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	Active
Core Processor	M68000
Number of Cores/Bus Width	1 Core, 8/16-Bit
Speed	33MHz
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/pro/item?MUrl=&PartUrl=mc68302ag33c

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

(DAPR), an 8-bit function code register (FCR), a 16-bit byte count register (BCR), a 16-bit channel mode register (CMR), and an 8-bit channel status register (CSR). These registers provide the addresses, transfer count, and configuration information necessary to set up a transfer. They also provide a means of controlling the IDMA and monitoring its status. All registers can be modified by the M68000 core. The IDMA also includes another 16-bit register, the data holding register (DHR), which is not accessible to the M68000 core and is used by the IDMA for temporary data storage.

3.1.2.1 Channel Mode Register (CMR)

The CMR, a 16-bit register, is reset to $0000.

15	14	13	12	11 10	9	8	7 6	5 4	3 2	1	0
—	ECO	INTN	INTE	REQG	SAPI	DAPI	SSIZE	DSIZE	BT	RST	STR

Bit 15—Reserved for future use.

ECO—External Control Option

0 = If the request generation is programmed to be external in the REQG bits, the control signals ($\overline{\text{DACK}}$ and $\overline{\text{DONE}}$) are used in the source (read) portion of the transfer since the peripheral is the source.

1 = If the request generation is programmed to be external in the REQG bits, the control signals ($\overline{\text{DACK}}$ and $\overline{\text{DONE}}$) are used in the destination (write) portion of the transfer since the peripheral is the destination.

INTN—Interrupt Normal

0 = When the channel has completed an operand transfer without error conditions as indicated by $\overline{\text{DONE}}$, the channel does not generate an interrupt request to the IMP interrupt controller. The DONE bit remains set in the CSR.

1 = When the channel has completed an operand transfer without error conditions as indicated by $\overline{\text{DONE}}$, the channel generates an interrupt request to the IMP interrupt controller and sets DONE in the CSR.

NOTE

An interrupt will only be generated if the IDMA bit is set in the interrupt mask register (IMR).

INTE—Interrupt Error

0 = If a bus error occurs during an operand transfer either on the source read (BES) or the destination write (BED), the channel does not generate an interrupt to the IMP interrupt controller. The appropriate bit remains set in the CSR.

1 = If a bus error occurs during an operand transfer either on BES or BED, the channel generates an interrupt to the IMP interrupt controller and sets the appropriate bit (BES or BED) in the CSR.

ET7—$\overline{\text{IRQ7}}$ Edge-/Level-Triggered

This bit is valid only in the dedicated mode.

0 = Level-triggered. An interrupt is made pending when $\overline{\text{IRQ7}}$ is low.

NOTE

The M68000 always treats level 7 as an edge-sensitive interrupt. Normally, users should not select the level-triggered option. The level-triggered option is useful when it is desired to make the negation of $\overline{\text{IRQ7}}$ cause the IOUT2–IOUT0 pins to cease driving a level 7 interrupt request when the MC68302 is used in the disable CPU mode. This situation is as follows:

For a slave-mode MC68302, when it is triggered by $\overline{\text{IRQ1}}$, $\overline{\text{IRQ6}}$, or $\overline{\text{IRQ7}}$ to generate an interrupt, its interrupt controller will output the interrupt request on pins IOUT2–IOUT0 to another processor (MC68302, MC68020, etc.) For cases when the slave MC68302 does not generate a level 4 vector (i.e., the VGE bit is cleared), one must set the ET1, ET6, and ET7 bits to level-triggered and then negate the $\overline{\text{IRQ1}}$, $\overline{\text{IRQ6}}$, and $\overline{\text{IRQ7}}$ lines externally in the interrupt handler code. If the ET1, ET6, and ET7 bits are set to edge-triggered and the VGE bit is clear, the IOUT2–IOUT0 pins will never be cleared.

1 = Edge-triggered. An interrupt is made pending when $\overline{\text{IRQ7}}$ changes from one to zero (falling edge).

ET6—IRQ6 Edge-/Level-Triggered

This bit is valid only in the dedicated mode.

0 = Level-triggered. An interrupt is made pending when $\overline{\text{IRQ6}}$ is low.

NOTE

While in disable CPU mode during the host processor interrupt acknowledge cycle for $\overline{\text{IRQ6}}$, if $\overline{\text{IRQ6}}$ is not continuously asserted, the interrupt controller will still provide the vector number (and $\overline{\text{DTACK}}$) according to the IV6 bit. The $\overline{\text{IACK6}}$ falling edge may be used externally to negate $\overline{\text{IRQ6}}$.

1 = Edge-triggered. An interrupt is made pending when $\overline{\text{IRQ6}}$ changes from one to zero (falling edge).

ET1—$\overline{\text{IRQ1}}$ Edge-/Level-Triggered

This bit is valid only in the dedicated mode.

0 = Level-triggered. An interrupt is made pending when $\overline{\text{IRQ1}}$ is low.

