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-BQFP Bumpered
Supplier Device Package	132-PQFP (46x46)
Purchase URL	https://www.e-xfl.com/product-detail/nxp-semiconductors/mc68302fc25c

The MC68302 can also be used in applications such as board-level industrial controllers performing real-time control applications with a local control bus and an X.25 packet network connection. Such a system provides the real-time response to a demanding peripheral while permitting remote monitoring and communication through an X.25 packet network.

1.2 FEATURES

The features of the IMP are as follows:

- On-Chip HCMOS MC68000/MC68008 Core Supporting a 16- or 8-Bit M68000 Family-System
- IB Including:
  - —Independent Direct Memory Access (IDMA) Controller with Three Handshake Signals: $\overline{\text{DREQ}}$, $\overline{\text{DACK}}$, and $\overline{\text{DONE}}$.
  - —Interrupt Controller with Two Modes of Operation
  - —Parallel Input/Output (I/O) Ports, Some with Interrupt Capability
  - —On-Chip 1152-Byte Dual-Port RAM
  - —Three Timers Including a Watchdog Timer
  - —Four Programmable Chip-Select Lines with Wait-State Generator Logic
  - —Programmable Address Mapping of the Dual-Port RAM and IMP Registers
  - —On-Chip Clock Generator with Output Signal
  - —System Control:
    - Bus Arbitration Logic with Low-Interrupt Latency Support
    - System Status and Control Logic
    - Disable CPU Logic (M68000)
    - Hardware Watchdog
    - Low-Power (Standby) Modes
    - Freeze Control for Debugging
    - DRAM Refresh Controller
- CP Including:
  - —Main Controller (RISC Processor)
  - —Three Independent Full-Duplex Serial Communications Controllers (SCCs)
  - —Supporting Various 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
  - —Six Serial DMA Channels for the Three SCCs
  - —Flexible Physical Interface Accessible by SCCs Including:
    - Motorola Interchip Digital Link (IDL)
    - General Circuit Interface (GCI, also known as IOM[3]-2)
    - Pulse Code Modulation (PCM) Highway Interface

[3] IOM is a trademark of Siemens AG

SECTION 2
MC68000/MC68008 CORE

The MC68302 integrates a high-speed M68000 processor with multiple communications peripherals. The provision of direct memory access (DMA) control and link layer management with the serial ports allows high throughput of data for communications-intensive applications, such as basic rate Integrated Services Digital Network (ISDN).

The MC68302 can operate either in the full MC68000 mode with a 16-bit data bus or in the MC68008 mode with an 8-bit data bus by tying the bus width (BUSW) pin low. $\overline{UDS}$/A0 functions as A0 and $\overline{LDS}$/$\overline{DS}$ functions as $\overline{DS}$ in the MC68008 mode.

NOTE

The BUSW pin is static and is not intended to be used for dynamic bus sizing. If the state of BUSW is changed during operation of the MC68302, erratic operation may occur.

Refer to the MC68000UM/AD, *M68000 8-/16-/32-Bit Microprocessors User's Manual*, for complete details of the on-chip microprocessor. Throughout this manual, references may use the notation M68000, meaning all devices belonging to this family of microprocessors, or the notation MC68000, MC68008, meaning the specific microprocessor products.

2.1 PROGRAMMING MODEL

The M68000 microprocessor executes instructions in one of two modes: user or supervisor. The user mode provides the execution environment for most of the application programs. The supervisor mode, which allows some additional instructions and privileges, is intended for use by the operating system and other system software.

Shown in Figure 2-1, the M68000 core programming model offers 16, 32-bit, general-purpose registers (D7–D0, A7–A0), a 32-bit program counter (PC), and an 8-bit condition code register (CCR) when running in user space. The first eight registers (D7–D0) are used as data registers for byte (8-bit), word (16-bit), and long-word (32-bit) operations. The second set of seven registers (A6–A0) and the stack pointer (USP in user space) may be used as software stack pointers and base address registers. In addition, the address registers may be used for word and long-word operations. All 16 registers may be used as index registers.

