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	25MHz
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	132-BPGA Exposed Pad
Supplier Device Package	132-PGA (34.5x34.5)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68302rc25c

identical. When any SCC, SCP, or SMC channel buffer descriptors or parameters are not used, their parameter RAM area can be used for additional memory. For detailed information about the use of the buffer descriptors and protocol parameters in a specific protocol, see 4.5 Serial Communication Controllers (SCCs). Base + 67E contains the MC68302 revision number. Revision A parts (mask 1B14M) correspond to the value $0001. Revision B parts (mask 2B14M and 3B14M which are described in this manual) correspond to the value $0002. Revision C and D parts have revision number $0003.

Table 2-8. Parameter RAM

Address	Width	Block	Description
Base + 400	4 Word	SCC1	Rx BD 0
Base + 408	4 Word	SCC1	Rx BD 1
Base + 410	4 Word	SCC1	Rx BD 2
Base + 418	4 Word	SCC1	Rx BD 3
Base + 420	4 Word	SCC1	Rx BD 4
Base + 428	4 Word	SCC1	Rx BD 5
Base + 430	4 Word	SCC1	Rx BD 6
Base + 438	4 Word	SCC1	Rx BD 7
Base + 440	4 Word	SCC1	Tx BD 0
Base + 448	4 Word	SCC1	Tx BD 1
Base + 450	4 Word	SCC1	Tx BD 2
Base + 458	4 Word	SCC1	Tx BD 3
Base + 460	4 Word	SCC1	Tx BD 4
Base + 468	4 Word	SCC1	Tx BD 5
Base + 470	4 Word	SCC1	Tx BD 6
Base + 478	4 Word	SCC1	Tx BD 7
Base + 480 • • • Base + 4BF		SCC1 SCC1	Specific Protocol Parameters
Base + 4C0 • • • Base + 4FF			Reserved (Not Implemented)
Base + 500	4 Word	SCC2	Rx BD 0
Base + 508	4 Word	SCC2	Rx BD 1
Base + 510	4 Word	SCC2	Rx BD 2
Base + 518	4 Word	SCC2	Rx BD 3
Base + 520	4 Word	SCC2	Rx BD 4
Base + 528	4 Word	SCC2	Rx BD 5
Base + 530	4 Word	SCC2	Rx BD 6
Base + 538	4 Word	SCC2	Rx BD 7
Base + 540	4 Word	SCC2	Tx BD 0
Base + 548	4 Word	SCC2	Tx BD 1
Base + 550	4 Word	SCC2	Tx BD 2
Base + 558	4 Word	SCC2	Tx BD 3
Base + 560	4 Word	SCC2	Tx BD 4
Base + 568	4 Word	SCC2	Tx BD 5
Base + 570	4 Word	SCC2	Tx BD 6/DRAM Refresh
Base + 578	4 Word	SCC2	Tx BD 7/DRAM Refresh

External Device Termination

If desired, a transfer may be terminated by the device even before the BCR is decremented to zero. If $\overline{\text{DONE}}$ is asserted one setup time prior to the S5 falling edge (i.e., before or with $\overline{\text{DTACK}}$) during a device access, then the channel operation will be terminated following the operand transfer (see the DNS bit in the CSR). STR is cleared, and an interrupt is generated if INTN is set. The BCR is also decremented, and the SAPR and/or DAPR are incremented in the normal fashion. The use of DONE is not limited to external request generation only; it may also be used to externally terminate an internally generated IDMA transfer sequence.

Error Termination

When a fatal error occurs during an IDMA bus cycle, a bus error is used to abort the cycle and terminate the channel operation. STR is cleared, either BED or BES is set, and an error interrupt is generated if INTE is set.

3.1.5 IDMA Programming

Once the channel has been initialized with all parameters required for a transfer operation, it is started by setting the start operation (STR) bit in the CMR. After the channel has been started, any register that describes the current operation may be read but not modified (SAPR/DAPR, FCR, or BCR).

Once STR has been set, the channel is active and either accepts operand transfer requests in external mode or generates requests automatically in internal mode. When the first valid external request is recognized, the IDMA arbitrates for the bus. The $\overline{\text{DREQ}}$ input is ignored until STR is set.

STR is cleared automatically when the BCR reaches zero and the channel transfer is either terminated by $\overline{\text{DONE}}$ or the IDMA cycle is terminated by a bus error.

Channel transfer operation may be suspended at any time by clearing STR. In response, any operand transfer in progress will be completed, and the bus will be released. No further bus cycles will be started while STR remains negated. During this time, the M68000 core may access IDMA internal registers to determine channel status or to alter operation. When STR is set again, if a transfer request is pending, the IDMA will arbitrate for the bus and continue normal operation.

Interrupt handling for the IDMA is configured globally through the interrupt pending register (IPR), the IMR, and the interrupt in-service register (ISR). Within the CMR in the IDMA, two bits are used to either mask or enable the presence of an interrupt reported in the CSR of the IDMA. One bit is used for masking normal termination; the other bit is used for masking error termination. When these interrupt mask bits in the CMR (INTN and INTE) are cleared and the IDMA status changes, status bits are set in the CSR but not in the IPR. When either INTN or INTE is set and the corresponding event occurs, the appropriate bit is set in the IPR, and, if this bit is not masked, the interrupt controller will interrupt the M68000 core.

NOTE

WPV will be set, regardless of the value of WPVE.