Port A Control Register(PACNT

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

0 = I/O 1 = Peripheral

Port A Data Direction Register(PADDR)

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

0 = Input 1 = Output

Port A Data Register(PADAT)

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

Port B Control Register(PBCNT)

15	8	7	6	5	4	3	2	1	0
RESERVED		CB	CB	CB	CB	CB	CB	CB	CB

0 = I/O 1 = Peripheral

Port B Data Direction Register(PBDDR)

15	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		DB	DB	DB	DB	DB	DB	DB	DB	DB	DB	DB	DB

0 = Input 1 = Output

Port B Data Register(PBDAT)

15	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PB	PB	PB	PB	PB	PB	PB	PB	PB	PB	PB	PB

Figure 3-6. Parallel I/O Port Registers

3.4 DUAL-PORT RAM

The CP has 1152 bytes of static RAM configured as a dual-port memory. The dual-port RAM can be accessed by the CP main controller or by one of three bus masters: the M68000 core, the IDMA, or an external master. The M68000 core and the IDMA access the RAM synchro-

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.

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

SDC2—Serial Data Strobe Control 2

0 = SDS2 signal is asserted during the B2 channel
1 = SDS1 signal is asserted during the B2 channel

SDC1—Serial Data Strobe Control 1

0 = SDS1 signal is asserted during the B1 channel
1 = SDS2 signal is asserted during the B1 channel

B2RB, B2RA—B2 Channel Route in IDL/GCI Mode or CH-3 Route in PCM Mode

00 = Channel not supported
01 = Route channel to SCC1
10 = Route channel to SCC2 (if MSC2 is cleared)
11 = Route channel to SCC3 (if MSC3 is cleared)

B1RB, B1RA—B1 Channel Route in IDL/GCI Mode or CH-2 Route in PCM Mode

00 = Channel not supported
01 = Route channel to SCC1
10 = Route channel to SCC2 (if MSC2 is cleared)
11 = Route channel to SCC3 (if MSC3 is cleared)

DRB, DRA—D-Channel Route in IDL/GCI Mode or CH-1 Route in PCM Mode

00 = Channel not supported
01 = Route channel to SCC1
10 = Route channel to SCC2 (if MSC2 is cleared)
11 = Route channel to SCC3 (if MSC3 is cleared)

MSC3—SCC3 Connection

0 = SCC3 is connected to the multiplexed serial interface (PCM, IDL, or GCI) chosen in MS1–MS0. NMSI3 pins are all available for other purposes.
1 = SCC3 is not connected to a multiplexed serial interface but is connected directly to the NMSI3 pins or SCP pins or is not used. The choice of general-purpose I/O port pins versus SCC3 functions is made in the port A control register. The choice of SCP pins versus SCC3 functions is made in the SPMODE register.

MSC2—SCC2 Connection

0 = SCC2 is connected to the multiplexed serial interface (PCM, IDL, or GCI) chosen in MS1–MS0. NMSI2 pins are all available for other purposes.
1 = SCC2 is not connected to a multiplexed serial interface but is either connected directly to the NMSI2 pins or not used. The choice of general-purpose I/O port pins versus SCC2 functions is made in the port A control register.

MS1—MS0—Mode Supported

00 = NMSI Mode
When working in NMSI mode, SCC1 is connected directly to the seven NMSI1 pins (RXD1, TXD1, RCLK1, TCLK1, $\overline{\text{CD1}}$, $\overline{\text{CTS1}}$, and $\overline{\text{RTS1}}$). SCC2 functions can be routed to port A as NMSI functions or configured instead as PA6–PA0. Four of the SCC3 functions can be routed to port A or retained as PA11–PA8. The other

CD10–CD0—Clock Divider

The clock divider bits and the prescaler determine the baud rate generator output clock rate. CD10–CD0 are used to preset an 11-bit counter that is decremented at the prescaler output rate. The counter is not otherwise accessible to the user. When the counter reaches zero, it is reloaded with the clock divider bits. Thus, a value of $7FF in CD10–CD0 produces the minimum clock rate (divide by 2048); a value of $000 produces the maximum clock rate (divide by 1).

NOTE

Because of SCC clocking restrictions, the maximum baud rate that may be used to clock an SCC is divide by 3.

When dividing by an odd number, the counter ensures a 50% duty cycle by asserting the terminal count once on a clock high and next on a clock low. The terminal count signals the counter expiration and toggles the clock.

DIV4—SCC Clock Prescaler Divide by 4

The SCC clock prescaler bit selects a divide-by-1 (DIV4 = 0) or divide-by-4 (DIV4 = 1) prescaler for the clock divider input. The divide-by-4 option is useful in generating very slow baud rates.