Table 2-2. M68000 Instruction Set Summary

Mnemonic	Description
ABCD ADD AND ASL ASR	Add Decimal with Extend Add Logical AND Arithmetic Shift Left Arithmetic Shift Right
Bcc BCHG BCLR BRA BSET BSR BTST	Branch Conditionally Bit Test and Change Bit Test and Clear Branch Always Bit Test and Set Branch to Subroutine Bit Test
CHK CLR CMP	Check Register Against Bounds Clear Operand Compare
DBcc DIVS DIVU	Decrement and Branch Conditionally Signed Divide Unsigned Divide
EOR EXG EXT	Exclusive OR Exchange Registers Sign Extend
JMP JSR	Jump Jump to Subroutine
LEA LINK LSL LSR	Load Effective Address Link Stack Logical Shift Left Logical Shift Right
MOVE MULS MULU	Move Source to Destination Signed Multiply Unsigned Multiply
NBCD NEG NOP NOT	Negate Decimal with Extend Negate No Operation Ones Complement
OR	Logical OR
PEA	Push Effective Address
RESET ROL ROR ROXL ROXR RTE RTR RTS	Reset External Devices Rotate Left without Extend Rotate Right without Extend Rotate Left with Extend Rotate Right with Extend Return from Exception Return and Restore Return from Subroutine
SBCD Scc STOP SUB SWAP	Subtract Decimal with Extend Set Conditionally Stop Subtract Swap Data Register Halves
TAS TRAP TRAPV TST	Test and Set Operand Trap Trap on Overflow Test
UNLK	Unlink

Table 2-5. M68000 Exception Vector Assignment

11	44	02C	SD	Line 1111 Emulator
12[1]	48	030	SD	(Unassigned, Reserved)
13[1]	52	034	SD	(Unassigned, Reserved)
14[1]	56	038	SD	(Unassigned, Reserved)
15	60	03C	SD	Uninitialized Interrupt Vector
16–23[1]	64	040	SD	(Unassigned, Reserved)
	92	05C	SD	
24	96	060	SD	Spurious Interrupt[3]
25	100	064	SD	Level 1 Interrupt Autovector
26	104	068	SD	Level 2 Interrupt Autovector
27	108	06C	SD	Level 3 Interrupt Autovector
28	112	070	SD	Level 4 Interrupt Autovector
29	116	074	SD	Level 5 Interrupt Autovector
30	120	078	SD	Level 6 Interrupt Autovector
31	124	07C	SD	Level 7 Interrupt Autovector
32–47	128	080	SD	TRAP Instruction Vectors[4]
	188	0BC	SD	
48–63[1]	192	0C0	SD	(Unassigned, Reserved)
	255	0FC	SD	
64–255	256	100	SD	User Interrupt Vectors
	1020	3FC	SD	

NOTES:

1. Vector numbers 12–14, 16–23, and 48–63 are reserved for future enhancements by Motorola (with vectors 60–63 being used by the M68302 (see 2.7 MC68302 IMP Configuration and Control)). No user peripheral devices should be assigned these numbers.
2. Unlike the other vectors which only require two words, reset vector (0) requires four words and is located in the supervisor program space.
3. The spurious interrupt vector is taken when there is a bus error indication during interrupt processing.
4. TRAP # n uses vector number 32 + n.

2.4.2 Exception Stacking Order

Exception processing saves the most volatile portion of the current processor context on top of the supervisor stack. This context is organized in a format called the exception stack frame. The amount and type of information saved on the stack is determined by the type of exception. The reset exception causes the M68000 to halt current execution and to read a new SSP and PC as shown in Table 2-5. A bus error or address error causes the M68000 to store the information shown in Figure 2-3. The interrupts, traps, illegal instructions, and trace stack frames are shown in Figure 2-4.

Table 2-8. Parameter RAM

Base + 580 • • • Base + 5BF		SCC2 SCC2	Specific Protocol Parameters
Base + 5C0 • • • Base + 5FF			Reserved (Not Implemented)
Base + 600 Base + 608 Base + 610 Base + 618 Base + 620 Base + 628 Base + 630 Base + 638 Base + 640 Base + 648 Base + 650 Base + 658 Base + 660 Base + 666 Base + 668 Base + 66A Base + 66C Base + 66E # Base +67A Base +67C Base +67E #	4 Word 4 Word 4 Word 4 Word 4 Word 4 Word 4 Word 4 Word 4 Word 4 Word 4 Word 4 Word 3 Word Word Word Word Word 6 Word Word Word Word	SCC3 SCC3 SCC3 SCC3 SCC3 SCC3 SCC3 SCC3 SCC3 SCC3 SCC3 SCC3 SMC SCC1 SCC1 SCC2 SCC2 SMC1–SMC2 SCP SCC1–SCC3 CP	RxBD0 RxBD1 RxBD2 RxBD3 RxBD4 RxBD5 RxBD6 RxBD7 TxBD0 TxBD1 TxBD2 TxBD3 ## Reserved RxBD TxBD RxBD TxBD Internal Use Rx/TxBD BERR Channel Number MC68302 Revision Number
Base + 680 • • • Base + 6BF		SCC3 SCC3	Specific Protocol Parameters
Base + 6C0 • • • Base + 7FF			Reserved (Not Implemented)

Modified by the CP after a CP or system reset.

