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/mc68302eh25c

Table Number | Title | Page Number

Section 5
Signal Description

Appendix D
MC68302 Applications

Appendix E
SCC Programming Reference

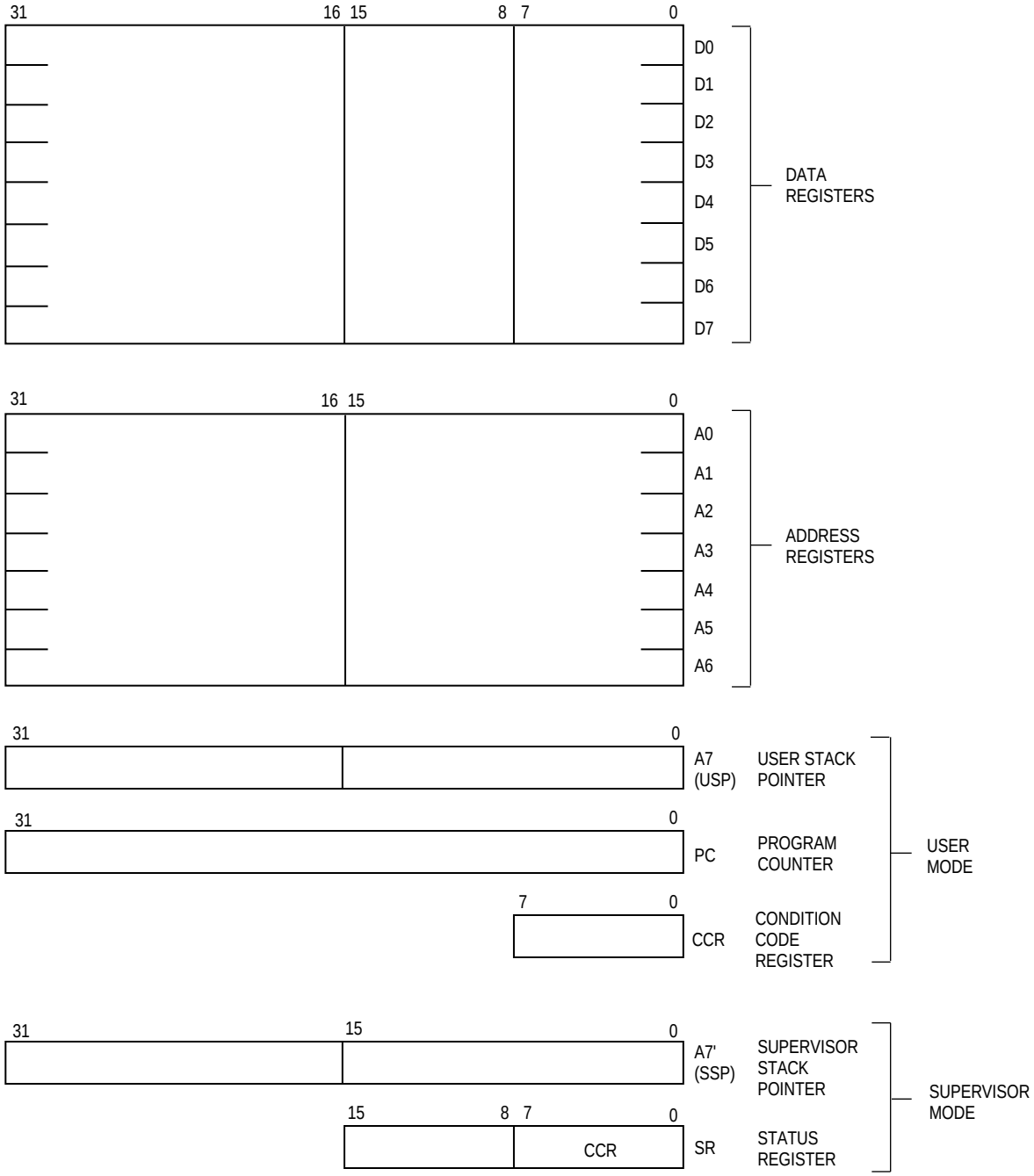


Figure 2-1. M68000 Programming Model

The supervisor's programming model includes supplementary registers, including the supervisor stack pointer (SSP) and the status register (SR) as shown in Figure 2-2. The SR contains the interrupt mask (eight levels available) as well as the following condition codes: overflow (V), zero (Z), negative (N), carry (C), and extend (X). Additional status bits indicate that the processor is in trace (T) mode and/or in a supervisor (S) state.