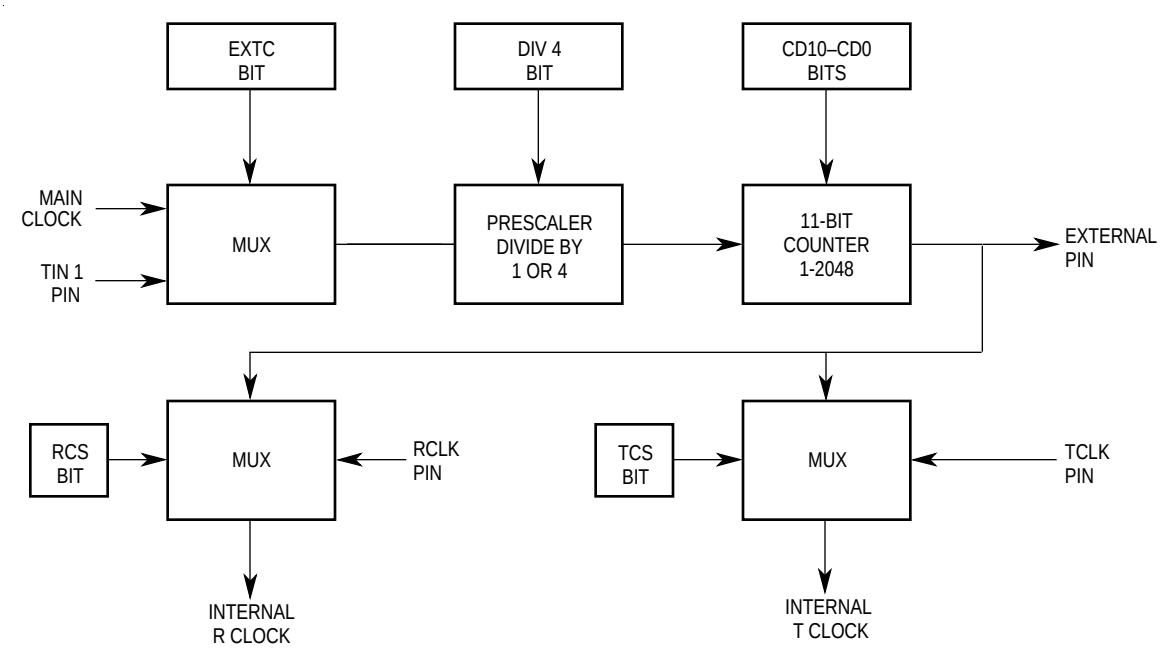


Figure 4-12. SCC Baud Rate Generator

4.5.2.1 Asynchronous Baud Rate Generator Examples

The UART circuitry always uses a clock that is 16x the baud rate. The ratio of the 16x UART clock to the system parallel clock must not exceed 1:2.5. For an internally supplied clock, an integer divider value must be used; therefore, the divider must be 3 or greater. Thus, using a clock divider value of 3 (programmed as 2 in the SCON) and a 16.67-MHz crystal gives a UART clock rate of 5.56 MHz and a baud rate of 347 kbaud. Assuming again a 16.67-MHz

for an internal clock, TCS and RCS may both be zero, or, for an external clock, they may both be one. The other two combinations are not allowed in this mode.

NOTE

If external loopback is desired (i.e., external to the MC68302), then the DIAG1–DIAG0 bits should be set for either normal or software operation, and an external connection should be made between the TXD and RXD pins. Clocks may be generated internally, externally, or an internally generated TCLK may be externally connected to RCLK. If software operation is used, the $\overline{\text{RTS}}$, $\overline{\text{CD}}$, and $\overline{\text{CTS}}$ pins need not be externally connected. If normal operation is used, the $\overline{\text{RTS}}$ pin may be externally connected to the $\overline{\text{CD}}$ pin, and the $\overline{\text{CTS}}$ pin may be grounded.

NOTE

Do not use this mode for loopback operation of IDL in the Serial Interface. Instead program the diag bits to Normal Operation, and (1) assert the L1GR pin externally from the S/T chip, or (2) configure the SDIAG1-0 bits in the SIMODE to Internal Loopback or Loopback Control.

10 = Automatic echo

In this mode, the channel automatically retransmits the received data on a bit-by-bit basis. The receiver operates normally, but the transmitter simply retransmits the received data. The $\overline{\text{CD}}$ pin must be asserted for the receiver to receive data, and the $\overline{\text{CTS}}$ line is ignored. The data is echoed out the TXD pin with a few nanosecond delay from RXD. No transmit clock is required, and the ENT bit in the SCC mode register does not have to be set.

NOTE

The echo function may also be accomplished in software by receiving buffers from an SCC, linking them to transmit buffer descriptors, and then transmitting them back out of that SCC.

11 = Software operation (CTS, CD lines under software control)

In this mode, the CTS and CD lines are just inputs to the SCC event (SCCE)and status (SCCS) registers. The SCC controller does not use these lines to enable/disable reception and transmission, but leaves low (i.e., active) in this mode. Transmission delays from RTS low are zero TCLKs (asynchronous protocols) or one TCLK (synchronous protocols).