Tx BD 4, 5, 6, and 7 are not initially available to SCC3. (See 4.5.5 Buffer Descriptors Table for information on how they may be regained.)

In addition to the internal dual-port RAM, a number of internal registers support the functions of the various M68000 core peripherals. The internal registers (see Table 2-9) are memory-mapped registers offset from the BAR point1616er and are located on the internal M68000 bus.

Table 3-2. IDMA Bus Cycles

Source Size	Source Address	Destination Size	Destination Address	Cycles 8-bit Bus	Cycles 16-bit Bus
8	X	8	X	RW	RW
8	X	16	Even	RWRW*	RRW
8	X	16	Odd	RWRW*	RRWW
16	Even	8	X	RWRW*	RWW
16	Odd	8	X	RWRW*	RRWW
16	Even	16	Even	RWRW*	RW
16	Even	16	Odd	RWRW*	RWW
16	Odd	16	Even	RWRW*	RRW
16	Odd	16	Odd	RWRW*	RRWW

* - Considered as 2 operands.

3.1.4.4 Transfer Request Generation

IDMA transfers may be initiated by either internally or externally generated requests. Internally generated requests can be initiated by setting STR in the CMR. Externally generated transfers are those requested by an external device using $\overline{\text{DREQ}}$ in conjunction with the activation of STR.

Internal Maximum Rate

The first method of internal request generation is a nonstop transfer until the transfer count is exhausted. If this method is chosen, the IDMA will arbitrate for the bus and begin transferring data after STR is set and the IDMA becomes the bus master. If no exception occurs, all operands in the data block will be transferred in sequential bus cycles with the IDMA using 100 percent of the available bus bandwidth (unless an external bus master requests the bus or the M68000 core has an unmasked pending interrupt request and BCLM = 1). See 3.1.6 DMA Bus Arbitration for more details.

Internal Limited Rate

To guarantee that the IDMA will not use all the available system bus bandwidth during a transfer, internal requests can be limited to the amount of bus bandwidth allocated to the IDMA. Programming the REQG bits to “internal limited rate” and the BT bits to limit the percentage of bandwidth achieves this result. As soon as STR is set, the IDMA module arbitrates for the bus and begins to transfer data when it becomes bus master. If no exception occurs, transfers will continue uninterrupted, but the IDMA will not exceed the percentage of bus bandwidth programmed into the control register (12.5%, 25%, 50%, or 75%). This percentage is calculated over each ensuing 1024 internal clock cycle period.

For example, if 12.5% is chosen, the IDMA will attempt to use the bus for the first 128 clocks of each 1024 clock cycle period. However, because of other bus masters, the IDMA may not be able to take its 128 clock allotment in a single burst.

External Burst Mode

For external devices requiring very high data transfer rates, the external burst mode allows the IDMA to use all the bus bandwidth to service the device. In the burst mode, the

3.10.4 Initialization

The user should first initialize the refresh routine parameters in the SCC2 parameter RAM. These parameters are the DRAM low starting address, the DRAM high starting address, the DRAM address increment step (number of bytes in a row), the count (number of rows), and a temporary count. Then, mask the PB8 bit in the IMR (unless an interrupt is desired on each refresh request). Next, the timer or baud rate generator should be programmed to provide the desired refresh clock to the PB8 pin. Next, the ERRE bit in the SCR should be set. Then, upon every high-to-low transition of PB8, the refresh routine executes one refresh (read) cycle.

ERRE—External RISC Request Enable

0 = Normal operation.
1 = When this bit is set, a high-to-low transition on PB8 causes the CP to execute the DRAM refresh routine.