RMCST—RMC Cycle Special Treatment

0 = The locked read-modify-write cycles of the TAS instruction will be identical to the M68000 ($\overline{AS}$ and $\overline{CS}$ will be asserted during the entire cycle). The arbiter will issue $\overline{BG}$, regardless of the M68000 core $\overline{RMC}$. If an IMP chip select is used, the $\overline{DTACK}$ generator will insert wait states on the read cycle only.

1 = The MC68302 uses $\overline{RMC}$ to negate $\overline{AS}$ and $\overline{CS}$ at the end of the read portion of the RMC cycle and reasserts $\overline{AS}$ and $\overline{CS}$ at the beginning of the write portion. $\overline{BG}$ will not be asserted until the end of the write portion. If an IMP chip select is used, the $\overline{DTACK}$ generator will insert wait states on both the read and write portion of the cycles.

The assertion of the $\overline{RMC}$ by the M68000 core is seen by the arbiter and will prevent the arbiter from issuing bus grants until the completion of M68000-initiated locked read-modify-write activity. After system reset, this bit defaults to zero.

EMWS—External Master Wait State (EMWS);

When EMWS is set and an external master is using the chip-select logic for $\overline{DTACK}$ generation or is synchronously reading from the internal peripherals (SAM = 1), one additional wait state will be inserted in external master cycle to external memory and peripherals and also in every cycle from the external master to MC68302 internal memory and peripherals. When EMWS is cleared, all synchronous internal accesses will be with zero wait states, and the chip-select logic will generate $\overline{DTACK}$ after the exact programmed number of wait states. The chip-select lines are asserted slightly earlier for internal master memory cycles than for an external master. EMWS should be set whenever these timing differences will necessitate an additional wait state for external masters. After system reset, this bit defaults to zero.

ADCE—Address Decode Conflict Enable

0 = $\overline{BERR}$ is not asserted by a conflict in the chip-select logic when two or more chip-select lines are programmed to overlap the same area.

1 = $\overline{BERR}$ is asserted by a conflict in the chip-select logic when two or more chip-select lines are programmed to overlap the same area.

After system reset, this bit defaults to zero.

NOTE

ADC will be set, regardless of the value of ADCE.

BCLM—Bus Clear Mask

0 = The arbiter does not use the M68000 core internal IPEND signal to assert the internal and external bus clear signals.

1 = The arbiter uses the M68000 core internal IPEND signal to assert the internal and external bus clear signals.

When an external master desires to gain ownership, the standard M68000 bus arbitration protocol should be used:

1. Issue $\overline{BR}$ (to the IMP on-chip bus arbiter).
2. Wait for $\overline{BG}$ (from the IMP on-chip bus arbiter).
3. When $\overline{BG}$ is asserted, wait for the negation of both $\overline{AS}$ and $\overline{BGACK}$.
4. Assert $\overline{BGACK}$ and begin external master bus cycles.
5. Negate $\overline{BR}$ (to the IMP on-chip bus arbiter), causing $\overline{BG}$ to be negated by the IMP on-chip bus arbiter.
6. Negate $\overline{BGACK}$ after the external master bus cycles have completed.

This protocol is also followed by the on-chip bus masters (IDMA, SDMA, and DRAM refresh) except that they request the bus internally from the on-chip bus arbiter.

In the disable CPU mode, the IMP makes requests for the bus rather than granting the bus. In such a system, the IMP functions as an external master, and the external processor (e.g., an MC68030) need not assert $\overline{BGACK}$ as it accesses the IMP's on-chip RAM and registers.

NOTE

When the IMP's BUSW pin is low causing the M68000 core to operate as an MC68008, the $\overline{BGACK}$ signal should still be used in bus arbitration control. On the original MC68008, the $\overline{BGACK}$ signal was not available externally, and therefore could not be used.

3.8.6 Hardware Watchdog

The hardware watchdog logic is used to assert $\overline{BERR}$ and set HWT when a bus cycle is not terminated by $\overline{DTACK}$ and after a programmable number of clock cycles has elapsed. The hardware watchdog logic has a 10-bit downcounter and a 4-bit prescaler. When enabled, the watchdog timer commences counting clock cycles as $\overline{AS}$ is asserted (for internal or external bus masters). The count is terminated normally by the negation of $\overline{AS}$; however, if the count reaches zero before $\overline{AS}$ is negated, $\overline{BERR}$ will be asserted until $\overline{AS}$ is negated. The hardware watchdog will operate with internal as well as external bus masters.

The hardware watchdog logic uses four bits in the SCR.

HWDEN—Hardware Watchdog Enable

0 = The hardware watchdog is disabled.
1 = The hardware watchdog is enabled.

After system reset, this bit defaults to one to enable the hardware watchdog.

reduce power. An external pullup should be used if TCLK1 is not driven externally. TSTCLK1 may be toggled at any time, but the SCC1 transmitter should be disabled and re-enabled if any dynamic change is made on TSTCLK1 during the operation of the SCC1 transmitter.

TSRCLK1—Three-state RCLK1

0 = Normal operation

1 = The RCLK1 pin is three-stated. This option may be used to prevent contention on the RCLK1 pin if an external clock is provided to the RCLK1 pin while the SCC1 baud rate generator is output on RCLK1. This option may also be chosen if it is required to run the SCC1 baud rate generator at high speed (for instance in a high speed UART application), but the RCLK1 output is not needed, and it is desired to reduce power. An external pullup should be used if RCLK1 is not driven externally. TSRCLK1 may be toggled at any time, but the SCC1 receiver should be disabled and re-enabled if any dynamic change is made on TSRCLK1 during the operation of the SCC1 receiver.