NOTE

The MC68302 provides several tools for enabling and disabling transmission and/or reception. Choosing the right tool is application and situation dependent. For the receiver, the tools are 1) the empty bit in the receive buffer descriptor, 2) the ENR bit, and 3) the ENTER HUNT MODE command. For the transmitter, the

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 = The TX bit in the UART event register will be set when this buffer has been serviced by the CP, which can cause an interrupt.

Bits 8–1—Reserved for future use.

CR—Clear-to-Send Report

This bit allows a choice of no delay between buffers transmitted in UART mode, versus a more accurate CTS lost error reporting and two bits of idle between buffers.

0 = The buffer following this buffer will be transmitted with no delay (assuming it is ready), but the CT bit may not be set in the correct Tx BD, or may not be set at all in a CTS lost condition. The user is advised to monitor the CTS bit in the UART event register for an indication of CTS lost, in addition to the CT bits in the Tx BDs. The CTS bit will always be set properly.
1 = Normal CTS lost (CT bit) error reporting, and two bits of idle occur between back-to-back buffers.

If the DIAG1–DIAG0 bits in the SCM are set to software operation (rather than normal operation), then this bit only affects the delay between buffers, not the CTS reporting, and would normally be set to zero.

A—Address

This bit is valid only in multidrop mode (UM0 = 1).
0 = This buffer contains data only.
1 = Set by the M68000 core, this bit indicates that this buffer contains address character(s). All the buffer's data will be transmitted as address characters.

P—Preamble

0 = No preamble sequence is sent.
1 = The UART sends one preamble sequence (9 to 13 ones) before sending the data.

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

CT—CTS Lost

0 = The $\overline{\text{CTS}}$ signal remained active during transmission.
1 = The $\overline{\text{CTS}}$ signal was negated during transmission.

Data Length

The data length is the number of octets that the CP should transmit from this BD's data buffer. It is never modified by the CP. This value should be normally greater than zero. The data length may be equal to zero with the P bit set, and only a preamble will be sent.

Tx Buffer Pointer

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

NOTE

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

4.5.11.16 UART Event Register

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

The UART 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 clear the internal interrupt request. This register is cleared at reset.

An example of the timing of various events in the UART event register is shown in Figure 4-23.

UN—Underrun

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

CT—CTS Lost

CTS in NMSI mode or L1GR (layer-1 grant) in IDL/GCI mode was lost during frame transmission. If data from more than one buffer is currently in the FIFO when this error occurs, this bit will be set in the Tx BD that is currently open.

Data Length

The data length is the number of octets the HDLC controller should transmit from this BD's data buffer. It is never modified by the CP. The value of this field should be greater than zero.

Tx Buffer Pointer

The transmit buffer pointer, which contains the address 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.

4.5.12.12 HDLC Event Register

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

The HDLC 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 clear the internal interrupt request. This register is cleared at reset.

An example of the timing of various events in the HDLC event register is shown in Figure 4-29.

nized by the BISYNC channel and to generate interrupts. On recognition of an event, the BISYNC controller sets the corresponding bit in the BISYNC event register. Interrupts generated by this register may be masked in the BISYNC mask register.

The BISYNC 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 BISYNC 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 BISYNC 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 character has been received and written to the buffer.

BSY—Busy Condition

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

TX—Tx Buffer

A buffer has been transmitted. This bit is set on the second to last bit of BCC or data.

RX—Rx Buffer

A complete buffer has been received on the BISYNC channel.

4.5.13.13 BISYNC Mask Register

The SCC mask register (SCCM) is referred to as the BISYNC mask register when the SCC is operating as a BISYNC controller. It is an 8-bit read-write register that has the same bit format as the BISYNC event register. If a bit in the BISYNC mask register is a one, the corresponding interrupt in the event register will be enabled. If the bit is zero, the corresponding interrupt in the event register will be masked. This register is cleared upon reset.

NOTE

This technique is not valid for the PCM envelope sync method when the time slots are less than six bits in length. In such a case, the user may clear EXSYN to cause transmission to begin, and then, for the receiver, provide the required SYN1–SYN2 sequence. To accomplish this, the user may configure the SCC to loopback mode with the EXSYN bit cleared and the DSR set to $FFFF. Then after 16 serial clocks, the receiver and transmitter are synchronized, and the SCC may be dynamically reconfigured to normal or software operation. At this point, reception begins immediately, and transmission begins after the transmit BD is made ready.

5. With the physical interface configured for IDL or GCI mode, the SCC may be configured with the EXSYN bit set and the DIAG1–DIAG0 bits set to either software operation or normal operation. In this case, the data will be byte-aligned to the B or D channel time slots.

Once synchronization is achieved for the transmitter, it will remain in effect until an error occurs, a STOP TRANSMIT command is given, or a buffer has completed transmission with the Tx BD last (L) bit set. Once synchronization is achieved for the receiver, it will remain in effect until an error occurs or the ENTER HUNT MODE command is given.