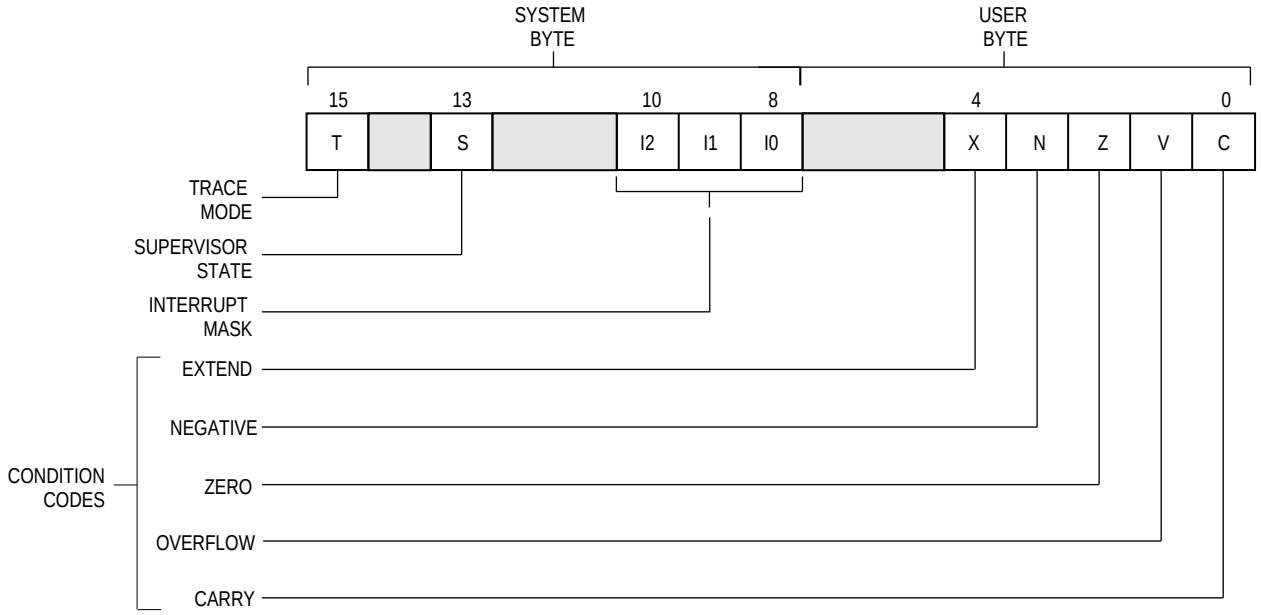


Figure 2-2. M68000 Status Register

2.2 INSTRUCTION SET SUMMARY

The five data types supported by the M68000 on the MC68302 are bits, binary-coded decimal (BCD) digits (4 bits), bytes (8 bits), words (16 bits), and long words (32 bits).

In addition, operations on other data types, such as memory addresses, status word data, etc., are provided for in the instruction set. Shown in Table 2-1, the 14 flexible addressing modes include six basic types:

- Register Direct
- Register Indirect
- Absolute
- Immediate
- Program Counter Relative
- Implied

The capability to perform postincrementing, predecrementing, offsetting, and indexing is included in the register indirect addressing modes. Program counter relative modes can also be modified via indexing and offsetting.

The M68000 instruction set is shown in Table 2-2.

Some basic instructions also have variations as shown in Table 2-3.

Special emphasis has been placed on the instruction set to simplify programming and to support structured high-level languages. With a few exceptions, each instruction operates

Table 2-9. Internal Registers

Base + 8A0	RES	16	SCC3	Reserved	
Base + 8A2	SCON3	16	SCC3	SCC3 Configuration Register	0004
Base + 8A4	SCM3	16	SCC3	SCC3 Mode Register	0000
Base + 8A6	DSR3	16	SCC3	SCC3 Data Sync. Register	7E7E
! Base + 8A8	SCCE3	8	SCC3	SCC3 Event Register	00
Base + 8A9	RES	8	SCC3	Reserved	
Base + 8AA	SCCM3	8	SCC3	SCC3 Mask Register	00
Base + 8AB	RES	8	SCC3	Reserved	
Base + 8AC	SCCS3	8	SCC3	SCC3 Status Register	00
Base + 8AD	RES	8	SCC3	Reserved	
Base + 8AE	RES	16	SCC3	Reserved	
Base + 8B0	SPMODE	16	SCM	SCP, SMC Mode and Clock Control Register	0000
Base + 8B2 #	SIMASK	16	SI	Serial Interface Mask Register	FFFF
Base + 8B4 #	SIMODE	16	SI	Serial Interface Mode Register	0000
Base + 8B6 • • • Base + FFF				Reserved (Not Implemented)	

\# Reset only upon a total system reset ($\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ assert together), but not on the execution of an M68000 RESET instruction. See the RESET pin description for details.

The output latches are undefined at total system reset.

! Event register with special properties (see 2.9 Event Registers).

2.9 EVENT REGISTERS

The IMP contains a few special registers designed to report events to the user. They are the channel status register (CSR) in the independent DMA, the interrupt pending register (IPR) and interrupt in-service register (ISR) in the interrupt controller, the timer event register 1 (TER1) in timer 1, the TER2 in timer 2, serial communication controller event register 1 (SCCE1) in SCC1, SCCE2 in SCC2, and SCCE3 in SCC3. Events in these register are always reported by a bit being set.

During the normal course of operation, the user software will clear these events after recognizing them. To clear a bit in one of these registers, the user software must WRITE A ONE TO THAT BIT. Writing a zero has no effect on the register. Thus, in normal operation, the hardware only *sets* bits in these registers; whereas, the software only *clears* them.

This technique prevents software from inadvertently losing the indication from an event bit that is “set” by the hardware between the software read and the software write of this register.

All these registers are cleared after a total system reset ($\overline{\text{RESET}}$ and $\overline{\text{HALT}}$ asserted together) and after the M68000 RESET instruction. Also some of the blocks (IDMA, timer1, timer2, and communication processor) have a reset (RST) bit located in a register in that block. This RST bit will reset that entire block, including any event registers contained therein.

Examples:

1. To clear bit 0 of SCCE1, execute "MOVE.B #$01,SCCE1"

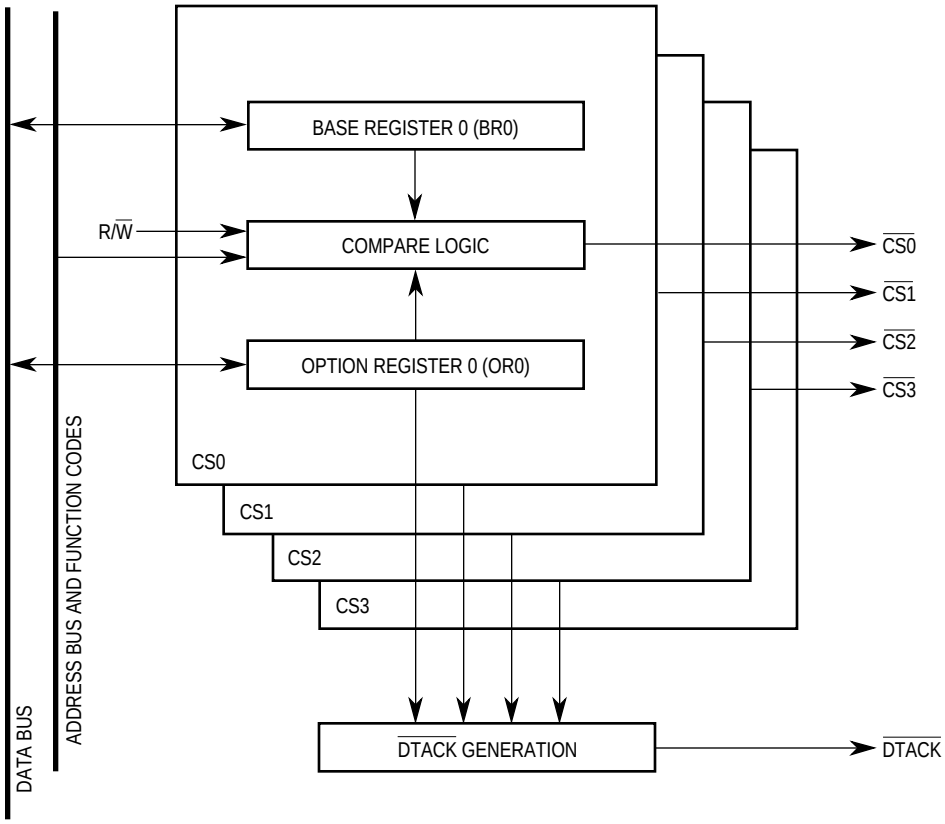


Figure 3-9. Chip-Select Block Diagram

The user should not normally program more than one chip-select line to the same area. When this occurs, the address compare logic will set address decode conflict (ADC) in the system control register (SCR) and generate $\overline{\text{BERR}}$ if address decode conflict enable (ADCE) is set. Only one chip-select line will be driven because of internal line priorities. $\overline{\text{CS0}}$ has the highest priority, and $\overline{\text{CS3}}$ the lowest. $\overline{\text{BERR}}$ will not be asserted on write accesses to the chip-select registers.

If one chip select is programmed to be read-only and another chip select is programmed to be write-only, then there will be no overlap conflict between these two chip selects, and the ADC bit will not be set.

When a bus master attempts to write to a read-only location, the chip-select logic will set write protect violation (WPV) in the SCR and generate $\overline{\text{BERR}}$ if write protect violation enable (WPVE) is set. The $\overline{\text{CS}}$ line will not be asserted.

NOTE

The chip-select logic is reset only on total system reset (assertion of $\overline{\text{RESET}}$ and $\overline{\text{HALT}}$). Accesses to the internal RAM and registers, including the system configuration registers (BAR and

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 $\overline{UDS}$.

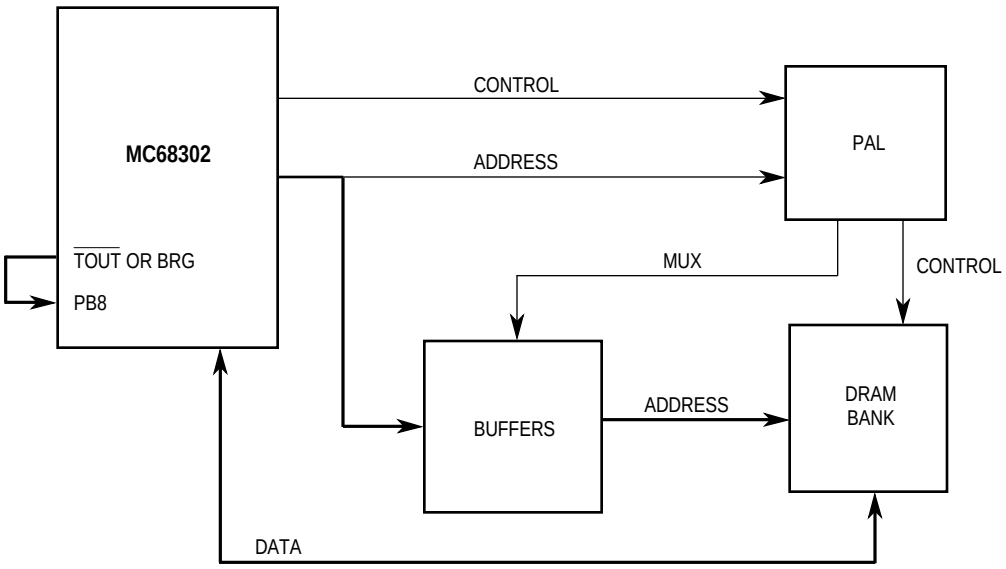


Figure 3-13. DRAM Control Block Diagram

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.

3. DRAM Refresh Controller
4. Commands Issued to the Command Register
5. SCC1 Receive Channel
6. SCC1 Transmit Channel
7. SCC2 Receive Channel
8. SCC2 Transmit Channel
9. SCC3 Receive Channel
10. SCC3 Transmit Channel
11. SMC1 Receive Channel
12. SMC1 Transmit Channel
13. SMC2 Receive Channel
14. SMC2 Transmit Channel
15. SCP Receive Channel
16. SCP Transmit Channel

For details on the DRAM refresh controller, see 3.10 Dynamic Ram Refresh Controller.

4.2 SDMA CHANNELS

Six serial (SDMA) channels are associated with the three full-duplex SCCs. Each channel is permanently assigned to service the receive or transmit operation of one of the SCCs and is always available, regardless of the SCC protocol chosen.

The SDMA channels allow flexibility in managing the data flow. The user can, on a buffer-by-buffer basis, determine whether data should be transferred between the SCCs and external memory or between the SCCs and on-chip dual-port RAM. This choice is controlled in each SCC buffer descriptor. The SCC to external memory path bypasses the dual-port RAM by allowing the SDMA channel to arbitrate for the M68000 bus directly. The SCC to dual-port RAM path saves external memory and eliminates the need to arbitrate for the bus.

