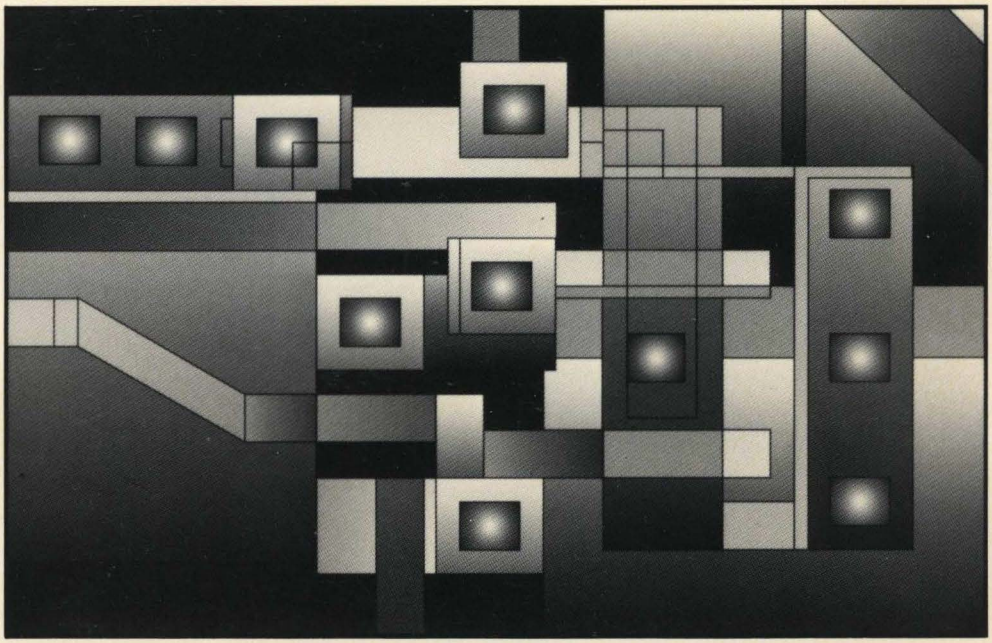
ZiLOG

TECHNICAL MANUAL



Z16C35

CMOS ISCC™ INTEGRATED SERIAL COMMUNICATIONS CONTROLLER

Q3/90

PREFACE

The purpose of this Technical Manual is to explain and illustrate the hardware and software technical details of the Z16C35, Integrated Serial Communications Controller (ISCC). The document is organized into six chapters and one appendix as follows:

1. General Description

Chapter 1 is an introductory section covering the key features, overview block diagrams, pin-out, and pin definitions.

2. Interfacing the ISCC

Chapter 2 provides the technical information to describe the bus interfaces, I/O interfaces, and register accesses.

3. Detailed Functional Descriptions

Chapter 3 covers the ISCC's functional details in the areas of receiver and transmitter DMA operations, baud rate generator, data encoding/decoding, digital phase locked loop, clock selection, and crystal oscillators.

4. Modes of Operation

Chapter 4 explains the serial data transmission and reception in asynchronous, bit- and byte-oriented synchronous modes.

5. Register Descriptions

Chapter 5 illustrates and explains the bit combinations and definitions for all ISCC registers.

6. Appendix A

Appendix A illustrates and explains the interfacing of the ISCC to Intel and Motorola microprocessors.

Table of Contents

Figure List

Table List

CHAPTER 1
GENERAL DESCRIPTION

1.1 INTRODUCTION

The Z16C35, ISCC is a CMOS superintegration device with a flexible Bus Interface Unit (BIU) connecting a built-in Direct Memory Access (DMA) cell to the CMOS Serial Communications Control (SCC) cell.

The ISCC is a dual-channel, multi-protocol data communications peripheral which easily interfaces to CPU's with either multiplexed or non-multiplexed address and data buses. The advanced CMOS process offers lower power consumption, higher performance, and superior noise immunity. The programming flexibility of the internal registers allow the ISCC to be configured for a wide variety of serial communications applications. The many on-chip features such as streamlined bus interface, four channel DMA, baud rate generators, digital phase-locked loops, and crystal oscillators dramatically reduce the need for external logic. Additional features, including a 10x19 bit status FIFO, are added to support high speed SDLC transfers using on-chip DMA controllers.

The ISCC can address up to 4 gigabytes per DMA channel by using the /UAS and /AS signals to strobe out 32-bit multiplexed addresses.

The ISCC handles asynchronous formats, synchronous byte-oriented protocols such as IBM Bisync, and synchronous bit-oriented protocols such as HDLC and IBM SDLC. This versatile device supports virtually any serial data transfer application (terminals, printers, diskette, tape drives, etc.).

The device can generate and check CRC codes in any synchronous mode and can be programmed to check data integrity in various modes. The ISCC also has facilities for modem controls in both channels. In applications where these controls are not needed, the modem controls can be used for general-purpose I/O.

The standard Zilog interrupt daisy chain is supported for interrupt hierarchy control. Internally, the SCC cell has higher interrupt priority than the DMA cell.

The DMA cell consists of four DMA channels; one for transmit and one for receive to and from each SCC channel, respectively.

The DMA cell adopts a simple fly-by mode DMA transfer, providing a powerful and efficient DMA access. The cell does not support memory-to-memory transfer.

Priorities between the four DMA channels are programmable to custom-fit user applications. Arbitration of Bus priority control signals between the ISCC DMA and other system DMA's should be handled outside the ISCC.

The BIU has a universal interface to most system/CPU bus structures and timing. The first write to the ISCC after a hardware reset will configure the bus interface type being implemented.

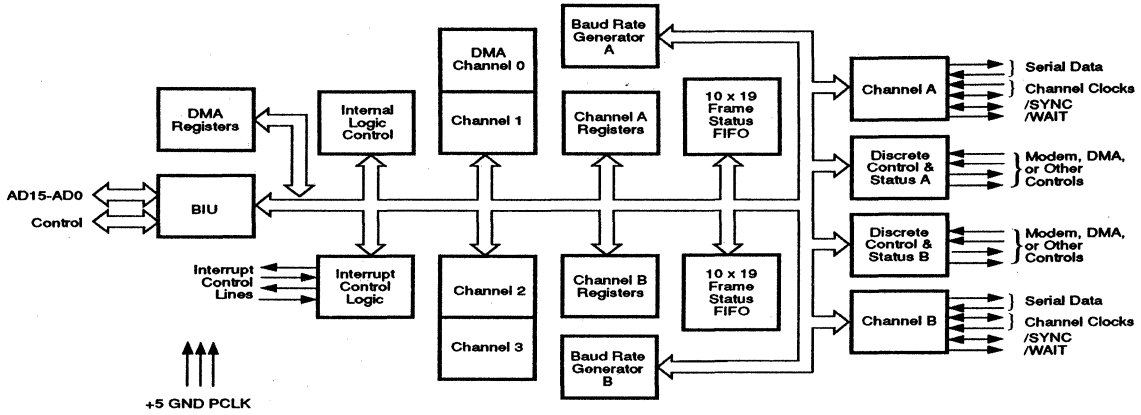


Figure 1-1. Block Diagram

1.2 FEATURES

- Low power CMOS technology
- Two general-purpose SCC channels, four DMA channels; and Universal Bus Interface Unit
- Software compatible to the Zilog CMOS SCC
- Four DMA channels; two transmit and two receive channels to and from the SCC
- Four gigabyte address range per DMA channel
- Flyby DMA transfer mode
- Programmable DMA channel priorities
- Independent DMA register set
- A Universal Bus Interface Unit providing simple interface to most CPUs multiplexed or non-multiplexed bus; compatible with 680x0 and 8x86 CPUs
- 32-bit addresses multiplexed to 16-pin address/data lines
- 8-bit data supporting high/low byte swapping
- 10 MHz timing
- 12.5 & 16 MHz timing planned
- 68-pin PLCC
- Supports all Zilog CMOS SCC features:
- Two independent, 0 to 4.0 M bit/second, full-duplex channels, each with a separate crystal oscillator, baud rate generator, and digital phase-locked loop circuit for clock recovery.
- Multi-protocol operation under program control; programmable for NRZ, NRZi, or FM data encoding.
- Asynchronous mode with five to eight bits and one, one and one-half, or two stop bits per character; programmable clock factor; break detection and generation; parity, overrun, and framing error detection.
- Synchronous mode with internal or external character synchronization on one or two synchronous characters and CRC generation and checking with CRC-16 or CRC-CCITT preset to either 1's or 0's.
- SDLC/HDLC mode with comprehensive frame-level control, automatic zero insertion and deletion, I-field residue handling, abort generation and detection, CRC generation and checking, and SDLC Loop mode operation.
- Local Loopback and Auto Echo modes
- Supports T1 digital trunk
- Enhanced SDLC 10x19 Status FIFO for DMA support
- Full CMOS SCC register set