3.10.5 DRAM Refresh Memory Map

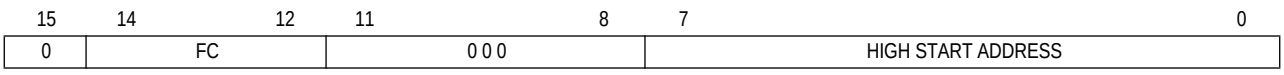
The DRAM refresh memory map replaces the SCC2 TxBD6 and TxBD7 structures in the parameter RAM. The wrap bit must therefore be set in SCC2 TxBD5 so that only six TxBDs are used for SCC2. These parameters should be written before the DRAM refresh controller receives its first request, but may be read at any time. They are undefined at reset.

Table 3-11. DRAM Refresh Memory Map Table

Address	Name	Width	Description
Base + 570 #	DRAM-High	Word	Dynamic RAM High Address and FC
Base + 572 #	DRAM-Low	Word	Dynamic RAM Low Address
Base + 574 #	INCREMENT	Word	Increment Step (number of bytes/row)
Base + 576 #	COUNT	Word	RAM Refresh Cycle Count (number of rows)
Base + 578	T-ptr-H	Word	Temporary Refresh High Address and FC
Base + 57A	T-ptr-L	Word	Temporary Refresh Low Address
Base + 57C #	T-count	Word	Temporary Refresh Cycles Count
Base + 57E	RESERVED	Word	Reserved

Initialized by the user (M68000 core).

DRAM_High—Dynamic RAM High Address and Function Codes



This 16-bit parameter contains the dynamic RAM address space function code output during the refresh cycle and the high eight bits of the dynamic RAM starting address. This parameter should be initialized by the user before activating the refresh routine.

NOTE

The FC bits should not be programmed to the value “111.”

three SCC3 functions ($\overline{CTS3}$, $\overline{RTS3}$, and $\overline{CD3}$) can be routed to replace the three SCP pins or else not used.
In NMSI mode, the MSC2 and MSC3 bits are ignored. The choice of general-purpose I/O port pins versus SCC2 and SCC3 functions is made in the port A control register. See 3.3 Parallel I/O Ports for an example and more information. The choice of SCP pins versus three SCC3 functions is made in the SPMODE register in the SCP. See 4.6 Serial Communication Port (SCP) for more details.

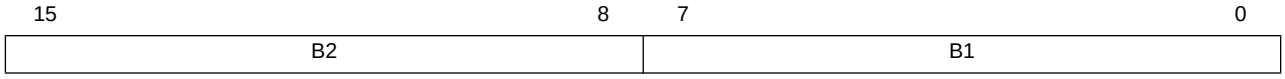
01 = PCM Mode
When working in PCM mode, each of the three multiplexed channels CH-1, CH-2, and CH-3 can be routed independently to each of the three SCCs. This connection is determined by the DRB, DRA, B1RB, B1RA, B2RB, and B2RA bits. SCC2 and SCC3 can be connected directly to their respective NMSI pins (if they are not needed for the PCM channels) as determined by the MSC3–MSC2 bits. In the NMSI case, the choice still exists for port/SCP functions versus SCC functions as described in case 00. The MSC3–MSC2 bits override the PCM routing for a specific SCC.

10 = IDL Mode
When working in IDL/GCI mode, each ISDN channel (D, B1, and B2) can be routed independently to each of the three SCCs. This connection is determined by the DRB, DRA, B1RB, B1RA, B2RB, and B2RA bits. SCC2 and SCC3 can be connected directly to their respective NMSI pins (if they are not needed for ISDN channels) determined by the MSC3–MSC2 bits. In the NMSI case, the choice still exists for port/SCP functions versus SCC functions as described in case 00. Note that the MSC3–MSC2 bits override the ISDN connection for a specific SCC.

11 = GCI Interface
Refer to the IDL mode description.

4.4.5.2 Serial Interface Mask Register (SIMASK)

The SIMASK register, a memory-mapped read-write register, is set to all ones by reset. SIMASK is used in IDL and GCI to determine which bits are active in the B1 and B2 channels. Any combination of bits may be chosen. A bit set to zero is not used by the IMP. A bit set to one signifies that the corresponding B channel bit is used for transmission and reception on the B channel. Note that the serial data strobes, SD1 and SD2, are asserted for the entire 8-bit time slot independent of the setting of the bits in the SIMASK register.