Figure 4-2 shows the paths of the data flow. Data from the SCCs may be routed directly to external RAM as shown in path 1. In path 2, data is sent over the peripheral bus to the internal dual-port RAM. The SMCs and SCP, shown in path 3, always route their data to the dual-port RAM since they only receive and transmit a byte at a time.

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.

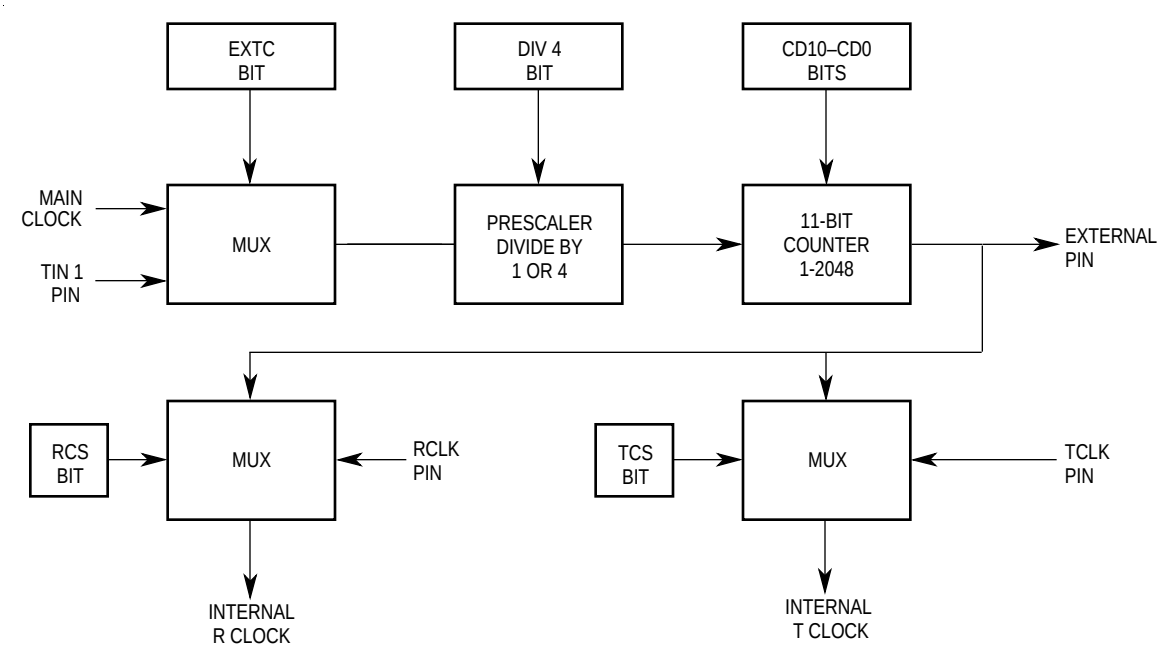


Figure 4-12. SCC Baud Rate Generator

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

table, after processing of this BD is complete. After using a BD, the CP sets the “ready” bit to not-ready; thus, the CP will never use a BD twice until the BD has been confirmed by the M68000 core.

The CP uses the receive BDs in a similar fashion. Once the receive side of an SCC is enabled, it starts with the first BD in that SCC's receive BD table. Once data arrives from the serial line into the SCC, the CP performs certain required protocol processing on the data and moves the resultant data (either bytes or words at a time depending on the protocol) to the buffer pointed to by the first BD. Use of a BD is complete when there is no more room left in the buffer or when certain events occur, such as detection of an error or an end-of-frame. Whatever the reason, the buffer is then said to be “closed,” and additional data will be stored using the next BD. Whenever the CP needs to begin using a BD because new data is arriving, it will check the “empty” bit of that BD. If the current BD is not empty, it will report a” busy” error. However, it will not move from the current BD until it becomes empty. When the CP sees the “wrap” bit set in a BD, it goes back to the beginning of the BD table, after use of this BD is complete. After using a BD, the CP sets the “empty” bit to not-empty; thus, the CP will never use a BD twice until the BD has been “processed” by the M68000 core.

In general, each SCC has eight transmit BDs and eight receive BDs. However, it is possible in one special case to assign up to 16 receive BDs at the expense of all transmit BDs. Since the transmit BDs directly follow the receive BDs in the memory map for each SCC, if an SCC is configured exclusively for half-duplex reception, it is possible to have up to 16 receive BDs available for that SCC.

If the DRAM refresh unit is used, SCC2 has six transmit BDs rather than the normal eight. SCC3 normally only has four transmit BDs. However, it is actually possible to regain additional Tx BDs for SCC3 as follows. The Tx BD table may be extended by two BDs to six BDs if the SMCs are not used. Additionally, all eight Tx BDs for SCC3 may be used if the following is considered: 1) the SCP and SMCs must not be used; 2) various words within the last two BDs will be changed by the CP during the initialization routine following any reset; and 3) the BERR channel number value will be written into the last BD after any SDMA bus error (see 4.5.8.4 Bus Error on SDMA Access), but this is not a major concern since the CP must be reset after any SDMA bus error.

4.5.6 SCC Parameter RAM Memory Map

Each SCC maintains a section in the dual-port RAM called the parameter RAM. Each SCC parameter RAM area begins at offset $80 from each SCC base area ($400, $500, or $600) and continues through offset $BF. Refer to Table 2-8 for the placement of the three SCC parameter RAM areas. Part of each SCC parameter RAM (offset $80–$9A), which is identical for each protocol chosen, is shown in Table 4-6. Offsets $9C–$BF comprise the protocol-specific portion of the SCC parameter RAM and are discussed relative to the particular protocol chosen.

sion of a block. For the receiver, the ENQ character designates the end of the block, but no CRC is expected.