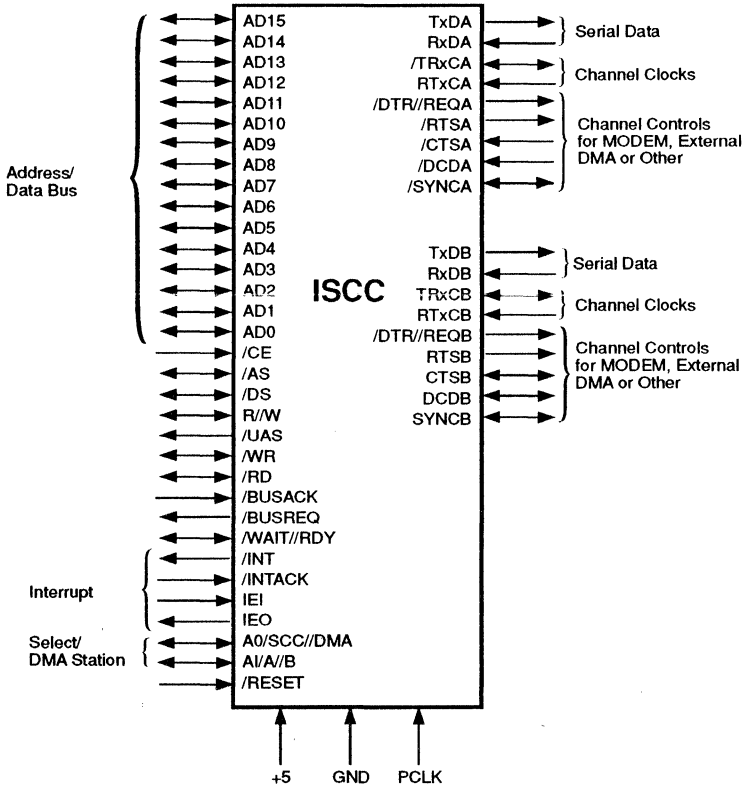


Figure 1-2. Pin Functions

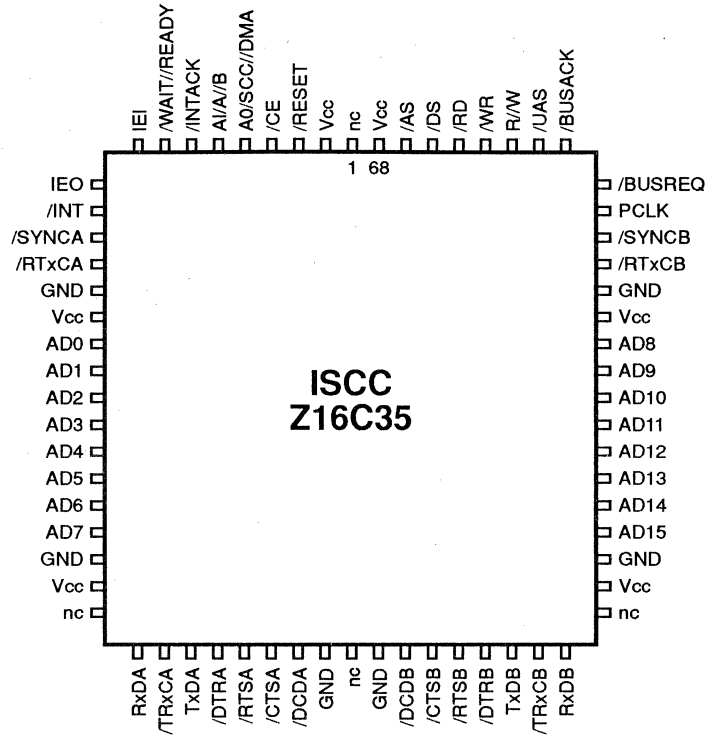


Figure 1-3. Pin Assignments

1.3 PIN DESCRIPTION

The following section describes the Z16C35 pin functions. Figures 1-2 and 1-3 detail the respective pin functions and pin assignments. All references to DMA are internal.

/CTSA, /CTSB. *Clear To Send* (inputs, active Low). These pins function as transmitter enables if they are programmed for Auto Enables (WR3, D5). If these pins are programmed as Auto Enables, a Low on the inputs enables the respective transmitters. If not programmed as Auto Enables, they may be used as general-purpose inputs. Both inputs are Schmitt-trigger buffered to accommodate slow rise-time inputs. The SCC cell detects transitions on these inputs and can interrupt the CPU on both low to high and high to low transitions.

/DCDA, /DCDB. *Data Carrier Detect* (inputs, active Low). These pins function as receiver enables if they are programmed for Auto Enables (WR3 D5), otherwise they are used as general-purpose input pins. Both pins are Schmitt-trigger buffered to accommodate slow rise time signals. The SCC cell detects transitions on these inputs and can interrupt the CPU on both low to high and high to low transitions.

/DTR//REQA, /DTR//REQB. *Data Terminal Ready / Request* (outputs, active Low). These pins are programmable (WR14, D2) to serve as either general purpose outputs or as DMA request lines. When programmed for the DTR function These outputs follow the state programmed into the DTR bit of Write Register 5 (WR5, D7). When programmed for the Ready mode, these pins serve as DMA requests for the transmitter. Note that this DMA request is not associated with the on-chip DMA and is intended for use in requesting DMA service from an external DMA.

IEI. *Interrupt Enable In* (input, active High). IEI is used with IEO to form an interrupt daisy chain when there is more than one interrupt driven device. A high IEI indicates that no other higher priority device has an interrupt under service or is requesting an interrupt.

IEO. *Interrupt Enable Out* (output, active High). IEO is High only if IEI is High and the CPU is not servicing the ISCC (SCC or DMA) interrupt or the ISCC is not requesting an interrupt (Interrupt Acknowledge cycle only). IEO is connected to the next lower priority device's IEI input and thus inhibits interrupts from lower priority devices.

/INT. *Interrupt* (output, active Low). This signal is activated when the SCC or DMA requests an interrupt. Note that /INT is pulled high and is not an open-drain output.

/INTACK. *Interrupt Acknowledge* (input, active Low). This is a strobe which indicates that an interrupt acknowledge cycle is in progress. During this cycle, the SCC and DMA interrupt daisy chain is resolved. The device is capable of returning an interrupt vector that may be encoded with the type of interrupt pending during this acknowledge cycle when /RD or /DS become high. /INTACK may be programmed to accept a status acknowledge, a single pulse acknowledge, or a double pulse acknowledge. This is programmed in the Bus Configuration Register (BCR). The double pulse acknowledge is compatible with 8x86 family microprocessors.

PCLK. *Clock* (input). This is the master SCC cell and DMA cell clock used to synchronize internal signals. PCLK is a TTL level signal. PCLK is not required to have any phase relationship with the master system clock.

RxDA, RxDB. *Receive Data* (inputs, active High). These input signals receive serial data at standard TTL levels.

/RTxCA, /RTxCB. *Receive/Transmit Clocks* (inputs, active Low). These pins can be programmed to several modes of operation. In each channel, /RTxC may supply the receive clock, the transmit clock, the clock for the baud rate generator, or the clock for the Digital Phase-Locked Loop. These pins can also be programmed for use with the respective /SYNC pins as a crystal oscillator. The receive clock may be 1, 16, 32, or 64 times the data rate in asynchronous modes.

/RTSA, /RTSB. *Request To Send* (outputs, active Low). When the Request To Send (RTS) bit in Write Register 5 is set, the /RTS signal goes Low. When the RTS bit is reset in the Asynchronous mode and Auto Enable is on, the signal goes High after the transmitter is empty. In Synchronous mode or in Asynchronous mode with Auto Enable off, the /RTS pin strictly follows the state of the RTS bit. Both pins can be used as general-purpose outputs.

/SYNCA, /SYNCB. *Synchronization* (inputs or outputs, active Low). These pins can act either as inputs, outputs, or part of the crystal oscillator circuit. In the Asynchronous Receive mode (crystal oscillator option not selected), these pins are inputs similar to /CTS and /DCD. In this mode, transitions on these lines affect the state of the Sync/Hunt status bits in Read Register 0 but have no other function.

In External Synchronization mode with the crystal oscillator not selected, these lines also act as inputs. In this mode, /SYNC must be driven Low two receive clock cycles after the last bit in the synchronous character is received. Character assembly begins on the rising edge of the receive clock immediately preceding the activation of /SYNC.

In the Internal Synchronization mode (Monosync and Bisync) with the crystal oscillator not selected, these pins act as outputs and are active only during the part of the receive clock cycle in which sync condition is not latched. These outputs are active each time a sync pattern is recognized (regardless of character boundaries). In SDLC mode, the pins act as outputs and are valid on receipt of a flag. The output is active for one receive clock period (refer to Chapter 4).

TxDA, TxDB. *Transmit Data* (outputs, active high). These output signals transmit serial data at standard TTL levels.

/TRxCA, /TRxCB. *Transmit/Receive Clocks* (inputs or outputs, active Low). These pins can be programmed in several different modes of operation. /TRxC may supply the receive clock or the transmit clock in the input mode or supply the output of the Digital Phase-Locked Loop, the crystal oscillator, the baud rate generator, or the transmit clock in the output mode.

/CE. *Chip Enable* (input, active Low). This signal selects the ISCC for a peripheral read or write operation. This signal is ignored when the ISCC is bus master.

AD15-AD0. *Data bus* (bidirectional, 3-state). These lines carry data and commands to and from the ISCC.

/RD. *Read* (bidirectional, active Low). When the ISCC is a peripheral (i.e. bus slave), this signal indicates a read operation and when the ISCC is selected, enables the ISCC's bus drivers. As an input, /RD indicates that the CPU wants to read from the ISCC read registers. During the Interrupt Acknowledge cycle, /RD gates the interrupt vector onto the bus if the ISCC is the highest priority device requesting an interrupt. When the ISCC is the bus master, this signal is used to read data. As an output, after the ISCC has taken control of the system buses, /RD indicates a DMA-controlled read from a memory or I/O port address.

/WR. *Write* (bidirectional, active Low). When the ISCC is selected, this signal indicates a write operation. As an input, this indicates that the CPU wants to write control or command bytes to the ISCC write registers. As an output, after the ISCC has taken control of the system buses /WR indicates a DMA-controlled write to a memory or I/O port address.

