}$ signals must be routed to the external bus arbiter.

D.7.6 Final Notes

It is important for the slave to negate its $\overline{BR}$ pin quickly after it asserts $\overline{BGACK}$ to meet the master's bus arbitration timing specifications. Thus, a 820Ω pullup resistor is used in Figure D-20.

Will the slave SDMA channels move SCC data to/from the low half of memory or the high half of memory? If the decision is the low half of memory, the following notes about A23 should be considered.

Since A23 on the master and the slave are not driven by the address bus, they must be at the proper level before beginning accesses to and from the slave MC68302. Thus, the slave's A23 pullup may have to be reduced from 10k to 1k to give a fast enough rise time (for instance, a slave SDMA access is immediately followed by a master access to the slave).

The decision of whether to pull the master's A23 pin up or down is made based on whether the SDMA on the slave will be storing its SCC data into the high half or the low half of memory, respectively. A 1k resistor may be required in this case as well. However, if the master MC68302 chip-select logic is used by the slave, the chip-select comparison for the A23 pin can be disabled. (If this trick is used, it is important that no peripherals or memory be mapped to the chip select's corresponding area in the upper half of system memory).

D.8 USING THE MC68302 TRANSPARENT MODE

The following paragraphs describe different ways that the totally transparent (promiscuous) mode on the MC68302 serial channels can be used.

D.8.1 Transparent Mode Definition

Transparent mode allows the transmission and reception of serial data over a serial communications controller (SCC) without any modification to that data stream. Contrast this with HDLC mode, where zero bits are occasionally inserted during transmission and stripped during reception, and with UART mode, where start and stop bits are inserted and stripped. Transparent mode provides a clear channel on which no bit-level manipulation is performed by the SCC. Any protocol run on transparent mode is performed in software. The job of an SCC in transparent mode is to function simply as a high-speed serial-parallel converter.

Transparent mode on the MC68302 also means a synchronous protocol; thus, a clock edge must be provided with each bit of data received or transmitted. Contrast this with the UART

D.9.1 Overview of the Board

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

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

D.9.2 Important Side Notes

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

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

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

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

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

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

I—Interrupt

0 = The TXB bit in the event register is not set when this buffer is closed.
1 = The TXB 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 a frame.
1 = This buffer is the last buffer in a frame.

TC—Tx CRC

0 = Transmit the closing flag after the last data byte.
1 = Transmit the CRC sequence after the last data byte.

Bits 9-2—Reserved for future use

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.1.1.5.2 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 butter.

E.1.1.5.3 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.1.2 Programming the SCC for HDLC

This section gives a generic algorithm for programming an SCC to handle HDLC. 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.1.2.1 CP INITIALIZATION.

1. Write the port A and port B control registers (PACNT and PBCNT) to configure SCC2 or SCC3 serial interface pins as peripheral pins, if SCC2 or SCC3 is used.
2. Write SIMODE to configure the SCCs physical interface.
3. Write SIMASK if IDL or GCI multiplexed mode was selected in SIMODE.

E.1.2.2 GENERAL AND HDLC PROTOCOL-SPECIFIC RAM INITIALIZATION.

4. Write RFCR/TFCR.

Table E-1. (a)Transparent Programming Model
Receive and Transmit Buffer Descriptors for SCCx