DBRG1—Disable BRG1

0 = Normal operation

1= The BRG1 pin is disabled and is driven high. This option should be chosen if it is required to run the SCC1 baud rate generator at high speed, but the BRG1 output is not needed and it is desired to reduce power. Although DBRG1 may be modified at any time, the user should note that glitches on BRG1 are not prevented by the MC68302 when the state of DBRG1 is changed.

Bits 10 - 0—Reserved. Should be written with zeros.

3.9.1 Freeze Control

Used to freeze the activity of selected peripherals, $\overline{FRZ}$ is useful for system debugging purposes. When $\overline{FRZ}$ is asserted:

- The CP main controller freezes its activity on the next clock (CLKO) and will continue in a frozen state as long as $\overline{FRZ}$ remains asserted. No new interrupt requests and no memory accesses (internal or external) will occur, and the main controller will not access the serial channels.
- The IDMA completes any bus cycle that is in progress (after $\overline{DTACK}$ is asserted) and releases bus ownership. No further bus cycles will be started as long as $\overline{FRZ}$ remains asserted.
- Each timer can be programmed to freeze by setting the appropriate bit in the SCR. After a one-clock (CLKO) delay, the selected timers will freeze their activity (count, capture) as long as $\overline{FRZ}$ remains asserted.

NOTE

Regardless of whether or not the freeze logic is used, FRZ must be negated during system reset.

FRZ1—Freeze Timer 1 Enable

0 = Freeze timer 1 logic is disabled.
1 = Freeze timer 1 logic is enabled.

After system reset, this bit defaults to zero.

FRZ2—Freeze Timer 2 Enable

0 = Freeze timer 2 logic is disabled.
1 = Freeze timer 2 logic is enabled.

After system reset, this bit defaults to zero.

FRZW—Freeze Watchdog Timer Enable

0 = Freeze watchdog timer logic is disabled.
1 = Freeze watchdog timer logic is enabled.

After system reset, this bit defaults to zero.

No other MC68302 peripherals are directly affected by the freeze logic; however, consequential errors such as receiver overruns in the SCC FIFOs may occur due to the CP main controller being disabled. Note that use of the freeze logic does not clear any IPR bits that were already set.

3.10 DYNAMIC RAM REFRESH CONTROLLER

The communications processor (CP) main (RISC) controller may be configured to handle the dynamic RAM (DRAM) refresh task without any intervention from the M68000 core. Use of this feature requires a timer or SCC baud rate generator (either from the MC68302 or externally), the I/O pin PB8, and two transmit buffer descriptors from SCC2 (Tx BD6 and Tx BD7).

The DRAM refresh controller routine executes in 25 clock cycles. Assuming a refresh cycle every 15.625 µs, two wait state DRAMs, and a 16.67-MHz EXTAL frequency, this routine uses about 10 percent of the microcontroller bandwidth and 4 percent of the M68000 bus bandwidth. The refresh cycle will not be executed during a period that a bus exception (i.e., $\overline{\text{RESET}}$, $\overline{\text{HALT}}$, or $\overline{\text{BERR}}$) is active. The refresh cycle is a standard M68000-type read cycle (an SDMA byte read cycle). It does not generate row address strobe (RAS) and column address strobe (CAS) to the external DRAM. These functions require an external PAL. Use of the DRAM refresh controller will slightly reduce the maximum possible serial data rates of the SCCs.

3.10.1 Hardware Setup

An output of timer 1 or timer 2 (the $\overline{\text{TOUT}}$ pin) or one of the SCC's baud rate generator outputs (BRG3–BRG1) should be connected externally to PB8. A high-to-low transition on this edge causes a request to be generated to the main controller to perform one refresh cycle. The DRAM refresh request takes priority over all SCC channels and commands given to the CP command register.

A block diagram of an MC68302 DRAM system is shown in Figure 3-13. The MC68302 generates standard M68000 read and write cycles that must be converted to DRAM read and write cycles. The address buffers provide the multiplexing of the row and column addresses

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

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

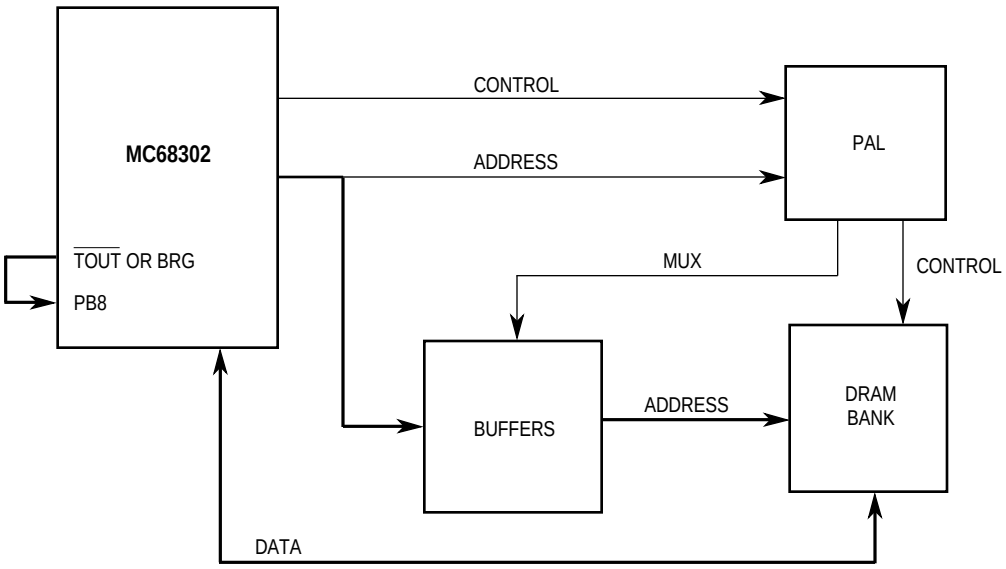


Figure 3-13. DRAM Control Block Diagram

3.10.2 DRAM Refresh Controller Bus Timing

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

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

3.10.3 Refresh Request Calculations

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

SECTION 4
COMMUNICATIONS PROCESSOR (CP)