/DS. *Data Strobe* (bidirectional, active Low). A Low on this signal indicates that the AD15-AD0 bus is used for data transfer. When the ISCC is not in control of the system bus and the external system is transferring information to or from the ISCC, /DS is a timing input used by the ISCC to move data to or from the AD15-AD0 bus. Data is written into the ISCC by the external system on the High to Low /DS transition. Data is read from the ISCC by the external system while /DS is Low. There are no timing requirements between /DS as an input and ISCC clock; this allows use of the ISCC with a system bus which does not have a bussed clock.

During a DMA operation when the ISCC is in control of the system, /DS is an output generated by the ISCC and used by the system to move data to or from the AD15-AD0 bus. When the ISCC has bus control, it writes to the external system by placing data on the AD15-AD0 bus before the High-to-Low /DS transition and holds the data stable until after the Low-to-High /DS transition; while reading from the external system, the Low-to-High transition of /DS inputs data from the AD15-AD0 bus into the ISCC.

R//W. *Read/Write* (bidirectional). Read polarity is High and write polarity is Low. When the ISCC is not in control of the system bus and the external system is transferring information to or from the ISCC, R//W is a status input used by the ISCC to determine if data is entering or leaving on the AD15-AD0 bus during /DS time. In such a case, Read (High) indicates that the system is requesting data from the ISCC and Write (Low) indicates that the system is presenting data to the ISCC. The only timing requirements for R//W as an input are defined relative to /DS. When the ISCC is in control of the system bus, R//W is an output generated by the ISCC, with Read (high) indicating that data is being requested from the addressed location or device, and Write (low) indicating that data is being presented to the addressed location or device.

/UAS. *Upper Address Strobe* (Output, active Low). This signal is used if the output address is more than 16-bit. The upper address, A31-A16, can be latched externally by the rising edge of this signal. /UAS is active first before /AS becomes active. This signal and /AS are used by the DMA cell.

/AS. *Lower Address Strobe* (Bidirectional, active Low). When the ISCC is bus master, this signal is an output, and is used as a lower address strobe for AD15-AD0. It is used in conjunction with /UAS since the address is 32-bits. This signal and /UAS are used by the DMA cell when it is bus master. When ISCC is not bus master, this signal is used in the multiplexed bus modes to latch the address on the AD lines. The /AS signal is not used in the non-multiplexed bus modes and should be tied to Vcc through a resistor in these cases.

/WAIT//RDY. *Wait/Ready* (bidirectional, active Low). This signal may be programmed to function either as a Wait signal or Ready signal during the BCR write. When the BCR is written to Channel A (A1/A//B High during the BCR write), this signal functions as a /WAIT and thus supports the READY function of 8X86 microprocessors family. When the BCR writes to Channel B (A1/A//B Low), this signal functions as a /READY and supports the /DTACK function of the 680X0 microprocessor family.

This signal is an output when the ISCC in not bus master. In this case, the /Wait//RDY signal indicates when the data is available during a read cycle; when the device is ready to receive data during a write cycle; and when a valid vector is available during an interrupt acknowledge cycle.

When the ISCC is the bus master (the DMA cell has taken control of the bus), the /Wait//RDY signal functions as a /WAIT or /READY input. Slow memories and peripheral devices can assert /WAIT to extend /DS during bus transfers. Similarly, memories and peripherals use /READY to indicate that its output is valid or that it is ready to latch input data.

/BUSACK. *Bus Acknowledge* (input, active Low). Signals the bus has been released to the DMA. If the /BUSACK goes inactive before the DMA transfer is completed, the current DMA transfer is aborted.

/BUSREQ. *Bus Request* (output, active Low). This signal is used by the DMA to obtain the bus from the CPU.

A0/SCC//DMA. *DMA Channel/SCC Select/DMA Select* (bidirectional). When this pin is used as input, a high selects the SCC cell and a low selects the DMA cell. When this pin is used as output, the signal on this pin is used in conjunction with A1/A//B pin output to identify which DMA channel is active. This information can be used by the user to determine whether to issue a DMA abort command. A0/SCC//DMA and A1/A//B output encoding is shown on the following page.

A1/A//B	A0/SCC//DMA	DMA channel
1	1	RxA
1	0	TxA
0	1	RxB
0	0	TxB

A1/A//B. *DMA Channel/Channel A/Channel B* (bidirectional). This signal, when used as input, selects the SCC channel in which the read and write operation occurs. Note that A0/SCC//DMA pin must be held high to select this feature. When this pin is used as an output, it is used in conjunction with the A0/SCC//DMA pin output to identify which DMA channel is active. During a DMA peripheral access, the A1/A//B pin is ignored.

/RESET. (input, active Low). This signal resets the device to a known state. The first write to the ISCC after a reset accesses the BCR to select additional bus options for the device.

CHAPTER 2
INTERFACING THE ISCC

2.1 INTRODUCTION

This chapter details the interfacing of the ISCC to a system. The Product Specification must be referenced for specific timing details.

2.2 BUS INTERFACE UNIT (BIU) DESCRIPTION

The ISCC contains a flexible bus interface that is compatible with a variety of microprocessors and microcontrollers. The device is designed to work with 8- or 16-bit bus systems and may be used with address/data multiplexed busses or non-multiplexed busses. The bus interface style is selected by certain actions which take place after a hardware reset.

The ISCC contains a Bus Configuration Register, the BCR. This register has no address and is only accessible in the first transaction to the ISCC after a hardware reset; this first transaction must be a write with AØ/sec//DMA Low and is automatically directed to the Bus Configuration Register by the ISCC. The Bus Configuration Register contains bits which program the byte swapping feature, the interrupt acknowledge type and other aspects of the bus interface configuration. Refer to Chapter 5 for BCR details.

The multiplexed bus is selected for the ISCC if there is an Address Strobe prior to or during the transaction which writes the BCR. If no Address Strobe is present prior to or during the transaction which writes the BCR, a non-multiplexed bus is selected. The address strobe is recognized whether or not the ISCC Chip Enable is active.

2.2.1 Non-multiplexed Bus Operation

When the ISCC is initialized for non-multiplexed operation, register addressing for the ISCC cell is (with the exception of WR0 and RR0), accomplished using an internal pointer accessed via WR0. Accessing internal registers by this means is a two step operation requiring a write to the pointer followed by access of the desired register. This is described in detail in later sections. Note that when the DMA is not used to address the data, the data registers must be accessed by pointing to Register 8. (This is in contrast to the Z8530 which allows direct addressing of the data registers through the C/D pin.)

When the ISCC is initialized for non-multiplexed operation, register addressing for the DMA cell (with the exception of CSAR) is accomplished in a manner similar to that used in the SCC cell. In this case the pointer is accessed in the Command Status Address Register (CSAR bits 4 - 0). The SCC cell and DMA cell pointers are independent. Detailed operation is described in a later section.

2.2.2 Multiplexed Bus Operation

When the ISCC is initialized for multiplexed bus operation, all registers in the SCC cell are directly addressable with the register address occupying AD5 through AD1, or AD4 through AD0 (Shift Left/Shift Right modes). The A0/SCC// DMA pin controls the SCC cell /DMA selection. The SCC cell channel A/B selection may be controlled either by the A0/A//B pin or by the A/B selection in the address on AD7-0 that is strobed into the ISCC with /AS. Use of this requires that the unused SCC channel select option to be set to Channel A. That is, if the A0/A//B pin is used to select the channel, then the AD bit for channel selection must select channel A (the actual bit is determined by the Shift Left/Shift Right mode employed) and conversly, if the AD bus bit is used to select the channel, then the A0/A//B pin must select channel A . Refer to the A0/SCC//DMA and A1/A//B pin descriptions for the encoding of these signals.

In the multiplexed bus mode of operation, the register pointer in WR0 of the SCC cell is ignored and has no effect on the accessing of the internal registers. Register access is made solely through the latched address. However, the pointer in the DMA Channel Command/Address Register functions in the multiplexed bus mode and may be used to access DMA registers in a manner identical to that in the non-multiplexed bus mode. To use the DMA pointer in the multiplexed bus mode, the multiplexed address must always address the CCAR of the DMA even though the actual register access will be made according to the pointer. This requires that in the normal multiplexed mode of operation with register access through the latched address, writes to the DMA CCAR must always write zeros to the pointer field.

In the multiplexed bus mode in some host configurations, address A0 may be used for byte transfer control in 16 bit systems. Therefore, it may be necessary to ignore A0 in the register decode. This is accomodated in the ISCC by providing an option to decode the multiplexed address from A1 upwards rather than from A0 upwards. This option is the Shift Left/Shift Right mode. The Shift Left/Shift Right modes for the address decoding for the internal registers (multiplexed bus) are separately programmable for the SCC cell and for the DMA cell. For the SCC cell the programming and operation is identical to that in the SCC; programming is accomplished through Write Register 0 (WR0), bits 1 and 0 (Figure 5-2). The programming of the Shift Left/Shift Right modes for the DMA cell is accomplished in the BCR, bit 0. In this case, the shift function is similar to that for the SCC cell; with Shift left, the internal register addresses are decoded from bits AD5 through AD1 and with Shift Right, the internal register addresses are decoded from bits AD4 through AD0.

When the multiplexed bus mode is selected, Write Register 0 (WR0) takes on the form of WR0 in the Z8030 (Figure 5-2).