Following control character reception (i.e., end of the block), the RCH bit in the BISYNC mask register should be set, re-enabling interrupts for each byte of received data.

4.5.14 DDCMP Controller

The byte-oriented digital data communications message protocol (DDCMP) was originated by DEC for use in networking products. The three classes of DDCMP frames are transparent (or maintenance) messages, data messages, and control messages (see Figure 4-34). Each class of frame starts with a standard two octet synchronization pattern and ends with a CRC. Depending upon the frame type, a separate CRC may be present for the header as well as the data portions of the frame. These CRCs use the same 16-bit generator polynomial as that used in HDLC.

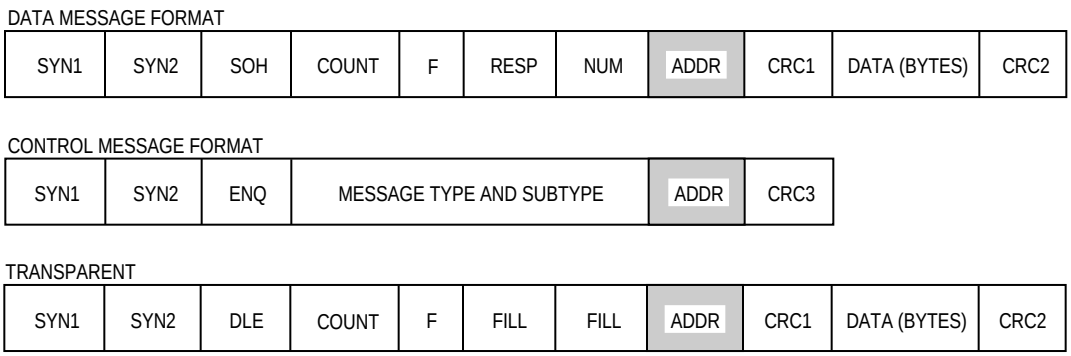


Figure 4-34. Typical DDCMP Frames

The most notable feature of the DDCMP frame is that the frame length is transmitted within the frame itself. Thus, any character pattern can be transmitted in the data field since the character count is responsible for ending the frame, not a special character. For this to work properly, the header containing the frame length must be protected, causing a need for a CRC in the frame header.

The bulk of the frame is divided into fields whose meaning depends on the frame type. Defined control characters are only used in the fixed-length frame headers (the fields between the synchronization octets and the first CRC). The following fields are one byte each: SYN1, SYN2, SOH, RESP, NUM, ADDR, ENQ, DLE, and FILL. The following fields are two bytes each: COUNT + F, CRC1, CRC2, and CRC3. The DATA field is a variable number of bytes, as defined in the COUNT field.

DDCMP communications can be either synchronous or asynchronous, with both types using the same frame format. Synchronous DDCMP frames require the physical layer to transmit the clock along with data over the link. Asynchronous DDCMP frames are composed of asynchronous UART characters, which together form the frame. The receiver and transmitter clocks are not linked; the receiver resynchronizes itself every byte using the start and stop bits of each UART character.

6.9 AC ELECTRICAL SPECIFICATIONS—DMA (see Figure 6-6 and Figure 6-7)

Num.	Characteristic	Symbol	16.67 MHz		20 MHz		25 MHz		Unit
			Min	Max	Min	Max	Min	Max	
80	$\overline{DREQ}$ Asynchronous Setup Time (see Note 1)	t_{REQASI}	15	—	15	—	10	—	ns
81	$\overline{DREQ}$ Width Low (see Note 2)	t_{REQL}	2	—	2	—	2	—	clks
82	$\overline{DREQ}$ Low to $\overline{BR}$ Low (see Notes 3 and 4)	$t_{REQLBRL}$	—	2	—	2	—	2	clks
83	Clock High to $\overline{BR}$ Low (see Notes 3 and 4)	t_{CHBRL}	—	30	—	25	—	20	ns
84	Clock High to $\overline{BR}$ High Impedance (see Notes 3 and 4)	t_{CHBRZ}	—	30	—	25	—	20	ns
85	$\overline{BGACK}$ Low to $\overline{BR}$ High Impedance (see Notes 3 and 4)	t_{BKLBRZ}	30	—	25	—	20	—	ns
86	Clock High to $\overline{BGACK}$ Low	t_{CHBKL}	—	30	—	25	—	20	ns
87	$\overline{AS}$ and $\overline{BGACK}$ High (the Latest One) to $\overline{BGACK}$ Low (when $\overline{BG}$ Is Asserted)	t_{ABHBKL}	1.5	2.5 +30	1.5	2.5 +25	1.5	2.5 +20	clks ns
88	$\overline{BG}$ Low to $\overline{BGACK}$ Low (No Other Bus Master) (see Notes 3 and 4)	t_{BGLBKL}	1.5	2.5 +30	1.5	2.5 +25	1.5	2.5 +20	clks ns
89	$\overline{BR}$ High Impedance to $\overline{BG}$ High (see Notes 3 and 4)	t_{BRHBGH}	0	—	0	—	0	—	ns
90	Clock on which $\overline{BGACK}$ Low to Clock on which $\overline{AS}$ Low	$t_{CLBKLAL}$	2	2	2	2	2	2	clks
91	Clock High to $\overline{BGACK}$ High	t_{CHBKH}	—	30	—	25	—	20	ns
92	Clock Low to $\overline{BGACK}$ High Impedance	t_{CLBKZ}	—	15	—	15	—	10	ns
93	Clock High to $\overline{DACK}$ Low	t_{CHACKL}	—	30	—	25	—	20	ns
94	Clock Low to $\overline{DACK}$ High	t_{CLACKH}	—	30	—	25	—	20	ns
95	Clock High to $\overline{DONE}$ Low (Output)	t_{CHDNL}	—	30	—	25	—	20	ns
96	Clock Low to $\overline{DONE}$ High Impedance	t_{CLDNZ}	—	30	—	25	—	20	ns
97	$\overline{DONE}$ Input Low to Clock Low (Asynchronous Setup)	t_{DNLTCH}	15	—	15	—	10	—	ns