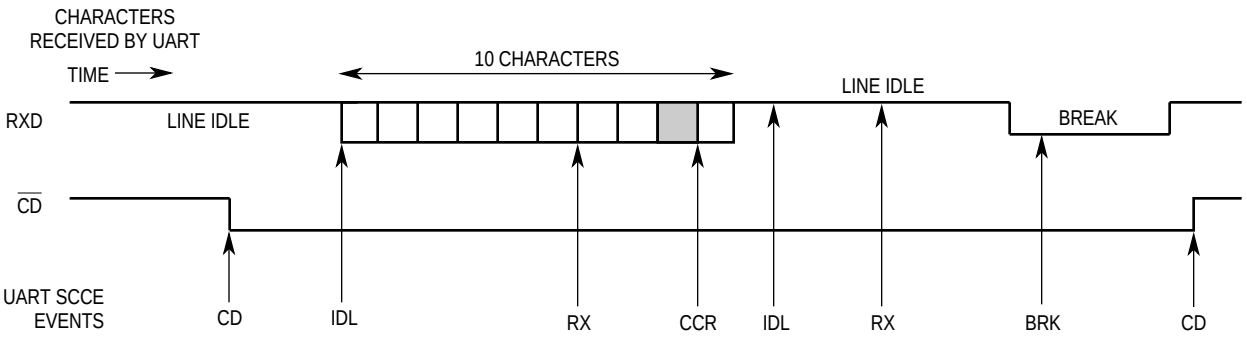


NOTE

Bit 0 of this register is the first bit transmitted or received on the IDL/GCI B1 channel.

4.5 SERIAL COMMUNICATION CONTROLLERS (SCCS)

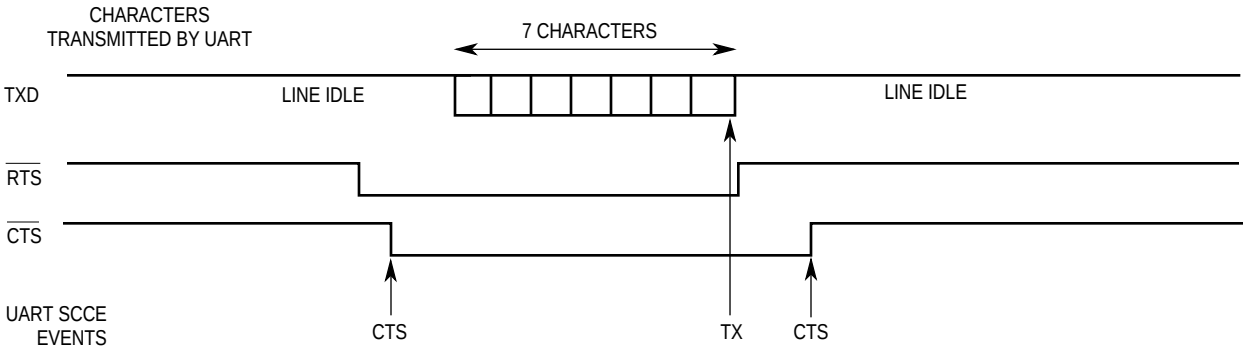
The IMP contains three independent SCCs, each of which can implement different protocols. This configuration provides the user with options for controlling up to three independent full-duplex lines implementing bridges or gateway functions or multiplexing up to three SCCs onto the same physical layer interface to implement a 2B + D ISDN basic rate channel or



NOTES:
1. The first RX event assumes receive buffers are six bytes each.
2. The second IDL event occurs after 9 to 13 ones received in a row.
3. The second RX event position is programmable based on the max_IDL value.
4. The BRK event occurs after the first break character is received.

LEGEND:

Is a receive control character defined not to be stored in the receive buffer.



NOTE: TX event assumes all seven characters were put into a single buffer.

Figure 4-23. UART Interrupt Events Example

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

CTS—Clear-To-End Status Changed

A change in the status of the $\overline{\text{CTS}}$ line was detected on the UART 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 $\overline{\text{CD}}$ line was detected on the UART channel. The SCC status register may be read to determine the current status.

IDL—IDLE Sequence Status Changed

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

Bits 11–6—Reserved for future use; should be written with zero.

COMMON SCC MODE BITS—See 4.5.3 SCC Mode Register (SCM) for a description of the DIAG1, DIAG0, ENR, ENT, MODE1, and MODE0 bits.

4.5.16.8 Transparent Receive Buffer Descriptor (RxBD)

The CP reports information about the received data for each buffer using BD. The Rx BD is shown in Figure 4-42. The CP closes the current buffer, generates a maskable interrupt, and starts to receive data into the next buffer after one of the following events:

- Detecting an error
- Detecting a full receive buffer
- Issuing the ENTER HUNT MODE command

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

Figure 4-42. Transparent Receive Buffer Descriptor

The first word of the Rx BD contains control and status bits.