2.2.3 Data Transfers

All data transfers to and from the ISCC are done in bytes even though the data may at special times occupy the lower or upper byte of the 16 bit bus. Bus transfers as a slave peripheral are done differently than bus transfers when the ISCC is the bus master during DMA transactions. The ISCC is fundamentally an 8 bit peripheral but supports 16 bit busses in the DMA mode. Slave peripheral and DMA transactions are described in the next paragraphs.

Data Bus Transfers as a Slave Peripheral: When accessed as a peripheral device (when the ISCC is not a bus master performing DMA transfers), only 8 bits are transferred. When the ISCC registers are read, the byte data present on the lower 8 bits of the bus is replicated on the upper 8 bits of the bus. Data is accepted by the ISCC only on the lower 8 bits of the bus.

ISCC DMA Bus Transfers: During DMA transfers, when the ISCC is bus master, only byte data is transferred. However, data may be transferred from the ISCC on the upper 8 bits of the bus or on the lower 8 bits of the bus. Moreover, odd or even byte transfers may be done on the lower or upper 8 bits of the bus. This is programmable and is described below.

During DMA transfers to memory from the ISCC, byte data only is transfered and the data appears on the lower 8 bits and is replicated on the upper 8 bits of the bus. Thus the data may be written to an odd or even byte of the system memory by address decoding and strobe generation.

During DMA transfers to the ISCC from memory, byte data only is transfered and normally data is accepted only on the lower 8 bits of the bus. However, the byte swapping feature may be used to enable data to be accepted on either the lower or upper 8 bits of the bus. The byte swapping feature is enabled by programming the Byte Swap Enable bit to a 1 in the BCR. The odd / even byte transfer selection is made by programming the Byte Swap Select bit in the BCR. If Byte Swap Select is a 1, then even address bytes (transfers where the DMA address has A0 equal 0) are accepted on the lower 8 bits of the bus and odd address bytes(transfers where the DMA address has A0 equal 1) are accepted on the upper 8 bits of the bus. If Byte Swap Select is a 0, then even address bytes (transfers where the DMA address has A0 equal 0) are accepted on the upper 8 bits of the bus and odd address bytes(transfers where the DMA address has A0 equal 1) are accepted on the lower 8 bits of the bus.

Table 2-1. ISCC Bus Access Summary

Process	Byte Enable	Swap Select	Action on Bus Lower 8 Bits	Action on Bus Upper 8 Bits
Read	x	x	data	same data
Write	x	x	data read	data ignored
DMA Write	0	x	data	same data
DMA Read	0	x	data read	data ignored
DMA Write	1	x	data	same data
DMA Read	1	0	depends upon A0 (see below)	

In the DMA Read with Byte Swap enabled:

ByteSwapSelect	A0	ISCC Accepts Data
0	0	Upper 8 Bits of Bus
0	1	Lower 8 Bits of Bus
1	0	Lower 8 Bits of Bus
1	1	Upper 8 Bits of Bus

In this table DMA read refers to a DMA controlled transfer from memory to the ISCC and DMA write refers to a DMA controllled transfer from the ISCC to memory. Read refers to a normal peripheral transaction where the CPU reads data from the ISCC and Write refers to a normal peripheral transaction where the CPU writes data to the ISCC.

2.3 I/O INTERFACE CAPABILITIES

The ISCC offers the choice of Polling, Interrupt (vectored or non-vectored), and DMA Transfer modes to transfer data, status, and control information to and from the CPU.

2.3.1 Polling

In this mode all interrupts and the DMA's are disabled. Three status registers in the SCC are automatically updated whenever any function is performed. For example, end-of-frame in SDLC mode sets a bit in one of these status registers. With polling, the CPU must periodically read a status register until the register contents indicate the need for some CPU action to be taken. Only one register in the SCC cell needs to be read; depending on the contents of the register, the CPU either reads data, writes data, or satisfies an error condition. Two bits in the register indicate the need for data transfer. An alternative is to poll the Interrupt Pending register to determine the source of an interrupt. The status for both SCC channels resides in one register.

2.3.2 Interrupts

When the ISCC responds to an Interrupt Acknowledge signal (INTACK) from the CPU, an interrupt vector is placed on the data bus. Both the SCC and the DMA contain vector registers. Depending on the source of interrupt, one of these vectors is returned, either unmodified or modified by the interrupt status to indicate the exact cause of the interrupt.

Each of the six sources of interrupt in the SCC (Transmit, Receive, and External/Status interrupts in both channels) and each DMA channel has three bits associated with the interrupt source: Interrupt Pending (IP), Interrupt Under Service (IUS), and Interrupt Enable (IE). If the IE bit is set for any given source of interrupt, then that source can request interrupts. The only exception to this rule is when the associate Master Interrupt Enable (MIE) bit is reset, then no interrupts are requested. Both the SCC cell and the DMA have an associated MIE bit. The IE bits in the SCC cell are write only, but the IE bits in the DMA are read/write.

The ISCC provides for nesting of interrupt sources with an interrupt daisy chain using the IEI, IEO, and /INTACK pins. As a microprocessor peripheral, the ISCC may request an interrupt only when no higher priority device is requesting one, e.g., when IEI is High. If the device in question requests an interrupt, it enables the /INT signal. The CPU then responds with /INTACK, and the interrupting cell places the vector on the data bus.

In the ISCC, the IP bit signals a need for interrupt servicing. When an IP bit is 1 and the IEI input pin is High, the /INT signal is activated, requesting an interrupt. In the SCC cell, if the IE bit is not set, then the IP for that source can never be set. The IP bits in the DMA cell are set independent of the IE bit.

The IUS bits signal that an interrupt request is being serviced. If an IUS is set, all interrupt sources of lower priority in the ISCC and external to the ISCC are prevented from requesting interrupts. The internal interrupt sources are inhibited by the state of the internal daisy chain, while lower priority devices are inhibited by the IEO output of the ISCC being pulled Low and propagated to subsequent peripherals. Internally, the SCC cell is higher priority than the DMA cell. An IUS bit is set during an Interrupt Acknowledge cycle if there are no higher priority devices requesting interrupts. The IUS bit must be cleared by the CPU. This is usually done at the end of the corresponding interrupt service routine.

Within the SCC portion of the ISCC there are three types of interrupts: Transmit, Receive, and External/Status. Each interrupt type is enabled under program control with Channel A having higher priority than Channel B, and with Receive, Transmit, and External/Status interrupts prioritized in that order within each channel. When the Transmit interrupt is enabled, the CPU is interrupted when the transmit buffer becomes empty. This implies that data has shifted from the transmit buffer to the transmitter, thus emptying the transmit buffer. When enabled, the receiver interrupts the CPU in one of three ways:

1. Interrupt on First Receive Character or Special Receive Condition

2. Interrupt on All Receive Characters or Special Receive Condition

3. Interrupt on Special Condition Only

Interrupt on First Character or Special Condition, and Interrupt on Special Condition Only, are typically used when doing block transfers with the DMA. A Special Receive Condition is one of the following: receiver overrun, framing error in Asynchronous mode, end-of-frame in SDLC mode and, optionally, a parity error. The Special Receive Condition interrupt is different from an Ordinary Receive Character Available interrupt only by the status placed in the vector during the Interrupt Acknowledge cycle. In Interrupt on First Receive Character, an interrupt occurs from Special Receive Conditions any time after the First Receive Character interrupt.

The main function of the External/Status interrupt is to monitor the signal transitions of the /CTS, /DCD, and /SYNC pins; however, an External/Status interrupt is also caused by a Transmit Underrun condition, or a zero count

in the baud rate generator, or by the detection of a Break (Asynchronous mode), Abort (SDLC mode) or EOP (SDLC Loop mode) sequence in the data stream. The interrupt caused by the Abort or EOP has a special feature allowing the ISCC to interrupt when the Abort or EOP sequence is detected or terminated. This feature facilitates the proper termination of the current message, correct initialization of the next message, and the accurate timing of the Abort condition in external logic.

2.3.3 DMA Interrupts

Each DMA in the ISCC has two sources of interrupt, which share an IP bit and an IUS bit, but have independent enables: Terminal Count and Abort. The Abort interrupt is generated when an active DMA channel is forced to terminate its transfers because /BUSACK is de-asserted during a transfer. The Terminal Count interrupt is generated when the DMA transfer count reaches zero. The DMA channels themselves are prioritized in a fixed order: Receive A, Transmit A, Receive B, and Transmit B.

When DMA transfers are used, the on-chip DMA channels transfer data directly to the transmit buffers or directly from the receive buffers. No other transfers are possible (for initialization, for example). The request signals from the receivers and transmitters are hard-wired to the request inputs of the DMA channels internally. Each DMA channel provides a 32-bit address which is either incremented or decremented with a 16-bit transfer length. Whenever a DMA channel receives a request from its associated receiver or transmitter and the DMA channel is enabled, the ISCC activates the /BUSREQ signal. Upon receipt of an active /BUSACK, the DMA channel transfers data between memory and the SCC cell. This transfer continues until the receiver or transmitter stops requesting a transfer or until the terminal count is reached, or /BUSACK is deactivated. The four DMA channels operate independently when the Request Per Channel option is selected; otherwise, all requests pending at the time of bus acquisition will be serviced before the bus is released. Each DMA channel is independently enabled and disabled.

2.4 REGISTER ACCESS

ISCC registers may be accessed explicitly, directly or indirectly. Explicit addressing occurs only for three registers in the ISCC: these are the Bus Configuration Register (for the first write after a hardware reset), the RDR (Receive Data Register) by a fly-by DMA read, and the TDR (Transmit Data Register) by a fly-by DMA write. In the non-multiplexed bus case, only WR0 / RR0 of the SCC cell and only the Channel Command / Address Register of the DMA cell are accessed directly. Other registers are accessed using the pointers in these directly accessed registers. In the multiplexed bus case, all registers (except the WR0, RR0 and CCAR) are accessed through a two step address / read - write bus transaction. In this case there are two options available for address decoding: shift right and shift left. These options are independently selectable for both the SCC cell and the DMA cell.