4.5.16.6 Transparent Error-Handling Procedure

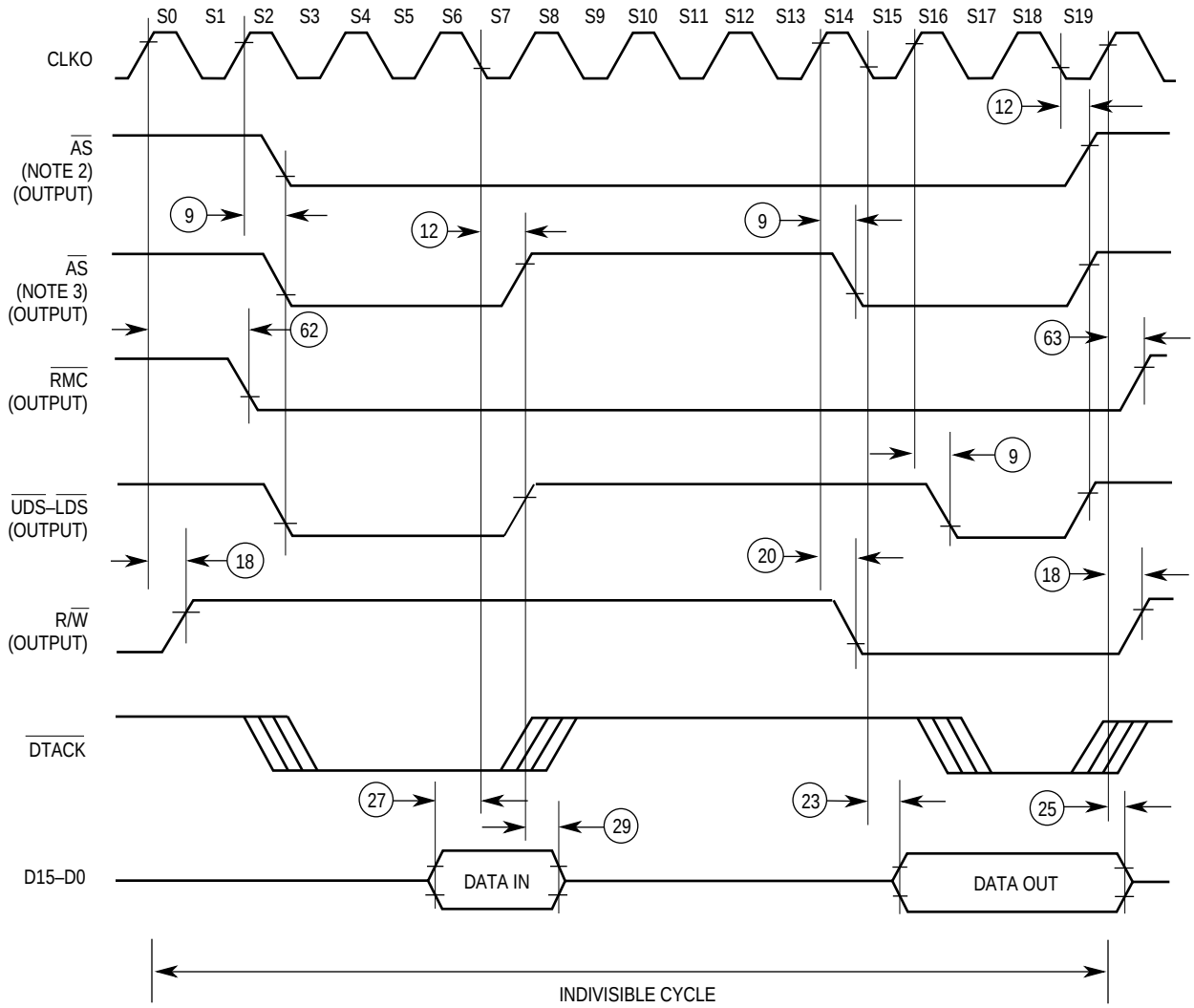
The transparent controller reports message reception and transmission error conditions using the channel BDs and the transparent event register. The modem interface lines can also be directly monitored in the SCC status register.

Transmission Errors:

1. Transmitter Underrun—When this error occurs, the channel terminates buffer transmission, closes the buffer, sets the underrun (UN) bit in the BD, and generates the TXE interrupt (if enabled). The channel resumes transmission after the reception of the RESTART TRANSMIT command. Underrun can occur after a transmit frame for which the L bit in the Tx BD was not set. In this case, only the TXE bit is set. The FIFO size is four words in transparent mode.
2. Clear-To-Send Lost During Message Transmission—When this error occurs and the channel is not programmed to control this line with software, the channel terminates buffer transmission, closes the buffer, sets the CTS lost (CT) bit in the BD, and generates the TXE interrupt (if enabled). The channel will resume transmission after the reception of the RESTART TRANSMIT command.