The CP includes the following modules:

- Main Controller (RISC Processor)
- Six Serial Direct Memory Access (SDMA) Channels
- A Command Set Register
- Serial Channels Physical Interface Including:

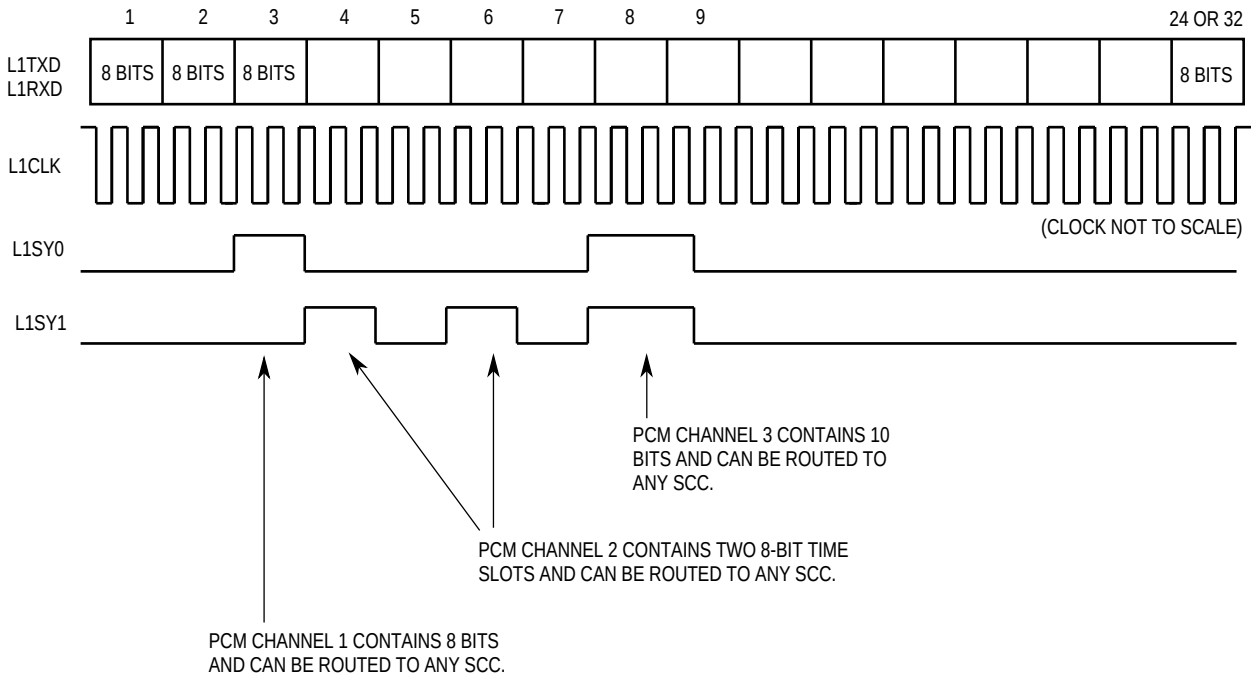
  —Motorola Interchip Digital Link (IDL)
  —General Circuit Interface (GCI), also known as IOM-2
  —Pulse Code Modulation (PCM) Highway Interface
  —Nonmultiplexed Serial Interface (NMSI) Implementing Standard
  —Modem Signals
- Three Independent Full Duplex Serial Communication Controllers (SCCs) Supporting the Following Protocols:

  —High-Level/Synchronous Data Link Control (HDLC/SDLC)
  —Universal Asynchronous Receiver Transmitter (UART)
  —Binary Synchronous Communication (BISYNC)
  —Synchronous/Asynchronous Digital Data Communications Message
  —Protocol (DDCMP)
  —Transparent Modes
  —V.110 Rate Adaption
- Serial Communication Port (SCP) for Synchronous Communication
- Two Serial Management Controllers (SMCs) to Support the IDL and GCI Management Channels

4.1 MAIN CONTROLLER

The CP main controller is a RISC processor that services the three SCCs, the SCP, and the SMCs. Its primary responsibilities are to work with the serial channels to implement the user-chosen protocol and to manage the SDMA channels that transfer data between the SCCs and memory. The CP main controller also executes commands issued by the M68000 core (or an external processor) and generates interrupts to the interrupt controller.

The operation of the main controller is transparent to the user, executing microcode located in a private internal ROM (see Figure 4-1). Commands may be explicitly written to the main controller by the M68000 core through the CP command register. Additionally, commands and status are exchanged between the main controller and the M68000 core through the



NOTE: Whenever the syncs are active, data from that SCC is transmitted and received using L1CLK edges.

Figure 4-10. PCM Channel Assignment on a T1/CEPT Line

4.4.4 Nonmultiplexed Serial Interface (NMSI)

The IMP supports the NMSI with modem signals. In this case, the serial interface connects the seven serial lines of the NMSI1/ISDN interface (RXD1, TXD1, RCLK1, TCLK1, CD1, CTS1, and RTS1) directly to the SCC1 controller. NMSI pins associated with SCC2 and SCC3 can be used as desired or left as general-purpose I/O port pins. See 3.3 Parallel I/O Ports for an example. $\overline{\text{RTS}}$ is an output of the transmitter, while $\overline{\text{CTS}}$ and $\overline{\text{CD}}$ are inputs to the transmitter and receiver, respectively. See 4.5.3 SCC Mode Register (SCM) and 4.4 Serial Channels Physical Interface for additional information.