2.4.1 SCC Cell Register Access, Multiplexed Bus

The registers in the ISCC in the multiplexed bus mode are addressed via the address on AD7-AD0 which is latched by the rising edge of /AS. As discussed in the paragraphs below, the address contains a bit to select the SCC cell channel (A or B). Although this selection is in the address, the A0/A//B input remains active and must be set to select Channel A for the selection bit in the AD7-0 address to function correctly. Conversely, the A0/A//B pin may also be used to select the channel instead of the bit in the AD7-0 address. In this case, the bit in the AD7-0 address must be set to select Channel A for the A0/A//B input to function correctly.

There are two address decoding modes: shift left and shift right. In shift left mode, the register address is decoded from AD5-AD1. This mode is set by a hardware reset.

In the shift left mode, the register address itself is placed on AD4-AD1 and the Channel Select bit, A/B, is decoded from AD5. The register map for this case is shown in Table 2-2.

Table 2-2. SCC Cell Address Map, Multiplexed Bus Mode, Shift Left

Address AD5-AD1	Write	Read
10000	WR0A	RR0A
10001	WR1A	RR1A
10010	WR2	RR2A
10011	WR3A	RR3A
10100	WR4A	(RR0A)
10101	WR5A	(RR1A)
10110	WR6A	(RR2A)
10111	WR7A	(RR3A)
11000	WR8A	RR8A
11001	WR9	(RR13A)
11010	WR10A	RR10A
11011	WR11A	(RR15A)
11100	WR12A	RR12A
11101	WR13A	RR13A
11110	WR14A	(RR10A)
11111	WR15A	RR15A

Note:
The above table applies to Channel 'B' also.

In Shift Right Mode, bits 0-1 in WR0A controls which bits will be decoded to form the register address. It is placed in this register to simplify programming when the current state of the Shift Right/Shift Left bit is not known.

The register address is decoded from AD4-AD0. The Shift Right/Shift Left bit is written via command to make the software writing to WR0 independent of the state of the Shift Right/Shift Left bit.

AD4-AD0 is the actual register address and AD0 determines the channel selection (A//B). The register map is shown in Table 2-3.

Because the ISCC SCC Cell does not contain 16 read registers, the decoding of the read registers is not complete; this is indicated in Table 2-2 and Table 2-3 by parentheses around the register name. These addresses may also be used to access the read registers.

Note also that in the multiplexed bus mode, only one WR2 and WR9 are shown in the address map; these registers may be written from either SCC cell channel.

Table 2-3. SCC Cell Address Map, Multiplexed Bus Mode, Shift Right

Address AD4-AD0	Write	Read
00000	WR0B	RR0B
00001	WR0A	RR0A
00010	WR1B	RR1B
00011	WR1A	RR1A
00100	WR2	RR2B
00101	WR2	RR2A
00110	WR3B	RR3B
00111	WR3A	RR3A
01000	WR4B	RR0B
01001	WR4A	RR0A
01010	WR5B	(RR1B)
01011	WR5A	(RR1A)
01100	WR6B	RR2B
01101	WR6A	RR2A
01110	WR7B	(RR3B)
01111	WR7A	(RR3A)
10000	WR8B	RR8B
10001	WR8A	RR8A
10010	WR9	(RR13B)
10011	WR9	(RR13A)
10100	WR10B	RR10B
10101	WR10A	RR10A
10110	WR11B	(RR15B)
10111	WR11A	(RR15A)
11000	WR12B	RR12B
11001	WR12A	RR12A
11010	WR13B	RR13B
11011	WR13A	RR13A
11100	WR14B	(RR10B)
11101	WR14A	(RR10A)
11110	WR15B	RR15B
11111	WR15A	RR15A

2.4.2 SCC Cell Register Access, Non - Multiplexed Bus

The registers in the SCC cell in the non-multiplexed bus mode are accessed in a two-step process, using a Register Pointer to perform the addressing. To access a particular register, the pointer bits must be set by writing to WR0 bits 2, 1, and 0 and, if required, using the Point High command to extend the three bit pointer to registers 8 through 15. This write to WR0 to set the pointer bits may be done in either channel. There is only one pointer register and it is used for both A and B channels. After the pointer bits are set, the next read or write cycle to the SCC cell will access the desired register in the channel selected during this read or write cycle. At the conclusion of this read or write cycle, the pointer bits are reset to "0s," so that the next access will be to WR0.

The fact that the pointer bits are reset to "0," unless explicitly set otherwise, means that WR0 and RR0 may also be accessed in a single cycle. That is, it is not necessary to write the pointer bits with "0" before accessing WR0 or RR0. There are three pointer bits in WR0, and these allow access to the registers with addresses 0 through 7. Note that a command may be written to WR0 at the same time that the pointer bits are written.

To access the registers with addresses 8 through 15, a special command must accompany the pointer bits; WR0(4-3)=001. This precludes concurrently issuing a command when pointing to these registers. The register map for the ISCC in the non-multiplexed bus mode is shown in Table 2-4 below. If, for some reason, the state of the pointer bits is unknown, they may be reset to "0" by performing a read cycle of the SCC cell. Once the pointer bits have been set, the desired channel is selected by the state of the A1/A//B pin during the actual read or write of the desired SCC cell register.

Table 2-4. SCC Cell Register Address Map Using Pointer (Non-multiplexed Bus Mode)

Using Null Command				Using Point High Command			
A0/A//B	Address D2 D1 D0	Write Register	Read Register	A0/A//B	Address D2 D1 D0	Write Register	Read Register
0	000	WR0B	RR0B	0	000	WR8B	RR8B
0	001	WR1B	RR1B	0	001	WR9	RR13B
0	010	WR2	RR2B	0	010	WR10B	RR10B
0	011	WR3B	RR3B	0	011	WR11B	(RR15B)
0	100	WR4B	(RR0B)	0	100	WR12B	RR12B
0	101	WR5B	(RR1B)	0	101	WR13B	RR13B
0	110	WR6B	(RR2B)	0	110	WR14B	(RR10B)
0	111	WR7B	(RR3B)	0	111	WR15B	RR15B
1	000	WR0A	RR0A	1	000	WR8A	RR8A
1	001	WR1A	RR1A	1	001	WR9A	(RR13A)
1	010	WR2	RR2A	1	010	WR10A	RR10A
1	011	WR3A	RR3A	1	011	WR11A	(RR15A)
1	100	WR4A	(RR0A)	1	100	WR12A	RR12A
1	101	WR5A	(RR1A)	1	101	WR13A	RR13A
1	110	WR6A	(RR2A)	1	110	WR14A	(RR10A)
1	111	WR7A	(RR3A)	1	111	WR15A	RR15A

2.4.3 SCC Cell Register Reset

Table 2-5 lists the contents of the SCC cell registers after a hardware reset and after a channel reset.

Table 2-5. SCC Cell Reset Values

Register	Hardware Reset	Channel Reset
WR0	00000000	00000000
WR1	00x00x00	00x00x00
WR2	xxxxxxxx	xxxxxxxx
WR3	xxxxxxx0	xxxxxxx0
WR4	xxxxx1xx	xxxxx1xx
WR5	0xx0000x	0xx0000x
WR6	xxxxxxxx	xxxxxxxx
WR7	xxxxxxxx	xxxxxxxx
WR9	110000xx	xx0xxxxx
WR10	00000000	0xx00000
WR11	00001000	xxxxxxxx
WR12	xxxxxxxx	xxxxxxxx
WR13	xxxxxxxx	xxxxxxxx
WR14	xx100000	xx1000xx
WR15	11111000	11111000
RR0	01xxx100	01xxx100
RR1	00000110	00000110
RR3	00000000	00000000
RR10	00000000	00000000

2.4.4 DMA Cell Registers

The DMA cell contains seventeen registers counting the Bus Configuration Register. All of these registers are read/write except the Bus Configuration Register (write only), the Channel Command Address Register (write only), the DMA Status Register (read only), the Interrupt Command Register (write only), and the Interrupt Status Register (read only).

The reset content of all of the DMA registers identified in the address map is all zeros.

2.4.5 DMA Register Access, Multiplexed Bus

The registers in the ISCC in the multiplexed bus mode are addressed via the address on AD7-AD0 which is latched by the rising edge of /AS.

There are two address decoding modes: shift left and shift right. In shift left mode, the register address is decoded from AD5-AD1. This mode is set by a hardware reset. In shift right mode, the register address is decoded from AD4-AD0. The shift right / shift left selection for the DMA is located in the Bus Configuration Register, bit D0. When set, this bit programs the Shift Right mode for the DMA and when reset, this bit programs the Shift Left mode.

The address map for the DMA registers is shown in Table 2-6. This Table is also applicable to the non multiplexed bus mode.