Reception Errors:

1. Overrun Error—The transparent controller maintains an internal three-word FIFO for receiving data. The CP begins programming the SDMA channel (if the data buffer is in external memory) when the first word is received into the FIFO. If a FIFO overrun occurs, the transparent controller writes the received data word to the internal FIFO over the previously received word. The previous word is lost. Next, the channel closes



NOTES:
1. For other timings than RMC, see Figures 6-2 and 6-3.
2. RMCST = 0 in the SCR.
3. RMCST = 1 in the SCR.
4. Wait states may be inserted between S4 and S5 during the write cycle and between S16 and S17 during the read cycle.
5. Read-modify-write cycle is generated only by the TAS instruction.

Figure 6-4. Read-Modify-Write Cycle Timing Diagram

6.11 AC ELECTRICAL SPECIFICATIONS—EXTERNAL MASTER INTERNAL SYNCHRONOUS READ/WRITE CYCLES

(see Figure 6-10, Figure 6-11, and Figure 6-12)

			16.67 MHz		20 MHz		25 MHz		
Num.	**Characteristic**	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**
110	Address Valid to $\overline{AS}$ Low	t_{AVASL}	15	—	12	—	10	—	ns
111	$\overline{AS}$ Low to Clock High	t_{ASLCH}	30	—	25	—	20	—	ns
112	Clock Low to $\overline{AS}$ High	t_{CLASH}	—	45	—	40	—	30	ns
113	$\overline{AS}$ High to Address Hold Time on Write	t_{ASHAH}	0	—	0	—	0	—	ns
114	$\overline{AS}$ Inactive Time	t_{ASH}	1	—	1	—	1	—	clk
115	$\overline{UDS}$/$\overline{LDS}$ Low to Clock High (see Note 2)	t_{SLCH}	40	—	33	—	27	—	ns
116	Clock Low to $\overline{UDS}$/$\overline{LDS}$ High	t_{CLSH}	—	45	—	40	—	30	ns
117	R/$\overline{W}$ Valid to Clock High (see Note 2)	t_{RWVCH}	30	—	25	—	20	—	ns
118	Clock High to R/$\overline{W}$ High	t_{CHRWH}	—	45	—	40	—	30	ns
119	$\overline{AS}$ Low to IAC High	t_{ASLIAH}	—	40	—	35	—	27	ns
120	$\overline{AS}$ High to IAC Low	t_{ASHIAL}	—	40	—	35	—	27	ns
121	$\overline{AS}$ Low to $\overline{DTACK}$ Low (0 Wait State)	t_{ASLDTL}	—	45	—	40	—	30	ns
122	Clock Low to $\overline{DTACK}$ Low (1 Wait State)	t_{CLDTL}	—	30	—	25	—	20	ns
123	$\overline{AS}$ High to $\overline{DTACK}$ High	t_{ASHDTH}	—	45	—	40	—	30	ns
124	$\overline{DTACK}$ High to $\overline{DTACK}$ High Impedance	t_{DTHDTZ}	—	15	—	15	—	10	ns
125	Clock High to Data-Out Valid	t_{CHDOV}	—	30	—	25	—	20	ns
126	$\overline{AS}$ High to Data High Impedance	t_{ASHDZ}	—	45	—	40	—	30	ns
127	$\overline{AS}$ High to Data-Out Hold Time	t_{ASHDOI}	0	—	0	—	0	—	ns
128	$\overline{AS}$ High to Address Hold Time on Read	t_{ASHAI}	0	—	0	—	0	—	ns
129	$\overline{UDS}$/$\overline{LDS}$ Inactive Time	t_{SH}	1	—	1	—	1	—	clk
130	Data-In Valid to Clock Low	t_{CLDIV}	30	—	25	—	20	—	ns
131	Clock Low to Data-In Hold Time	t_{CLDIH}	15	—	12	—	10	—	ns

NOTES:
1. Synchronous specifications above are valid only when SAM = 1 in the SCR.
2. It is required that this signal not be asserted prior to the previous rising CLKO edge (i.e., in the previous clock cycle). It must be recognized by the IMP no sooner than the rising CLKO edge shown in the diagram.

channel was always SCC1.

7. This data applies to MC68302 masks 2B14M, 3B14M, or later.
8. The following explanation concerns a fast HDLC channel and two slower channels: When the fast HDLC is 1:9, two HDLCs can run at 1:224. Thus, with a 16.67-MHz dock, SCC1 can run at 1.85 Mbps; SCC2 and SCC3 can run at 74 kbps. Two HDLCs can also run without equal values: one at 1:128 and one at 1:238. When the fast HDLC is 1:10, two HDLCs can run at 1:128. When the fast HDLC is 1:9, two UARTs can run at 1:396 (*16). When the fast HDLC is 1:10, two UARTs can run at 1:10 (*16).
9. Performance results above showed no receive overruns or transmit underruns in several minutes of continuous transmission/reception. Reduction of the above ratios by a single value (e.g., 1:35 reduced to 1:34) did show an overrun or underrun within several minutes.
10. All results assume the DRAM refresh controller is not operating; otherwise, performance is slightly reduced.
11. Unless specifically stated, all table results assume continuous full-duplex operation. Results for half-duplex were not measured, but will be roughly 2x better.