$\overline{\text{CTS}}$ and $\overline{\text{CD}}$ may be programmed to control transmission and reception automatically or to just generate interrupts.

4.4.5 Serial Interface Registers

There are two serial interface registers: SIMODE and SIMASK. The SIMODE register is a 16-bit register used to define the serial interface operation modes. The SIMASK register is a 16-bit register used to determine which bits are active in the B1 and B2 channels of ISDN.

4.4.5.1 Serial Interface Mode Register (SIMODE)

If the IDL or GCI mode is used, this register allows the user to support any or all of the ISDN channels independently. Any extra SCC channel can then be used for other purposes in

Table 4-6. SCC Parameter RAM Memory Map

Address	Name	Width	Description
SCC Base + 80 #	RFCR	Byte	Rx Function Code
SCC Base + 81 #	TFCR	Byte	Tx Function Code
SCC Base + 82 #	MRBLR	Word	Maximum Rx Buffer Length
SCC Base + 84 ##		Word	Rx Internal State
SCC Base + 86 ##		Byte	Reserved
SCC Base + 87 ##	RBD#	Byte	Rx Internal Buffer Number
SCC Base + 88		2 Words	Rx Internal Data Pointer
SCC Base + 8C		Word	Rx Internal Byte Count
SCC Base + 8E		Word	Rx Temp
SCC Base + 90 ##		Word	Tx Internal State
SCC Base + 92 ##		Byte	Reserved
SCC Base + 93 ##	TBD#	Byte	Tx Internal Buffer Number
SCC Base + 94		2 Words	Tx Internal Data Pointer
SCC Base + 98		Word	Tx Internal Byte Count
SCC Base + 9A		Word	Tx Temp
SCC Base + 9C			First Word of Protocol-Specific Area
SCC Base + BF			Last Word of Protocol-Specific Area

Initialized by the user (M68000 core).

Modified by the CP following a CP or system reset.

Certain parameter RAM values need to be initialized by the user before the SCC is enabled. Those values not so designated are initialized/written by the CP. Once initialized, most parameter RAM values will not need to be accessed in user software since most of the activity is centered around the transmit and receive buffer descriptors, not the parameter RAM. However, if the parameter RAM is accessed by the user, the following should be noted. The parameter RAM can be read at any time. The parameter RAM values related to the SCC transmitted can only be written 1) whenever the ENT bit in the SCM is zero or 2) after a STOP TRANSMIT command and before a RESTART TRANSMIT command. The parameter RAM values related to the SCC receiver can only be written 1) whenever the ENR bit in the SCM is zero or 2) if the receiver has previously been enabled, after the ENTER HUNT MODE command and before the ENR bit is set. See 4.5.10 Disabling the SCCs for a discussion of when the SCC registers may be changed.

The registers (see Table 4-6) that typically need to be accessed by the user are described in the following paragraphs.

4.5.6.1 Data Buffer Function Code Register (TFCR, RFCR)

This register defines the address space of the receive (RFCR) and transmit (TFCR) data buffers. These registers must be initialized if the SCC is used.

NOTE

The value of the function code register for any channel may be equal to that of any other, but do not initialize FC2–FC0 with the value “111” which causes a conflict with the interrupt acknowledge cycle to occur.

7	6	5	4	3	2	1	0
0	FC2	FC1	FC0	0	0	0	0

T2,T1—Message Type

00 = Data message
01 = Control message
10 = Maintenance message
11 = Reserved

Bits 7–6—Reserved for future use.

CF—CRC Follow Error

The character following the CRC for this message was not one of SOH, ENQ, DLE, SYN, or IDLE. The receiver then enters hunt mode.

FR—Framing Error

A character with a framing error was received. The associated character may be found at the last location in this buffer. A framing error is detected by the UART controller when no stop bit is detected in the receive data string.

NOTE

This error can occur only on asynchronous DDCMP links.

PR—Parity Error

A character with a parity error was received. The associated character may be found at the last location in this buffer.

NOTE

This error can occur only on asynchronous DDCMP links.

CR—Rx CRC Error

A message with a CRC error was received in the header (CRC1) or data (CRC2) fields or a control message (CRC3).

OV—Overrun

A receiver overrun occurred during message reception.

CD—Carrier Detect Lost

The CD signal was negated during message reception. This bit is valid only when working in NMSI mode.

Data Length

The data length is the number of octets that the DDCMP controller has written to this BD's data buffer. It is written by the CP once as the BD is closed.

NOTE

The actual buffer size should be greater than or equal to eight (to ensure the header is received in one buffer).

	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.

$\overline{FRZ}$—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.

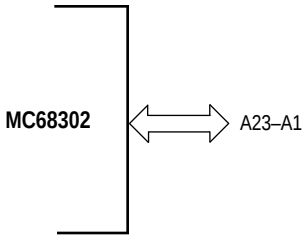


Figure 5-4. Address Bus Pins

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.

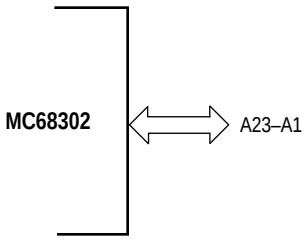


Figure 5-5. Data Bus Pins

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.