Table 2-6. DMA Address Map

Address*	Name	Description
xxxxx	BCR	Bus Configuration Register
00000	CCAR	Channel Command / Address Register (Write)
00000	DSR	DMA Status (Read)
00001	ICR	Interrupt Control Register
00010	IVR	Interrupt Vector Register
00011	ICSR	Interrupt Command Register (Write)
00011	ISR	Interrupt Status Register (Read)
00100	DER	DMA Enable / Disable Register
00101	DCR	DMA Control Register
00110		Reserved Address
00111		Reserved Address
01000	RDCRA	Receive DMA Count Register, Channel A (Low Byte)
01001	RDCRA	Receive DMA Count Register, Channel A (High Byte)
01010	TDCRA	Transmit DMA Count Register, Channel A (Low Byte)
01011	TDCRA	Transmit DMA Count Register, Channel A (High Byte)
01100	RDCRB	Receive DMA Count Register, Channel B (Low Byte)
01101	RDCRB	Receive DMA Count Register, Channel B (High Byte)
01110	TDCRB	Transmit DMA Count Register, Channel B (Low Byte)
01111	TDCRB	Transmit DMA Count Register, Channel B (High Byte)
10000	RDARA	Receive DMA Address Register, Channel A (Bits 0-7)
10001	RDARA	Receive DMA Address Register, Channel A (Bits 8-15)
10010	RDARA	Receive DMA Address Register, Channel A (Bits 16-23)
10011	RDARA	Receive DMA Address Register, Channel A (Bits 24-31)
10100	TDARA	Transmit DMA Address Register, Channel A (Bits 0-7)
10101	TDARA	Transmit DMA Address Register, Channel A (Bits 8-15)
10110	TDARA	Transmit DMA Address Register, Channel A (Bits 16-23)
10111	TDARA	Transmit DMA Address Register, Channel A (Bits 24-31)
11000	RDARB	Receive DMA Address Register, Channel B (Bits 0-7)
11001	RDARB	Receive DMA Address Register, Channel B (Bits 8-15)
11010	RDARB	Receive DMA Address Register, Channel B (Bits 16-23)
11011	RDARB	Receive DMA Address Register, Channel B (Bits 24-31)
11100	TDARB	Transmit DMA Address Register, Channel B (Bits 0-7)
11101	TDARB	Transmit DMA Address Register, Channel B (Bits 8-15)
11110	TDARB	Transmit DMA Address Register, Channel B (Bits 16-23)
11111	TDARB	Transmit DMA Address Register, Channel B (Bits 24-31)

Note:

* Address in this Table is AD5-1 in the Multiplexed Bus with the Shift Left mode selected, AD4-0 in the Multiplexed Bus with the Shift Right mode selected, and D4 - D0 of the Channel Command / Address Register in the Non-multiplexed Bus mode.

2.4.6 DMA Register Access, Non-multiplexed Bus Mode

The registers in the DMA cell in the non-multiplexed bus mode are accessed in a two-step process, using a Register Pointer to perform the addressing. To access a particular register, the pointer bits must be set by writing to the Channel Command / Address Register bits 4 through 0. After the pointer bits are set, the next read or write cycle to the DMA cell will access the desired register. At the conclusion of this read or write cycle, the pointer bits are reset to "0s," so that the next access will be to the Channel Command / Address Register.

The fact that the pointer bits are reset to "0," unless explicitly set otherwise, means that the Channel Command /Address Register may be accessed in a single cycle. That is, it is not necessary to write the pointer bits with "0" before accessing the Channel Command / Address Register. This permits single access DMA enabling and resetting the highest IUS through the encoded DMA Commands.

2.4.7 Notes on Pointer Accesses

The non-multiplexed bus accesses are accomplished as described in the preceeding paragraphs using the DMA pointer for the DMA cell and the SCC cell pointer for channels A and B. These two pointers are completely independent. If one of these pointers is written to with a pointer value in preparation for a read or write to the selected register, the pointer will hold its value until the corresponding cell is accessed. For example, suppose the SCC cell pointer is written to in preparation to read an SCC cell register in the next (or even subsequent) software program steps. Before this SCC cell read takes place, a DMA interrupt occurs and the program enters the interrupt service routine prior to the SCC register read. In the interrupt service routine, several DMA register acceses are made. When the program exits the interrupt service routine and returns to the interrupted process, the register access to the SCC cell register proceeds correctly; the pointer was left unaltered. A converse situation is true for the DMA cell.

It should be clear, however, that if an interrupt routine is envoked between the pointer write and the register access, there can be conflict if the same cell is accessed in the interrupt service routine. Assume in the above example that the interrupt service routine accesses the SCC cell also. Since the pointer has already been written, a second write (the one in the interrupt service routine) will not write to the pointer in WR0 but will write to the pointed to register. Subsequent register access will also be incorrect. This suggests that the pointer write and subsequent register access be an uninterruptable pair and that the SCC Cell and DMA cell or the processor interrupts be disabled during the register access sequence.

CHAPTER 3
ISCC DMA AND ANCILLARY SUPPORT CIRCUITRY

3.1 INTRODUCTION

The most important feature of the ISCC other than SCC cell is the integrated, four channel DMA controller. As in the original SCC, the serial channels of the ISCC are supported by ancillary circuitry for generating clocks and performing data encoding and decoding. This chapter presents a description of these functional blocks.

3.2 DMA

The ISCC contains four independent DMA Channels, one for each receiver and transmitter. The DMA channels operate in fly-by mode; a 32 bit transfer address is generated along with the bus acquisition signals for executing the DMA transfer. Each DMA consists of a 32 bit address counter, a 16 bit (transfer) counter, and the required sequencing and control circuitry.

The DMA is set up by initializing the address resisters with the starting address of the DMA transfer and the count registers for the length of the block. Following this, the option to increment or decrement the address after a transfer is selected. Other DMA selections that must be programmed include the DMA priority, if separate bus requests are to be made for each DMA channel, the programming of the interrupt vector and the option to include interrupt status in the vector. Note that a no vector interrupt option is also possible. Following this, the Interrupt On Abort is programmed as desired, the individual channel interrupt enables are programmed, the Master Interrupt Enable is set (if interrupts are used), and lastly the appropriate DMA channels are enabled.

3.2.1 Receiver DMA Operation

Assuming the receiver has been appropriately set up, the DMA request will be made when the receive FIFO contains a byte and will continue to hold the bus and transfer bytes until the FIFO is empty. Once started, the DMA for the channel continues until the FIFO is empty even though a request from a higher priority DMA channel arises. Upon completion of the current DMA channel service, the next highest priority DMA channel commences its operation. The ISCC continues to hold the bus until all pending DMA requests have been served. Note that if the Bus Request Per Channel option has been selected, then the bus will be released and subsequently re-requested for each channel. At the completion of the block transfer (terminal count reached), an interrupt will be generated, if enabled. If selected, the interrupt vector will indicate the interrupt source according to Table 3-1.

Table 3-1. DMA Interrupt Vector Modification

IV3	IV2	IV1	Interrupt Source
0	0	0	No Interrupt Pending
0	0	1	Not Possible
0	1	0	Not Possible
0	1	1	Not Possible
1	0	0	Rx A Interrupt Pending
1	0	1	Rx B Interrupt Pending
1	1	0	Tx A Interrupt Pending
1	1	1	Tx B Interrupt Pending

An Interrupt Pending only modifies the interrupt vector if the corresponding Interrupt Enable bit is set. Note that software may have to test status bits to determine if the channel interrupt is due to terminal count or an abort.

When the receive DMA enable bit is set, a DMA request is made if the receive FIFO contains a character at the time, or no request will be made until a character enters the receive FIFO. Note that DMA requests will follow the state of the receive FIFO even though the receiver is disabled. Thus, if the receiver is disabled and the DMA is still enabled, the DMA will transfer the previously received data correctly. In this mode the DMA requests directly follow the state of the receive FIFO. This operation is essentially equivalent to the DMA requests following the state of the Receive Character Available bit in the SCC cell in Read Register 0.

The SCC cell will not generate a DMA request in the case of a special receive condition in the Receive Interrupt on First Character or Special Condition mode, or the Receive Interrupt on Special Condition Only mode.

In these two interrupt modes any receive character with a special receive condition is locked at the top of the FIFO until an Error Reset command is issued. This character in the receive FIFO would ordinarily cause additional DMA Requests after the first time it is read. However, the logic in the SCC cell guarantees no extra DMA transfers by terminating DMA requests after the time the character with the special receive condition is read, and the FIFO locked. DMA requests are held off until after the Error Reset command has been issued.

Once the FIFO is locked, it allows the checking of the Receive Error FIFO (RR1) to find the cause of the error. Locking the data FIFO therefore, will stop the error status from popping out of the Receive Error FIFO. Also, since DMA request will become inactive, the interrupt (Special Condition) can be serviced. Once the FIFO is unlocked by the Error Reset command, DMA requests again follow the state of the receive FIFO.

3.2.2 Transmitter DMA Operation

With the DMA enabled, the status of an empty transmitter FIFO triggers the DMA to request the bus and begin DMA transfer to the transmit FIFO. Once this DMA channel is selected for service, DMA transfers continue until the transmit FIFO is full (or until terminal count is reached if there are not enough bytes remaining to fill the FIFO). Once started, the DMA for the channel continues until the FIFO is full even though a request from a higher priority DMA channel arises. Upon completion of the current DMA channel service, the next highest priority DMA channel commences its operation. The ISCC continues to hold the bus until all pending DMA requests have been served. Note that if the Bus Request Per Channel option has been selected, then the bus will be released and subsequently re-requested for each channel. At the completion of the block transfer (terminal count reached), an interrupt will be generated, if enabled. If selected, the interrupt vector will indicate the interrupt source according to Table 3-1.

An Interrupt Pending only modifies the interrupt vector if the corresponding Interrupt Enable bit is set. Note that software may have to test status bits to determine if the channel interrupt is due to terminal count or an abort.