NOTES:
1. $\overline{DREQ}$ is sampled on the falling edge of CLK in cycle steal and burst modes.
2. If #80 is satisfied for $\overline{DREQ}$, #81 may be ignored.
3. $\overline{BR}$ will not be asserted while $\overline{BG}$, $\overline{HALT}$, or $\overline{BERR}$ is asserted.
4. Specifications are for DISABLE CPU mode only.
5. $\overline{DREQ}$, $\overline{DACK}$, and $\overline{DONE}$ do not apply to the SDMA channels.
6. DMA and SDMA read and write cycle timing is the same as that for the M68000 core.

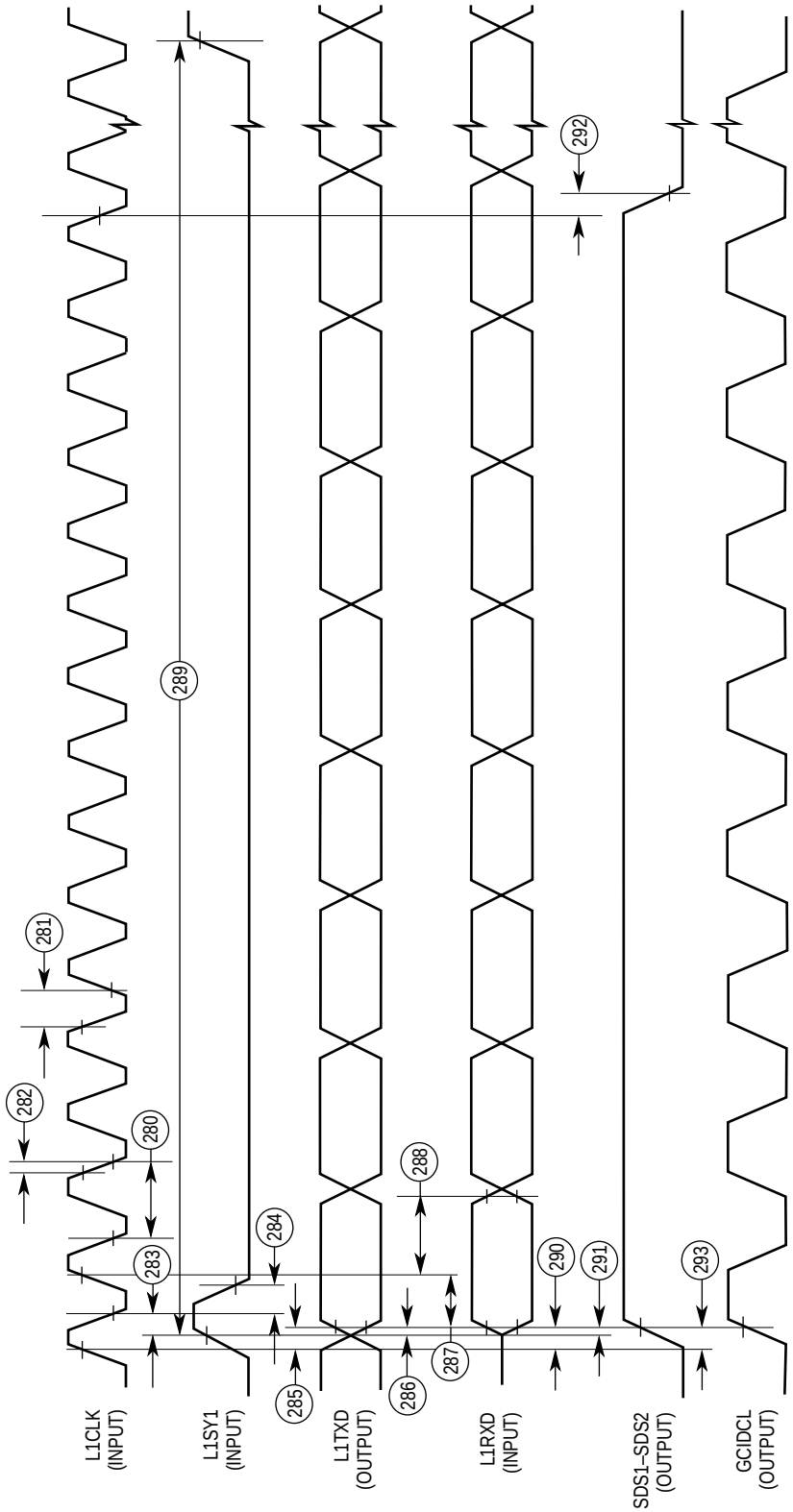


Figure 6-21. GCI Timing Diagram

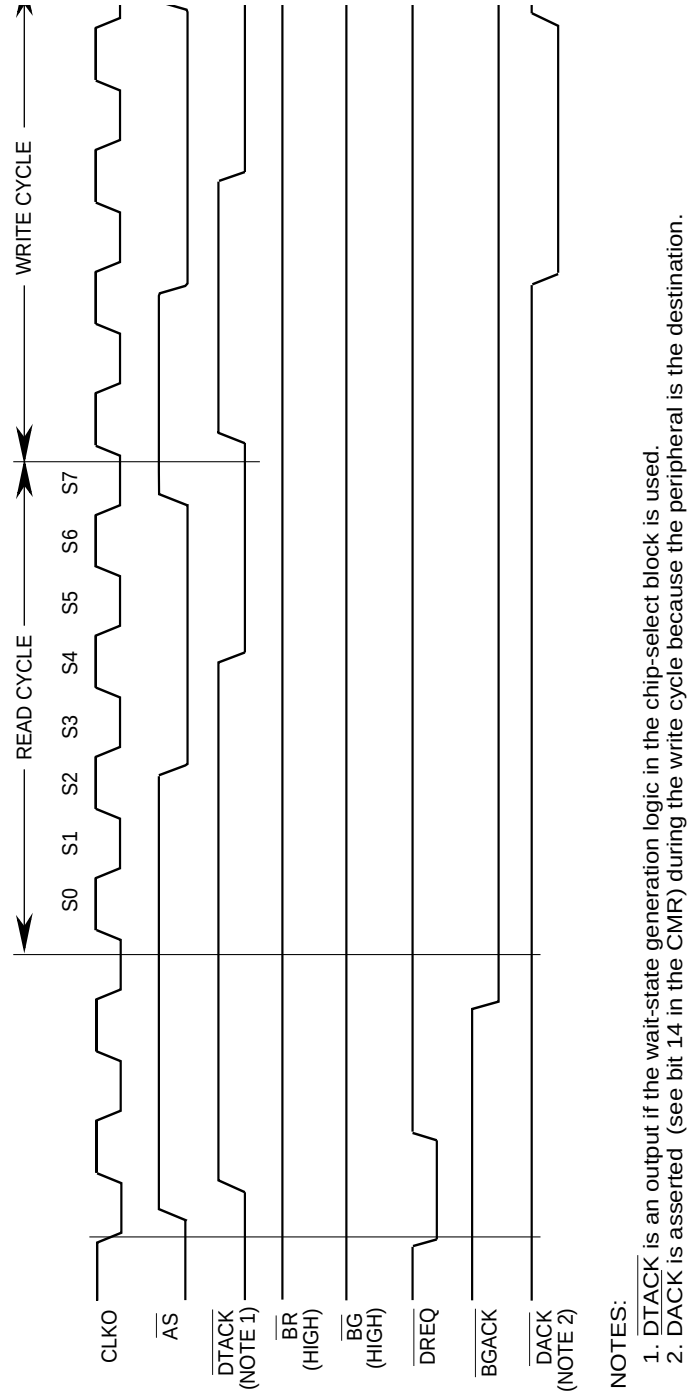
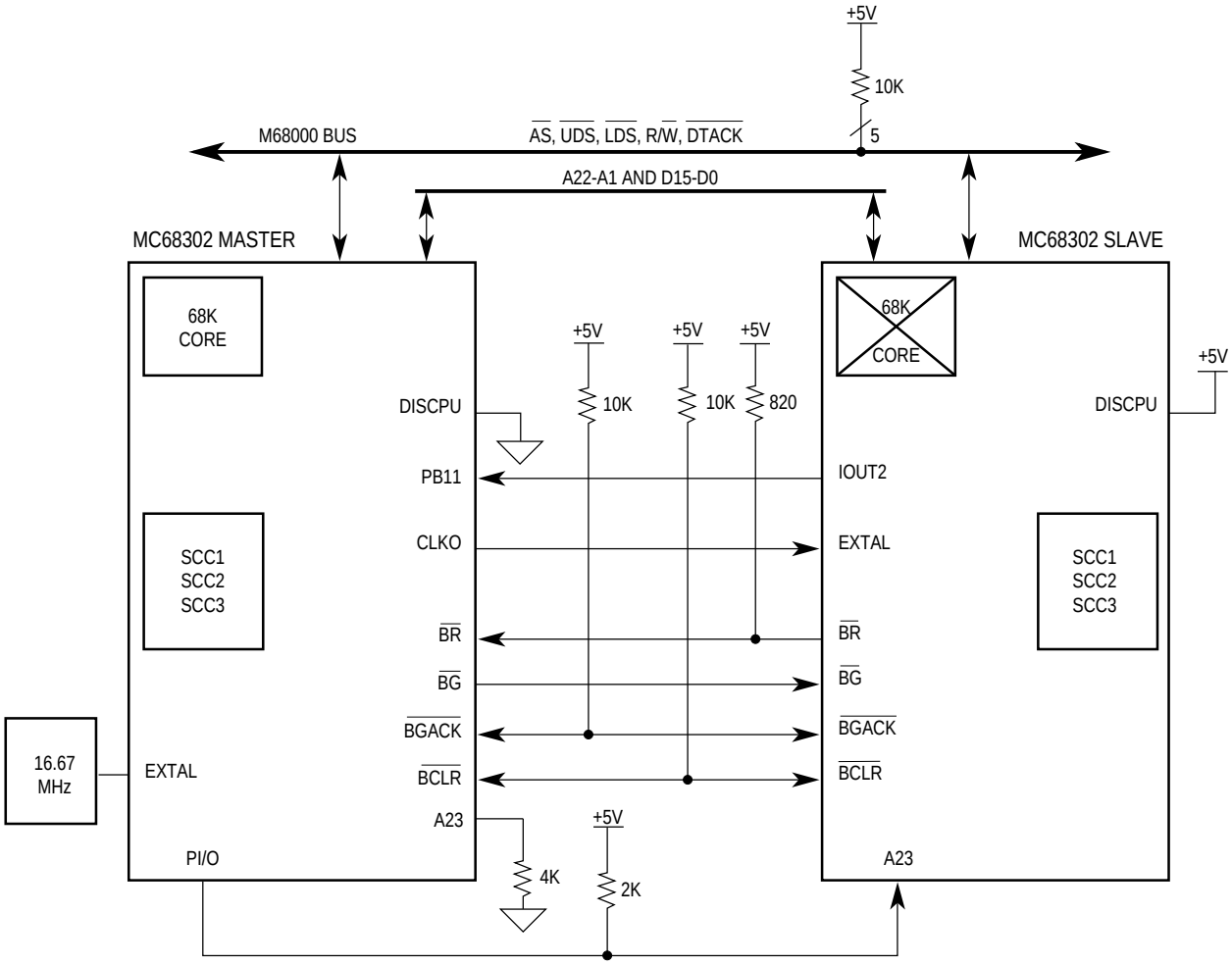


Figure D-7. Typical IDMA External Cycles (Normal Operation)

Figure D-8 shows the peripheral terminating the current IDMA transfer. Note that the DONE signal has been asserted by the peripheral. The IDMA will indicate the transfer has been completed by setting bits in the CSR (see Table D-3).