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 this 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 CP. After it sets this bit, the M68000 core should not write to any fields of this BD when this bit is set. The empty bit will remain set while the CP 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 CP will receive incoming data into the first BD in the table. Setting this bit allows the use of fewer than eight BDs to conserve internal RAM.

NOTE

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

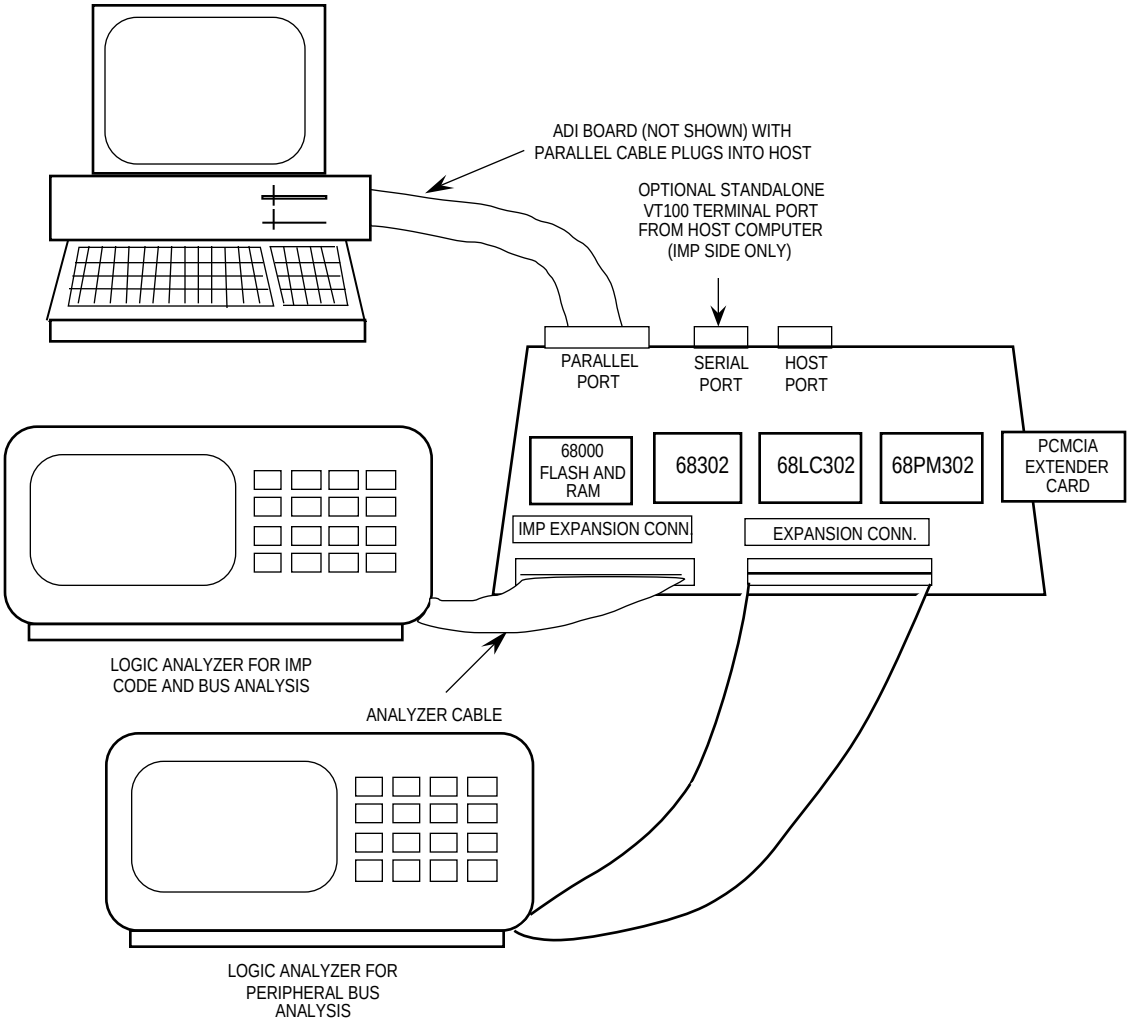


Figure B-2. MC68302FADS

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.

The interconnection between the MC68302 SCP and the MC145475 is straightforward. The MC68302 is an SCP master device and will generate the clock timing and the SCP_ENABLE for the SCP transaction. The SCP_EN signal is driven by a parallel output pin and must be handled by software (asserted for each transaction between the MC68302 and the MC145475).