Note that the DMA request will follow the state of the transmit FIFO even though the transmitter is disabled. Thus, if the DMA is enabled, the DMA may write data to the SCC cell before the transmitter is enabled. This will not cause a problem in Asynchronous mode but may cause problems in Synchronous mode because the ISCC will send data in preference to flags or sync characters. Thus a data character in the transmit FIFO may get transmitted prior to the frame sync character or opening flag. It may also complicate the CRC initialization, which cannot be done until after the transmitter is enabled. DMA requests essentially follow the Tx Buffer Empty bit in the SCC cell Read Register 0.

3.3 BAUD RATE GENERATOR

The Baud Rate Generator (BRG) is essential for asynchronous communications. Each channel in the ISCC contains a programmable baud rate generator. Each generator consists of two 8-bit, time-constant registers forming a 16-bit time constant, a 16-bit down counter, and a flip-flop on the output that makes the output a square wave. On start-up, the flip-flop on the output is set High, so that it starts in a known state, the value in the time-constant register is loaded into the counter, and the counter begins counting down. When a count of zero is reached, the output of the baud rate generator toggles, the value in the time-constant register is loaded into the counter, and the process starts over. A block diagram of the baud rate generator is shown in Figure 3-1.

The time-constant can be changed at any time, but the new value does not take effect until the next load of the counter (i.e., after zero count is reached).

No attempt is made to synchronize the loading of a new time-constant with the clock used to drive the generator. When the time-constant is to be changed, the generator should be stopped first by writing to an enable bit in WR14. After loading the time constant, the BRG can be started again. This ensures the loading of a correct time constant, but loading will not be taking place until zero count or a reset occurs.

If neither the transmit clock nor the receive clock are programmed to come from the /TRxC pin, the output of the baud rate generator may be made available for external use on the /TRxC pin.

The clock source for the baud rate generator is selected by bit D1 of WR14. When this bit is set to "0," the baud rate generator uses the signal on the /RTxC pin as its clock, independent of whether the /RTxC pin is a simple input or

part of the crystal oscillator circuit. When this bit is set to "1," the baud rate generator is clocked by PCLK. To avoid metastable problems in the counter, this bit should be changed only while the baud rate generator is disabled, since arbitrarily narrow pulses can be generated at the output of the multiplexer when it changes state.

The BRG is enabled while bit D0 of WR14 is set to 1 and disabled while this bit is set to 0 and it is disabled after a hardware reset. To prevent metastable problems when the baud rate generator is first enabled, the enable bit is synchronized to the baud rate generator clock. This introduces an additional delay when the baud rate generator is first enabled. This is shown in Figure 3-2. The baud rate generator is disabled immediately when bit D0 of WR14 is set to "0," because the delay is only necessary on start-up. The baud rate generator may be enabled and disabled on the fly, but this delay on start-up must be taken into consideration.

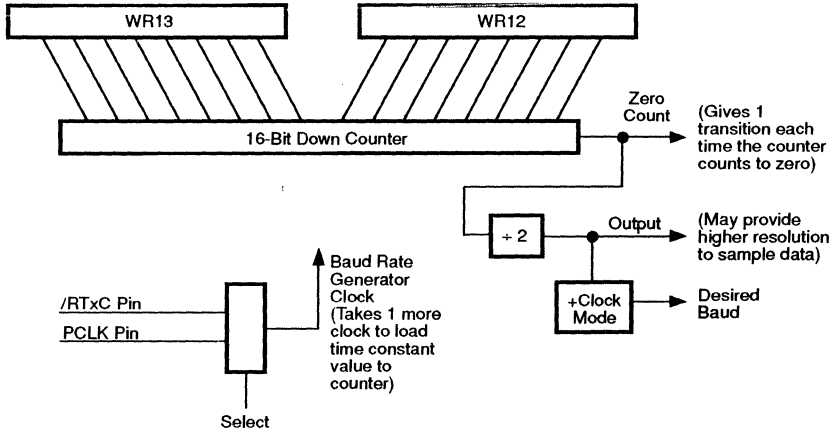


Figure 3-1. Baud Rate Generator

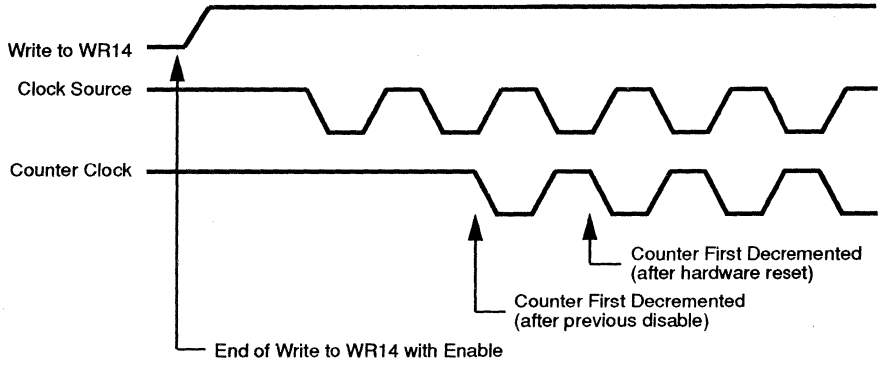


Figure 3-2. Baud Rate Generator Start Up

The formulas relating the baud rate to the time-constant and vice versa are shown below. The clock mode in the formula is the ratio of the receive clock applied to the ISCC relative to the data rate. The ISCC may be programmed to accept a receive clock that is one, sixteen, thirty-two, or sixty-four times the data rate (refer to the description of WR4 and the descriptions in Chapter 4).

$$\text{Time Constant} = \frac{\text{Clock Frequency}}{2*(\text{Clock Mode})*(\text{Baud Rate})} - 2$$

$$\text{Baud Rate} = \frac{\text{Clock Frequency}}{2*(\text{Clock Mode})*(\text{Time Constant} + 2)}$$

In these formulas, the baud rate generator clock frequency (PCLK or /RTxC) is in Hertz, the desired baud rate in bits/second and the time constant is dimensionless. The example in Table 3-1 assumes a 2.4576 MHz clock (from /RTxC) clock factor of 16 and shows the time constant for a number of popular baud rates.

For example:

$$TC = \frac{2.4576 \times 10^6}{2 \times 16 \times 150} = 510$$

Table 3-2. Baud Rates for 2.4576 MHz Clock and 16x Clock Factor

Time Constant		Baud Rate
Decimal	Hex	
0	0000	38400
2	0002	19200
6	0006	9600
14	000E	4800
30	001E	2400
62	003E	1200
126	007E	600
254	00FE	300
510	01FE	150

Initializing the baud rate generator is done in three steps. First, the time-constant is determined and loaded into WR12 and WR13. Next, the processor must select the clock source for the baud rate generator by setting bit D1 of WR14. Finally, the baud rate generator is enabled by setting bit D0 of WR14 to "1."

Note that the first write to WR14 is not necessary after a hardware reset if the clock source is the /RTxC pin. This is because a hardware reset automatically selects the /RTxC pin as the baud rate generator clock source.

3.4 DATA ENCODING/DECODING

The ISCC provides four different data encoding methods, selected by bits D6 and D5 in WR10. An example of these four encoding methods is shown in Figure 3-3. Any encoding method may be used in any X1 mode in the ISCC, asynchronous or synchronous. The data encoding selected is active even though the transmitter or receiver may be idling or disabled. The data encoding methods are shown in Figure 3-3.

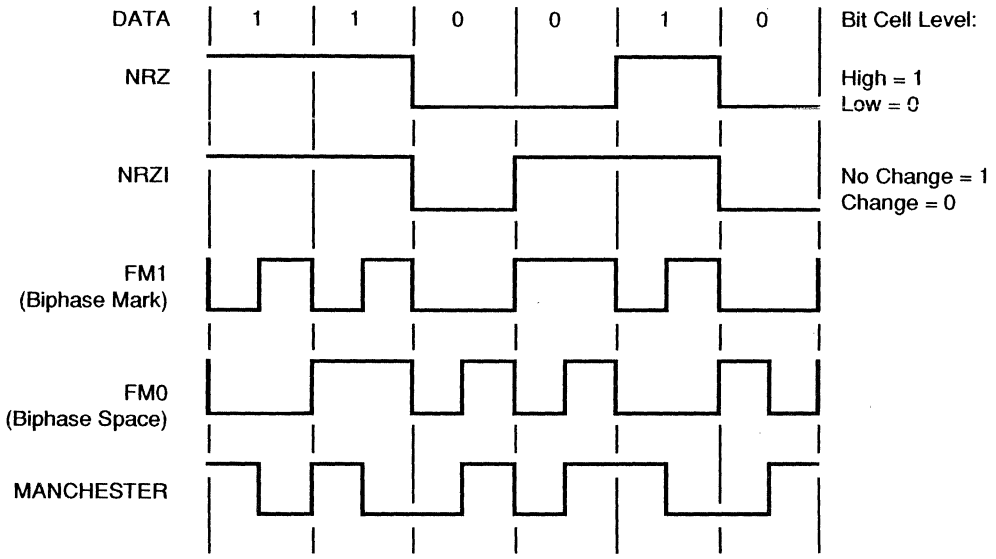


Figure 3-3. Data Encoding Methods

In NRZ, encoding a "1" is represented by a HIGH level and a "0" is represented by a LOW level. In this encoding method, only a minimal amount of clocking information is available in the data stream in the form of transitions on bit-cell boundaries. In an arbitrary data pattern, this may not be sufficient to generate a clock for the data from the data itself.