5.12 TYPICAL SERIAL INTERFACE PIN CONFIGURATIONS

Table 5-4 shows typical configurations of the physical layer interface pins for an ISDN environment. Table 5-6 shows potential configurations of the physical layer interface pins for a non-ISDN environment. The IDMA, IACK, and timer pins can be used in all applications either as dedicated functions or as PIO pins.

Table 5-5. Typical ISDN Configurations

Pins	Connected To	Used As
NMSI1 or ISDN I/F	SCC1 and SCC3	SCC1 Used as ISDN D-ch SCC3 Used as ISDN B2-ch
NMSI2	SCC2	SCC2 is Connected to Terminal
NMSI3	PA12–PA8 SCP	PIO (Extra Modem Signals and SCP Select Signals) Status/Control Exchange

NOTES:
1. ISDN environment with SCP port for status/control exchange and with existing terminal (for rate adaption).
2. D-ch is used for signaling.
3. B1-ch is used for voice (external CODEC required).
4. B2-ch is used for data transfer.

Table 5-6. Typical Generic Configurations

Pins	Connected To	Used As
NMSI1 or ISDN I/F	SCC1	Terminal with Modem
NMSI2	SCC2	Terminal with Modem
NMSI3 (5)	SCC3	Terminal without Modem
NMSI3 (3)	SCP	Status/Control Exchange

NOTE: Generic environment with three SCC ports (any protocol) and the SCP port. SCC3 does not use modem control signals.

5.13 NMSI1 OR ISDN INTERFACE PINS

The NMSI1 or ISDN interface pins are shown in Figure 5-10.

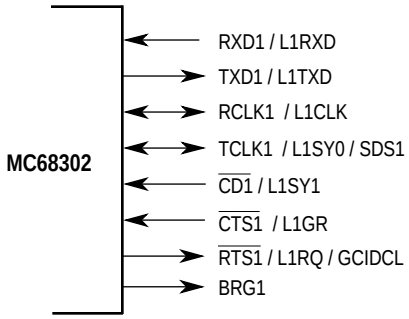


Figure 5-10. NMSI1 or ISDN Interface Pins

synchronous input using the asynchronous input setup time (#47).

4. For power-up, the MC68302 must be held in the reset state for 100 ms to allow stabilization of on-chip circuit. After the system is powered up #56 refers to the minimum pulse width required to reset the processor.
5. If the asynchronous input setup (#47) requirement is satisfied for $\overline{\text{DTACK}}$, the $\overline{\text{DTACK}}$ asserted to data setup time (#31) requirement can be ignored. The data must only satisfy the data-in to clock low setup time (#27) for the following clock cycle.
6. When $\overline{\text{AS}}$ and R/$\overline{\text{W}}$ are equally loaded (±20%), subtract 5 ns from the values given in these columns.
7. The MC68302 will negate $\overline{\text{BG}}$ and begin driving the bus if external arbitration logic negates $\overline{\text{BR}}$ before asserting $\overline{\text{BGACK}}$.
8. The minimum value must be met to guarantee proper operation. If the maximum value is exceeded, $\overline{\text{BG}}$ may be reasserted.
9. This specification is valid only when the RMCST bit is set in the SCR register.
10. Occurs on S0 of SDMA read/write access when the SDMA becomes bus master.
11. Specification may be exceeded during the TAS instruction if the RMCST bit in the SCR is set.

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.

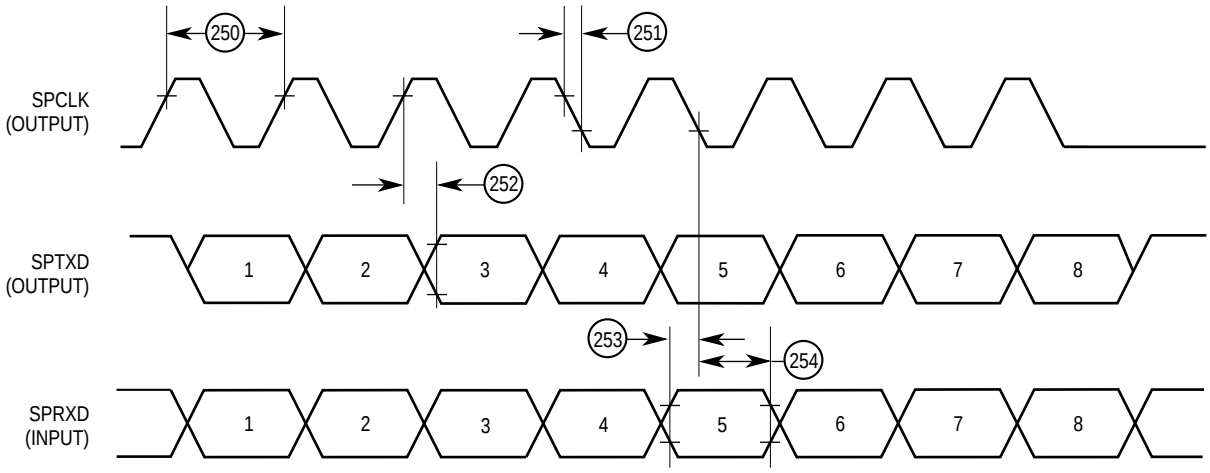
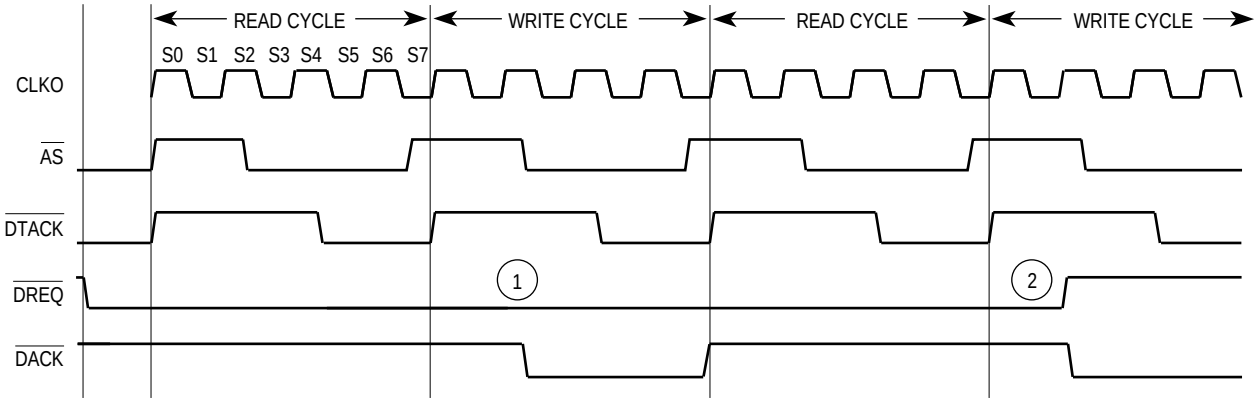