Since operation at very high data rates is characteristic of HDLC-framed channels rather than BISYNC-, DDCMP-, or async-framed channels, the user can also use the MC68302 in conjunction with either the Motorola MC68605 1984 CCITT X.25 LAPB controller, the MC68606 CCITT Q.921 multilink LAPD controller, or the MC145488 dual data link controller. These devices fully support operation at T1/CEPT rates (and above) and can operate with their serial clocks "gated" onto subchannels of such an interface. These devices are full M68000 bus masters. The MC68605 and MC68606 perform the full data-link layer protocol as well as support various transparent modes within HDLC-framed operation. The MC145488 provides HDLC-framed and totally transparent operation on two full-duplex channels.

initialization corresponds to the recommended order described in 4.5.7 SCC Initialization. The second part is a set of loops waiting for data to arrive to be retransmitted out of the SCC3. The third part is the SCC3 receive interrupt handler. Transmit interrupts are masked in this example.

D.4.2 Organization of Buffers

In the MC68302, there is no such thing as an receive register (Rx) or transmit register (Tx). Rather, a flexible structure called a buffer descriptor (BD) is used. In this example, two Rx BDs and two Tx BDs are used. Each BD is set up to point to a one-byte location for data. Thus, the receiver and transmitter are double-buffered. The number of Rx or Tx BDs can be changed simply by changing the number of BDs initialized in the code (and two other lines documented in the code). However, using at least two BDs has advantages as noted in the following paragraphs.

The structure of the buffers is shown in Figure D-6.

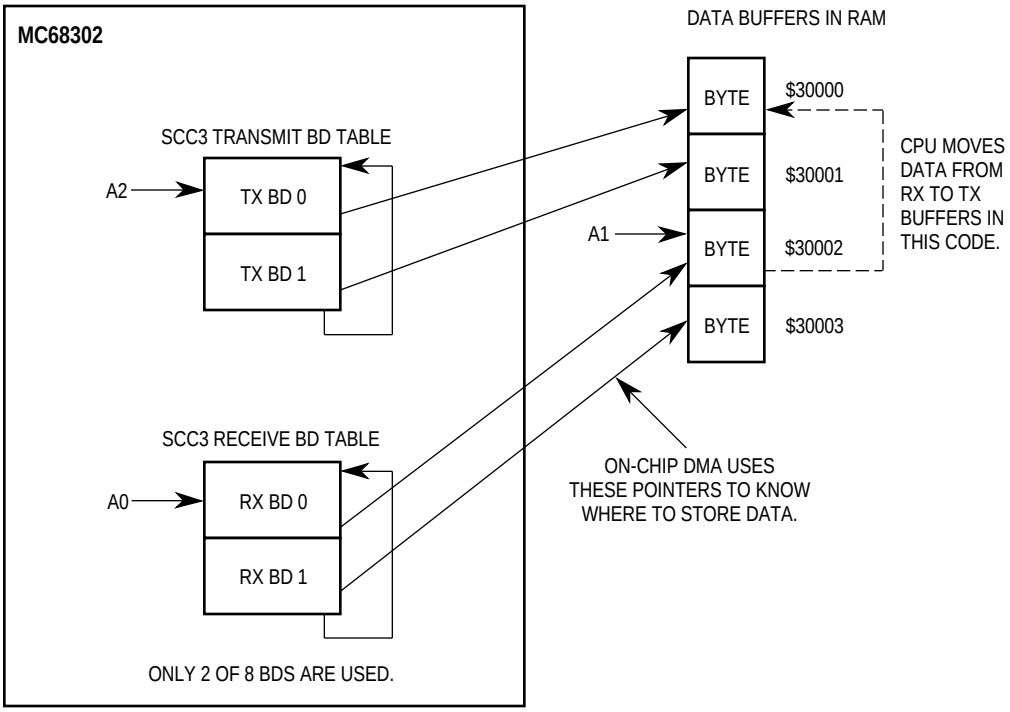


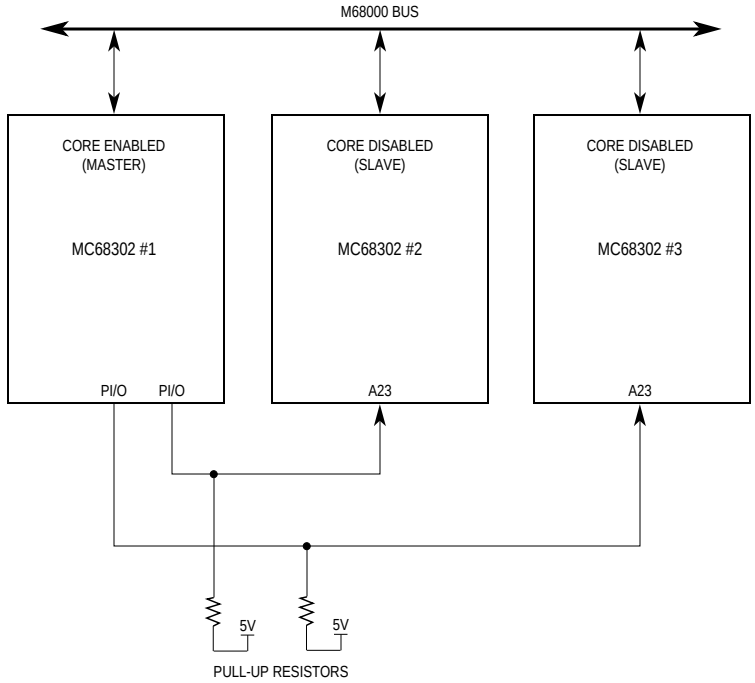
Figure D-6. Transmit and Receive BD Tables and Buffers