Initialized by User	Offset Hex	Name	Initialized by User	Offset Hex	Name
Yes	00	Rx BD 0 Control/Status	Yes	40	Tx BD 0 Control/Status
	02	Rx BD 0 Data Count	Yes	42	Tx BD 0 Data Count
Yes	04	Rx BD 0 Data Pointer (High Word)	Yes	44	Tx BD 0 Data Pointer (High Word)
Yes	06	Rx BD 0 Data Pointer (Low Word)	Yes	46	Tx BD 0 Data Pointer (Low Word)
Yes	08	Rx BD 1 Control/Status	Yes	48	Tx BD 1 Control/Status
	0A	Rx BD 1 Data Count	Yes	4A	Tx BD 1 Data Count
Yes	0C	Rx BD 1 Data Pointer (High Word)	Yes	4C	Tx BD 1 Data Pointer (High Word)
Yes	0E	Rx BD 1 Data Pointer (Low Word)	Yes	4E	Tx BD 1 Data Pointer (Low Word)
Yes	10	Rx BD 2 Control/Status	Yes	50	Tx BD 2 Control/Status
	12	Rx BD 2 Data Count	Yes	52	Tx BD 2 Data Count
Yes	14	Rx BD 2 Data Pointer (High Word)	Yes	54	Tx BD 2 Data Pointer (High Word)
Yes	16	Rx BD 2 Data Pointer (Low Word)	Yes	56	Tx BD 2 Data Pointer (Low Word)
Yes	18	Rx BD 3 Control/Status	Yes	58	Tx BD 3 Control/Status
	1A	Rx BD 3 Data Count	Yes	5A	Tx BD 3 Data Count
Yes	1C	Rx BD 3 Data Pointer (High Word)	Yes	5C	Tx BD 3 Data Pointer (High Word)
Yes	1E	Rx BD 3 Data Pointer (Low Word)	Yes	5E	Tx BD 3 Data Pointer (Low Word)
Yes	20	Rx BD 4 Control/Status	Yes	60	Tx BD 4 Control/Status
	22	Rx BD 4 Data Count	Yes	62	Tx BD 4 Data Count
Yes	24	Rx BD 4 Data Pointer (High Word)	Yes	64	Tx BD 4 Data Pointer (High Word)
Yes	26	Rx BD 4 Data Pointer (Low Word)	Yes	66	Tx BD 4 Data Pointer (Low Word)
Yes	28	Rx BD 5 Control/Status	Yes	68	Tx BD 5 Control/Status
	2A	Rx BD 5 Data Count	Yes	6A	Tx BD 5 Data Count
Yes	2C	Rx BD 5 Data Pointer (High Word)	Yes	6C	Tx BD 5 Data Pointer (High Word)
Yes	2E	Rx BD 5 Data Pointer (Low Word)	Yes	6E	Tx BD 5 Data Pointer (Low Word)
Yes	30	Rx BD 6 Control/Status	Yes	70	Tx BD 6 Control/Status
	32	Rx BD 6 Data Count	Yes	72	Tx BD 6 Data Count
Yes	34	Rx BD 6 Data Pointer (High Word)	Yes	74	Tx BD 6 Data Pointer (High Word)
Yes	36	Rx BD 6 Data Pointer (Low Word)	Yes	76	Tx BD 6 Data Pointer (Low Word)
Yes	38	Rx BD 7 Control/Status	Yes	78	Tx BD 7 Control/Status
	3A	Rx BD 7 Data Count	Yes	7A	Tx BD 7 Data Count
Yes	3C	Rx BD 7 Data Pointer (High Word)	Yes	7C	Tx BD 7 Data Pointer (High Word)
Yes	3E	Rx BD 7 Data Pointer (Low Word)	Yes	7E	Tx BD 7 Data Pointer (Low Word)

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

.

Table E-1 (b). Transparent Programming Model (Continued)
General Parameter and Transparent Protocol-Specific RAM for SCCx

Initialized by User	Offset Hex	Name		Initialized by User	Offset Hex	Name
Yes	80	RFCR	TFCR		A2	Reserved
Yes	82	MRBLR			A4	Reserved
	84	Rx Internal State			A6	Reserved
	86	Reserved	Rx Internal Buffer No.		A8	Reserved
	88 8A	Rx Internal Data Pointer (High Word) Rx Internal Data Pointer (Low Word)			AA	Reserved
	8C	Rx Internal Byte Count			AC	Reserved
	8E	Rx Temp			AE	Reserved
	90	Tx Internal State			B0	Reserved
	92	Reserved	Tx Internal Buffer No.		B2	Reserved
	94 96	Tx Internal Data Pointer (High Word) TX Internal Data Pointer (Low Word)			B4	Reserved
	98	Tx Internal Byte Count			B6	Reserved
	9A	Tx Temp			B8	Reserved
	9C	Reserved			BA	Reserved
	9E	Reserved			BC	Reserved
	A0	Reserved			BE	Reserved

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

E—Empty

0 = This data buffer is full or has been closed due to an error condition.
1 = This data buffer is empty; must be set by the user to enable reception into this 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 receive BD table.
1 = This is the last BD in the receive BD table.

I—Interrupt

0 = No interrupt is generated when this buffer is closed.
1 = The RX bit in the event register is set when this buffer is closed.

Bits 11-2—Reserved for future use; should be written with zero by the user.

OV—Overrun

0 = No receiver overrun occurred.
1 = A receiver overrun condition occurred during frame reception.

CD—Carrier Detect Lost (valid only in NMSI mode)

0 = No CD lost was detected.
1 = CD was negated during frame reception.

E.3.1.4.2 Receive Buffer Data Length. This 16-bit value is written by the IMP to indicate the number of data bytes received into the data buffer.

E.3.1.4.3 Receive Buffer Pointer. This 32 bit value is written by the user to indicate the address where the data is to be stored.

E.3.1.5 TRANSMIT BUFFER DESCRIPTORS. Each SCC has eight transmit 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	R	X	W	I	L	—	—	—	—	—	—	—	—	—	UN	CT
OFFSET +2	DATA LENGTH															
OFFSET +4	TX BUFFER POINTER															
OFFSET +6																

E.3.1.5.1 Transmit BD Control/Status Word. To initialize the buffer, the user should write bits 15-11 and clear bits 1-0. The IMP clears bit 15 when the buffer is transmitted or closed due to an error and sets bits 1-0 depending on which error occurred.