Figure 6-19. Serial Communication Port Timing Diagram

Figure D-10 illustrates the activation of external burst mode by using the DREQ signal as a level-sensitive input to the IDMA. If DREQ is asserted when the IDMA is accessing the peripheral (indicated by DACK being asserted) as shown in Figure D-10, then the IDMA will continue servicing the peripheral by performing another sequence of operand transfer cycles. The external burst mode stops when the DREQ signal is deasserted prior to the trailing edge of S3 in the cycle where DACK is asserted.



NOTE: DREQ is sampled on the falling edge of clock.

LEGEND:

(1) DREQ asserted prior to DTACK = continue burst mode transfer

(2) DREQ negated prior to DTACK = relinquish the bus

Figure D-10. Burst Mode Cycles

D.5.7 Internal Interrupt Sequence

An interrupt acknowledge cycle (IACK) occurs when an allowed internal or external interrupt request is pending and the priority of the interrupt is higher than the current microprocessor run level. The interrupt acknowledge cycle begins at the conclusion of instruction execution in state S0. All internal resources, including the IDMA, generate INRQ requests at level 4.

The four registers used in interrupt processing are as follows:

1. The interrupt mask register (IMR) contains the flags that, when set, allow the INRQ source to initiate service.
2. The interrupt pending register (IPR) contains bits that correspond to the INRQ source requesting service.
3. The interrupt in-service register (ISR) indicates which internal interrupts are currently being processed (usually only one at a time).
4. The global interrupt mode register (GIMR) has bits that specify interrupt modes such as the edge or level of an input that triggers an interrupt.

A level 4 interrupt may be generated by the IDMA upon completion of a data block transfer. Interrupt processing of IDMA transfers is possible by 1) setting the IDMA interrupt enable (bit 11) in the IMR and 2) setting one or both interrupt enable (INTN and INTE) bits in the CMR (see Table D-1). Once in the interrupt handler, four bits in the CSR indicate the reason for termination of an IDMA data block (see Table D-2).

method works because the parallel I/O lines default as inputs to the MC68302 and can therefore all be pulled high initially. After the slave BARs are programmed, the parallel I/O lines on the master should be reconfigured as inputs; otherwise, a contention could occur on A23 when a slave's DMA is accessing the bus.

This method is the easiest because it requires no external glue. It costs one parallel I/O line per slave on the master MC68302 and reduces the address space of each slave from 16 MB to 8 MB, neither of which should be a problem in most systems. If A23 is really needed on the slaves, it can be regained, but extra logic is required.

D.7.4 Dealing with Interrupts

The following example is the easiest method for dealing with interrupts from the slaves. It assumes that any other external interrupt sources are sent directly to the master MC68302 without using the interrupt controllers on the slaves.

1. The internal interrupts cause the slaves to force out level 4 on their $\overline{IOUT2}$-$\overline{IOUT0}$ pins. (AVEC, RMC, and CS0 are not available on the slaves.)
2. IOUT2 from the slave is connected to the master PB8, PB9, PB10, or PB11 pin if the master is in normal interrupt mode, or to $\overline{IRQ1}$ or $\overline{IRQ6}$ if it is in dedicated interrupt mode. Thus, in normal mode the slave interrupts will arrive at level 4, and in dedicated mode they will arrive at either level 1 or 6.
3. The best method of operation is for the slaves not to generate the vector during the interrupt acknowledge cycle. The master MC68302 can generate it and then read the IPR of the slave to determine the actual source of the interrupt.