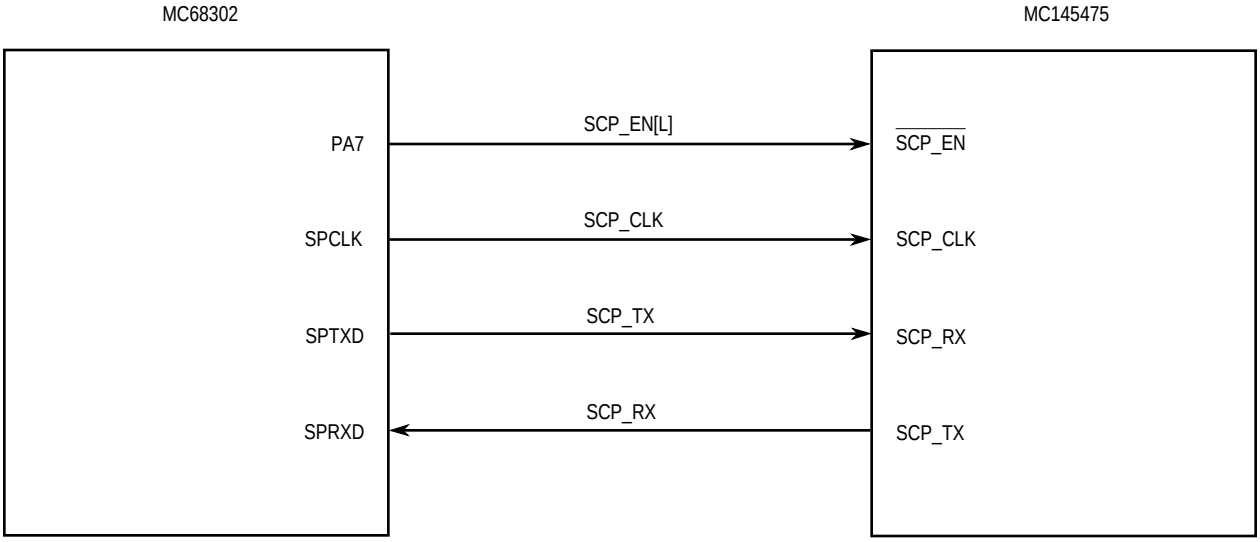


Figure D-16. SCP Bus Interconnection

D.6.11 SCP Configuration

The SCP is configured by setting the SCP mode register. Setting the SCP mode register to $2F will configure the SCP to the clock invert mode (clock is idle high, transmitted data is shifted on falling edges and received bits are sampled on rising edges), and the internal baud rate generator will divide the system clock by 32.

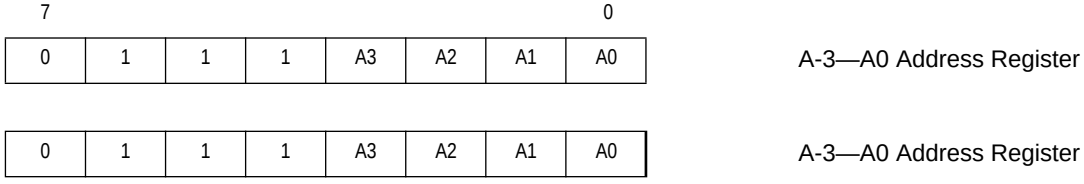
Each data transaction is triggered by setting the start bit high (after software has configured the SCP buffer descriptor).

D.6.12 SCP Data Transactions

The MC145475 has two sets of read/write programmable registers. One is the nibble (4 bits) register set and the other is the byte (8 bits) register set.

The registers are set to default value on reset. Read/write operation is made via the SCP channel.

A byte register write is made by writing two bytes with the following format:



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).