To make the application more general, the data buffers were located in external RAM; however, internal RAM could have been used. Each BD points to just one byte in memory as shown. Note that the data buffers do not need to be consecutive as shown.

From one to eight BDs may be used for both the transmit and receive operation. Use of eight BDs for the transmit side of SCC3 requires that the SCP and SMCs not be used. As data rates increase substantially beyond the 9600 baud of this example, the use of more BDs and more data bytes per BD becomes justified. Why were two Rx BDs and two Tx BDs chosen rather than just one each?

fully functioning MC68302s, each having an isolated bus and the ability to send data and messages between them (e.g., through a shared RAM). However, another approach is possible.

By using the MC68302 “disable CPU logic” feature, enabled with the DISCPU pin, the MC68302 can be converted into an intelligent slave peripheral that no longer has its M68000 core operating. The SDMA channels and IDMA channel request the bus externally through the bus request ($\overline{BR}$) pin. (When not in slave mode, these channels request the bus internally to the on-chip bus arbiter, with no external indication visible.) A typical slave mode example is shown in Figure D-19. A single master MC68302 (i.e., one with the M68000 core enabled) can access and control one or more slave MC68302s. (i.e., ones with the M68000 core disabled.)



NOTE: A23 is not used by the slaves.

Figure D-19. Typical Slave Mode Example

Use of the “disable CPU logic” feature in a multiple MC68302 system depends mainly on the amount of protocol processing required by the M68000 core. If the data rates are high and the amount of protocol processing required on each channel is significant, the M68000 core may be the limiting factor in communications performance. Thus, further increases in serial rates will not yield additional packets/sec performance. In such a case, a faster processor (such as the MC68020/MC68030) could be used to control all three MC68302 devices in slave mode.

The bus utilization of the SDMA channels on the three MC68302 devices is not usually a significant factor. For instance, if three SCC channels are running full duplex at 64 kbps, the respective SDMA channels consume less than 1 percent of the M68000 bus. You can calculate this figure for your design by determining how often a bus cycle to memory is required

TX—Tx Buffer

0 = No interrupt.
1 = A buffer has been transmitted on the UART channel (set only if the I bit in the Tx buffer descriptor is set).

RX—Rx Buffer

0 = No interrupt.
1 = A buffer was received on the UART channel (set only if the I bit in the Rx buffer descriptor is set).

E.2.1.2.5 UART Mask Register (SCCM). This 8-bit register is located at offset $88A (SCC1), $89A (SCC2), and $8AA (SCC3) on D15-D8 of a 16-bit data bus. The SCCM is used to enable and disable interrupt events reported by the SCCE. The mask bits correspond to the interrupt event bit shown in the SCCE. A bit should be set to one to enable the corresponding interrupt in the SCCE.

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

E.2.1.2.6 UART Status Register (SCCS). This 8-bit register is located at offset $88C (SCC1), $89C (SCC2), and $8AC (SCC3), on D15-D8 of a 16-bit data bus. The SCCS register reflects the current status of the RXD, $\overline{CD}$, and $\overline{CTS}$ lines as seen by the SCC.

7	6	5	4	3	2	1	0
—	—	—	—	—	ID	CD	CTS

ID—Idle Status on the Receiver Line (valid only when the ENR bit is set and the receive clock is running)

0 = Receiver Line is not idling.
1 = Either $\overline{CD}$ is not asserted or the receiver line is idling while $\overline{CD}$ is asserted.

$\overline{CD}$—Carrier Detect Status Changed (valid only when the ENR bit is set and the receive clock is running)

0 = $\overline{CD}$ is asserted.
1 = $\overline{CD}$ is not asserted.

$\overline{CTS}$—Clear-To-Send Status Changed (valid only when the ENT bit is set and the transmit clock is running)

0 = $\overline{CTS}$ is asserted.
1 = $\overline{CTS}$ is not asserted.

E.2.1.3 GENERAL AND UART PROTOCOL-SPECIFIC PARAMETER RAM. Each SCC has 32 words of parameter RAM used to configure receive and transmit operation, store temporary parameters for the CP, and maintain counters. The first 14 words are general parameters, which are the same for each protocol. The last 18 words are specific to the protocol selected. The following sections discuss the parameters that the user must initialize to configure the UART operation.