NOTE

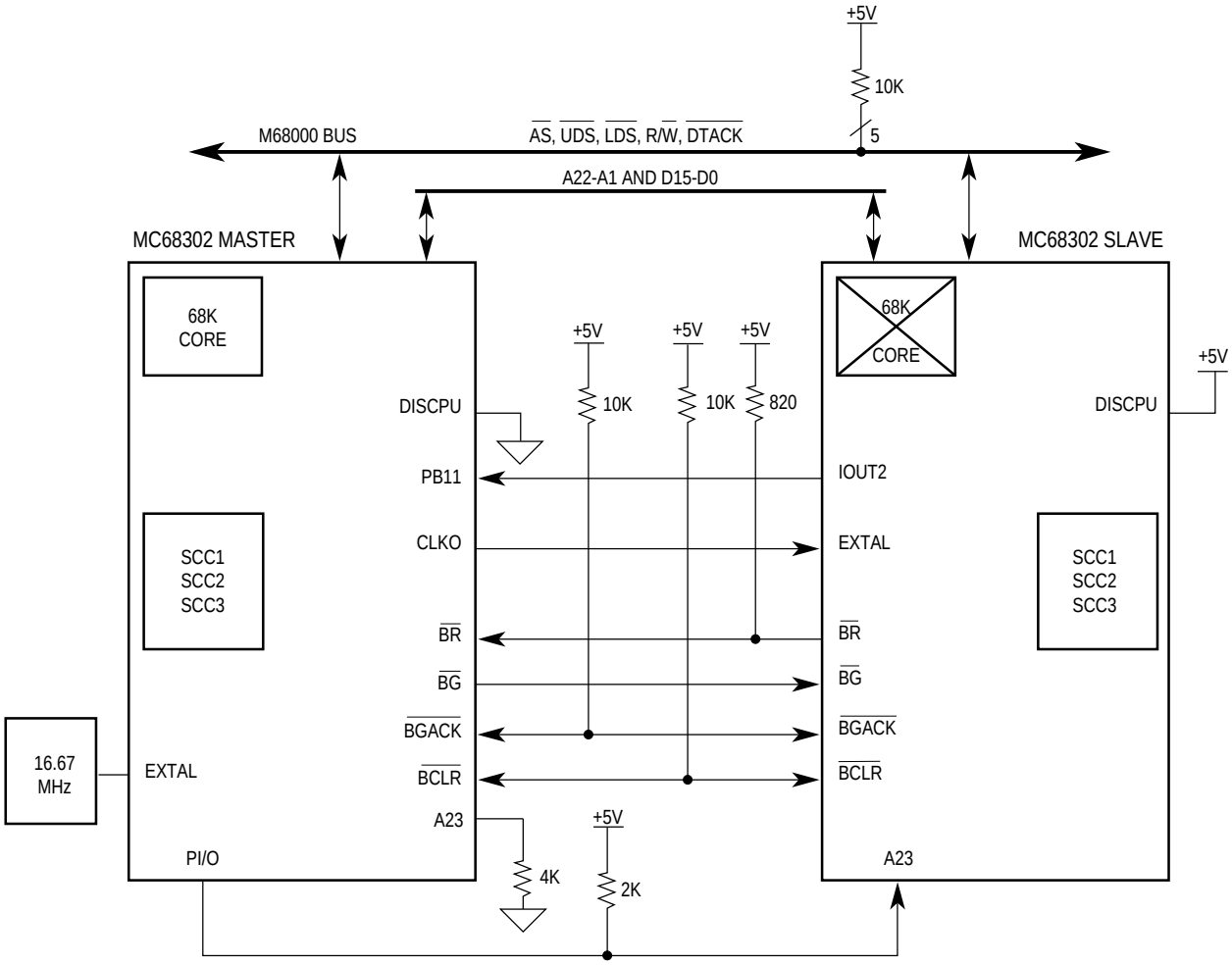
Why not use the vector generation enable (VGE) bit in each slave MC68302 to allow the slaves to generate different vectors for each of their internal peripherals? This could be done; however, the slaves must be tricked into responding to an unique interrupt level (or else multiple vectors could collide on the bus simultaneously). The external decoding and address buffer logic required to do this slows down the interface timing (and adds expense). Rather, the VGE bit is intended for applications where a single MC68302 is a slave to another processor such as the MC68020.

To use the interrupt controller on a slave MC68302 (in either normal or dedicated mode) to handle interrupt levels 1, 6, or 7 from an external peripheral, connect the slave's $\overline{IOUT2}$-$\overline{IOUT0}$ pins directly to the master's $\overline{IPL2}$-$\overline{IPL0}$ pins. The master MC68302 will supply the vector for levels 1, 6, and 7, and level 4 of the slave will be interpreted as an error vector (00000). Upon branching to this vector, the master MC68302 software should then check the slave's IPR to identify the source.

D.7.5 Arbitration

If only one slave is present, no arbiter is required because the $\overline{BR}$ as an output on the slave can be sent directly to the $\overline{BR}$ input on the master. Figure D-20 shows a dual master-slave system using this arbitration scheme. Note that the $\overline{BCLR}$ pin from the slave MC68302 can be used to give the 12 SDMA channels in the system priority over the two IDMA channels in the system. $\overline{BCLR}$ is asserted whenever an SDMA channel wants the bus, and, will tem-

protocol, where a 16x oversampling is employed to effectively extract the clock information from the data. With transparent mode (and all the other protocols), this clock can be generated internally with a baud rate generator or can be provided externally.



NOTE: The M68000 core on the master controls all six SCCs.

Figure D-20. Dual Master-Slave System

D.8.2 Applications for Transparent Mode

There are several basic applications for the use of transparent mode.

First, some data may need to be moved serially but may not require protocol superimposed — for example, voice data. There is no reason to encode voice data, and no error correction is needed. With voice data, an occasional dropped bit will not interfere with the data stream in any significant way. The MC68302 transparent mode works well for this type of application.

Second, some board-level applications require a serial-to-parallel and parallel-to-serial conversion. Often this is done to allow communication between chips on the same board. The

E.3.1.2 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 transparent functions have been given for those parameters and bits that are protocol specific.

E.3.1.2.1 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 of 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 TCLK pin.

CD1 0-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.3.1.2.2 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 transparent-specific parameters. Note that reserved bits in registers should be written as zeros.

15	14	13	12	11	10	9	8
—	EXSYN	NTSYN	REVD	—	—	—	—

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