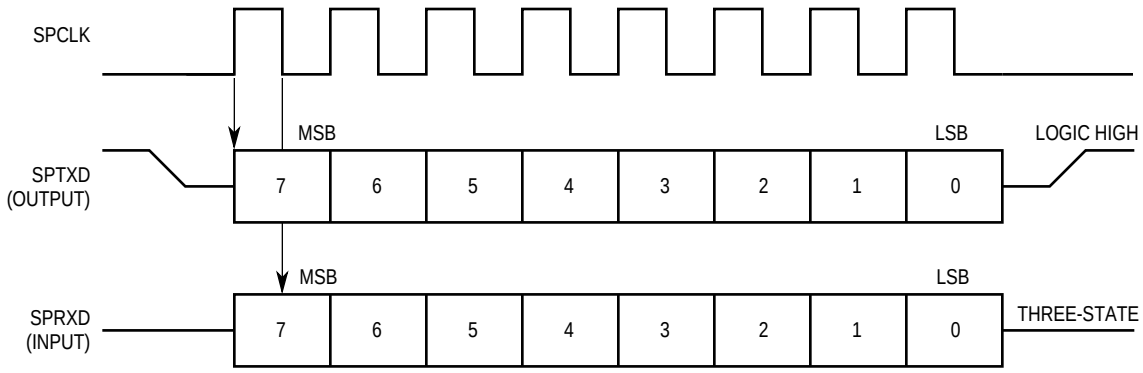
D.8 USING THE MC68302 TRANSPARENT MODE

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

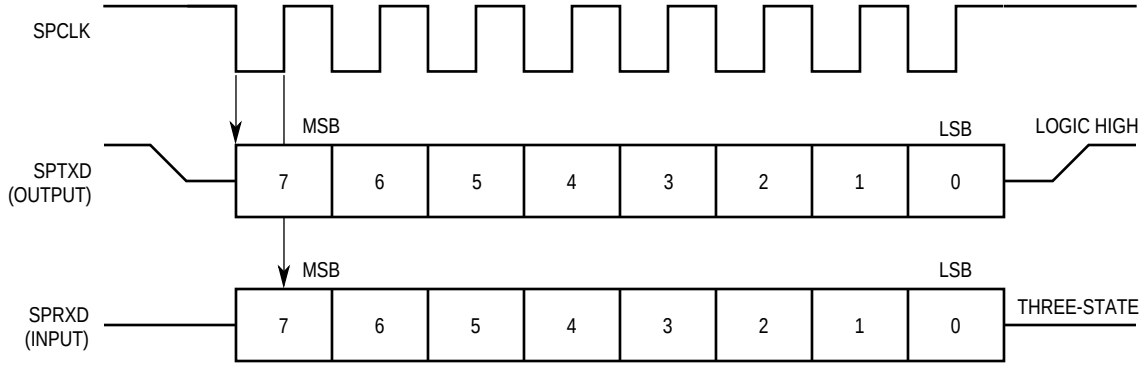
Transparent mode allows the transmission and reception of serial data over a serial communications controller (SCC) without any modification to that data stream. Contrast this with HDLC mode, where zero bits are occasionally inserted during transmission and stripped during reception, and with UART mode, 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



NOTE: Transmitted data bits shift on rising edges; received bits are sampled on falling edges.

(a) CI = 0



NOTE: Transmitted data bits shift on falling edges; received bits are sampled on rising edges.

(b) CI = 1

Figure D-31. SCP Timing

D.9 AN APPLETALK® NODE WITH THE MC68302 AND MC68195

The following paragraphs describe a hardware design that uses the MC68195 LocalTalk Adaptor (LA) to interface the MC68302 to AppleTalk. The LA is designed to work directly with the MC68302 for this purpose. The design is also suitable to those wishing to build a proprietary HDLC-based LAN.

The design as shown works with a set of LocalTalk chip drivers, that allow frames to be sent and received on the network. If these drivers are to be used, the interface between the LA and the MC68302 must be identical to that shown in Figure D-32.

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.

X—External Buffer

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

W—Wrap (final BD in table)

0 = This is not the last BD in the transmit BD table.
1 = This is the last BD in the transmit BD table.

I—Interrupt

0 = 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.

CR—Clear-to-Send Report

0 = The buffer following this buffer, if ready, will be transmitted with no delay; less precise CT bit reporting.
1 = Normal CTS lost (CT bit) error reporting, and two bits of idle occur between back-to-back buffers.

A—Address

0 = This buffer contains data only.
1 = This buffer contains address character(s) only.

P—Preamble

0 = No preamble sequence is sent.
1 = The UART sends a preamble sequence (set of 9 to 13 bits) before sending the data.

Bits 8–1—Reserved for future use

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.2.1.5.2 Transmit Buffer Data Length. This 16-bit value is written by the user to indicate the number of octets to be transmitted from the data butter.

E2.1.5.3 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.2.2 Programming the SCC for UART

This section gives a generic algorithm for programming an SCC to handle UART. 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—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.

W

X