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

In the Tx BD, the last (L) bit should be set, and the TB, BR, TD, and TR bits should be cleared. In each Rx BD, the CR bit, which indicates a bad BCS, should be ignored.

If all the frames are of a fixed length, you do not need to use ETX. Instead, disable the whole control character table, and set the MRBLR to the frame length. If MRBLR = 2, for example, then you can send and receive the following frame types:

syn-syn-Data-Data-syn-syn-syn-syn-Data-Data-syn-syn

where syn is a one-byte sync character that cannot be sent as data.

To be able to send the sync character within the data stream requires full BISYNC capabilities in a mode called BISYNC transparent, which is not discussed in this subsection.

D.8.6.2 TRANSYNC MODE. In the normal transparent mode examples discussed previously, both the NTSYN and the EXSYN bits were set in the SCM register. Also, different ways of using BISYNC mode have been described in which both the NTSYN and EXSYN bits are cleared. However, what happens if you set NTSYN and clear EXSYN? The answer is a combination of transparent and BISYNC modes that is referred to here as TRANSYNC.

On the transmission side, normal transparent operation takes place with no sync characters transmitted. On the receive side, however, reception will not be synchronized until the pattern in the DSR is matched on the line. In other words, the $\overline{CD}$ (sync) function is eliminated on transmit and is replaced with the DSR matching function on receive. Recall that $\overline{CD}$ and $\overline{CTS}$ can still control transmission and reception in TRANSYNC mode if the DIAG1-DIAG0 bits are set for normal mode and not software operation mode.

NOTE

When NTSYN is cleared and EXSYN is set, the result is normal BISYNC mode except that the external synchronization function, $\overline{CD}$ (sync), is required for proper reception. Syncs are transmitted in this mode, but are not required on receive. This is the opposite of TRANSYNC mode.

D.8.7 Gating Clocks in NMSI Mode

If the behavior of CTS and CD (sync) are not what is needed for an application, there is always the possibility of gating clocks to the SCC. The term “gating clocks” usually means providing clocks to an SCC only while it is in the act of transmitting or receiving, but at no other time. Gating clocks is a requirement in some multidrop applications and can be useful for many special applications. Gating clocks is only possible if the clocks are inputs to the SCC since the internal SCC baud rate generators do not support gating clocks.

The gating of clocks can provide extra control over the transmission and reception of data, albeit with extra logic external to the MC68302. The SCCs are designed with static logic; thus, the clock signal may be held in a constant high/low state for any period of time. Whenever clocks are provided externally (and especially when they are gated), care should be taken to avoid glitches, excessive ringing, and very long rise/fall times in a very noisy environment. If the minimum clock high/low time is violated, erratic operation can result, which can cause an SCC to immediately transition to an error state such as underrun or overrun.

E.2.1.1.2 Serial Interface Mode Register (SIMODE).

15	14	13	12	11	10	9	8
SETZ	SYNC/SCIT	SDIAG1	SDIAG0	SDC2	SDC1	B2RB	B2RA

7	6	5	4	3	2	1	0
B1RB	B1RA	DRB	DRA	MSC3	MSC2	MS1	MS0

SETZ—Set L1TXD to Zero (valid only for the GCI interface)

0 = Normal operation.
1 = L1TXD output set to a logic zero (used in GCI activation).

SYNC/SCIT—SYNC Mode/SCIT Select Support

0 = One pulse wide prior to the 8-bit data.
1 = N pulses wide and envelopes the N-bit data.

SDIAG1, SDIAG0—Serial Interface Diagnostic Mode

00 = Normal operation.
01 = Automatic echo.
10 = Internal loopback.
11 = Loopback control.

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).

B1 RB, B1 RA—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).

E—End of Table

0 = This entry is valid.
1 = This entry is not valid. No valid entries lie beyond this entry.

R—Reject Character

0 = The control character is written into the receive data buffer, and the buffer is then closed. A new receive buffer is used if there is more data in the message.
1 = The control character is written to the RCCR instead of the received data butter. The current buffer is not closed.

E.2.1.3.12 Character8—Control Characters8. This 16-bit table entry defines a control character that should be compared to the incoming characters or is used to transmit out-of sequence characters, such as XON and XOFF.

15	14	13	12	11	10	9	8	7 … 0
E	R	REA	I	CT	0	0	A	CHARACTER8

E—End of Table

0 = This entry is valid as a receive control character.
1 = This entry contains a character to be transmitted out of sequence.

R—Reject Character

0 = Must be zero to use this entry as a flow control transmission character, otherwise, function is the same as for CHARACTERS 1-7.
1 = For receive control characters, the meaning is the same as for CHARACTERS 1-7.

REA—Ready

0 = Character is not ready for transmission
1 = Character is ready for transmission

I—Interrupt

0 = No interrupt.
1 = The TX bit in the event register will be set when this character is transmitted.

CT—Clear-to Send Lost

0 = $\overline{\text{CTS}}$ remained asserted.
1 = $\overline{\text{CTS}}$ was negated during transmission of this character.

A—Address

0 = Address bit of zero will be transmitted if a multidrop mode is chosen.
1 = Address bit of one will be transmitted if a multidrop mode is chosen.

E.2.1.4 RECEIVE BUFFER DESCRIPTORS. Each SCC has eight receive buffer descriptors. Each buffer descriptor consists of four words as shown below. Reserved bits in registers should be written as zeros.

INDEX

W

X