In NRZI, encoding a "1" is represented by no change in the level and a "0" is represented by a change in the level. As in NRZ, only a minimal amount of clocking information is available in the data stream, in the form of transitions on bit cell boundaries. In an arbitrary data pattern this may not be sufficient to generate a clock for the data from the data itself. In the case of SDLC, where the number of consecutive "1s" in the data stream is limited, a minimum number of transitions to generate a clock are guaranteed.

In FM1 encoding, also known as biphase mark, a transition is present on every bit cell boundary, and an addition transition may be present in the middle of the bit cell. In FM1 a "0" is sent as no transition in the center of the bit cell and a "1" is sent as a transition in the center of the bit cell. FM1 encoded data contains sufficient information to recover a clock from the data.

In FM0 encoding, also known as biphase space, a transition is present on every bit cell boundary and an additional transition may be present in the middle of the bit cell. In FM0, a "1" is sent as no transition in the center of the bit cell and a "0" is sent as a transition in the center of the bit cell. FM0 encoded data contains sufficient information to recover a clock from the data.

Manchester encoding, which is not directly supported, always produces a transition at the center of the bit cell. If the transition is Low to High, the bit is "0." If the transition is High to Low, the bit is "1." ISCC can be used to decode Manchester (biphase level) data by using the DPLL in the FM mode and programming the receiver for NRZ data. (See section 3.5.3)

The data encoding method should be selected in the initialization procedure before the transmitter and receiver are enabled, but no other restrictions apply. Note, in Figure 3-3, that in NRZ and NRZI the receiver samples the data only on one edge. However, in FM1 and FM0 the receiver samples the data on both edges. Also, as shown in Figure 6-4, the transmitter defines bit cell boundaries by one edge in all cases and uses the other edge in FM1 and FM0 to create the mid-bit transition.

3.5 DIGITAL PHASE-LOCKED LOOP (DPLL)

Each channel of the SCC cell contains a digital phase-locked loop that can be used to recover clock information from a data stream with NRZI, FM or NRZ encoding. The DPLL is driven by a clock nominally 32 (NRZI) or 16 (FM) times the data rate. The DPLL uses this clock, along with the data stream, to construct a receive clock for the data. This clock can then be used as the ISCC receive clock, the transmit clock, or both.

Figure 3-4 shows a block diagram of the digital phase-locked loop. It consists of a 5-bit counter, an edge detector, and a pair of output decoders. The clock for the DPLL comes from the output of a two-input multiplexer, and the two outputs go to the transmitter and receive clock multiplexers. The DPLL is controlled by the seven commands that are encoded in bits D7, D6 and D5 of WR14.

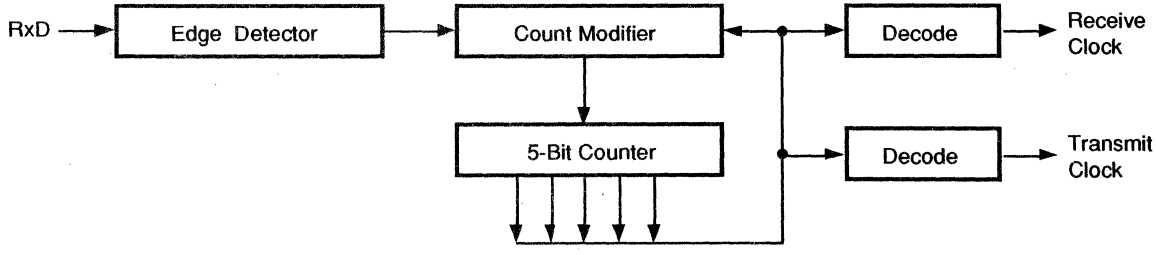


Figure 3-4. Digital Phase Lock Loop

The clock for the DPLL is selected by two of the commands in WR14, that is:

WR14 (7-5) = 100	BRG Clock Source
WR14 (7-5) = 101	/RTxC Pin Clock Source

The first command selects the baud rate generator as the clock source. The other command selects /RTxC pin as the clock source, independent of whether the /RTxC pin is a simple input or part of the crystal oscillator circuit.

Initialization of the DPLL may be done at any time during the initialization sequence, but should preferably be done after the clock modes have been selected in WR11, and before the receiver and transmitter are enabled. When initializing the DPLL, the clock source should be selected first, followed by the selection of the operating mode.

To avoid metastable problems in the counter, the clock source selection should be made only while DPLL is disabled, since arbitrarily narrow pulses can be generated at the output of the multiplexer when it changes status.

The DPLL is enabled by issuing the Enter Search Mode command in WR14; that is WR14 (7-5) = 001. The Enter Search Mode command unlocks the counter, which is held while the DPLL is disabled, and enables the edge detector. If the DPLL is already enabled when this command is issued, the DPLL also enters Search Mode.

Enter Search Mode is also used to reset the DPLL to a known state if it is suspected that synchronization has been lost. Note that the DPLL and the receiver are independent, so whether the receiver is disabled or not enabled, DPLL will sample whatever is on the RxD line.

DPLL requires a transition in every bit cell, and if this transition is not present in two consecutively sampled bit cells, the DPLL will automatically enter search mode and the DPLL will not provide any clock output.

In Search mode, the counter is held at a specific count and no outputs are provided. The DPLL remains in this status until an edge is detected in the receive data stream. This first edge is assumed to occur on a bit cell boundary, and the DPLL will begin providing an output to the receiver that will properly sample the data. From this point on the DPLL will adjust its output to remain in phase with the receive data. If the first edge that the DPLL sees does not occur on a bit cell boundary, the DPLL will eventually lock on to the receive data, but it will take longer to do so.

The DPLL may be programmed to operate in either of two modes, as selected by command in WR14.

WR14 (7-5) = 111 for NRZI mode and
WR14 (7-5) = 110 for FM mode

Note that a channel or hardware reset disables the DPLL, selects the /RTxC pin as the clock source for the DPLL, and places it in the NRZI mode.

As in the case of the clock source selection, the mode of operation should only be changed while the DPLL is disabled to prevent unpredictable results.

In the NRZI mode, the DPLL clock must be 32 times the data rate. In this mode, the transmit and receive clock outputs of the DPLL are identical, and the clocks are phased so that the receiver samples the data in the middle of the bit cell. In NRZI mode, the DPLL does not require a transition in every bit cell, so this mode is useful for recovering the clocking information from NRZ and NRZI data streams.

In the FM mode, the DPLL clock must be 16 times the data rate. In this mode the transmit clock output of the DPLL lags the receive clock outputs by 90 degrees to make the transmit and receive bit cell boundaries the same, because the receiver must sample FM data at one-quarter and three-quarters bit time.

3.5.1 DPLL Operation in the NRZI Mode

To operate in NRZI mode, the DPLL must be supplied with a clock that is 32 times the data rate. The DPLL uses this clock, along with the receive data, to construct receive and transmit clock outputs that are phased to properly receive and transmit data.

To do this, the DPLL divides each bit cell into four regions, and makes an adjustment to the count cycle of the 5-bit counter dependent upon in which region a transition on the receive data input occurred. This is shown in Figure 3-5.

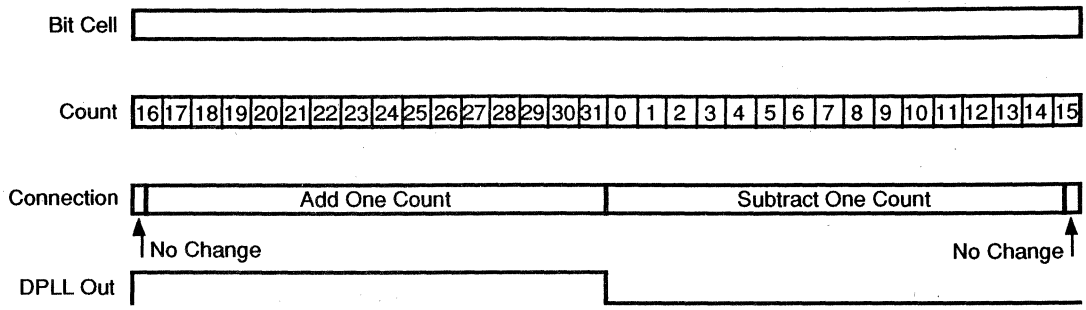


Figure 3-5. DPLL in NRZI Mode

Ordinarily, a bit cell boundary will occur between count 15 and count 16, and the DPLL output will cause the data to be sampled in the middle of the bit cell. However, four different situations may happen:

The DPLL actually allows the transition marking a bit cell boundary to occur anywhere during the second half of count 15 or the first half of count 16 without making a correction to its count cycle.

If the transition marking a bit cell boundary occurs between the middle of count 16 and count 31, the DPLL is sampling the data too early in the bit cell. In response to this, the DPLL extends its count by one during the next 0 to 31 counting cycle, which effectively moves the edge of the clock that samples the receive data closer to the center of the bit cell.

If the transition occurs between count 0 and the middle of count 15, the output of the DPLL is sampling the data too late in the bit cell. To correct this, the DPLL shortens its count by one during the next 0 to 31 counting cycle, which effectively moves the edge of the clock that samples the receive data closer to the center of the bit cell.

If the DPLL does not see any transition during a counting cycle, no adjustment is made in the following counting cycle.

If an adjustment to the counting cycle is necessary, the DPLL modifies count 5, either deleting it or doubling it. Thus, only the LOW time of the DPLL output will be lengthened or shortened.

While the DPLL is in search mode, the counter remains at count 16 where the DPLL outputs are both HIGH. The missing clock latches in the DPLL which may be accessed in RR10. They are not used in NRZI mode. An example of the DPLL in operation is shown in Figure 3-6.

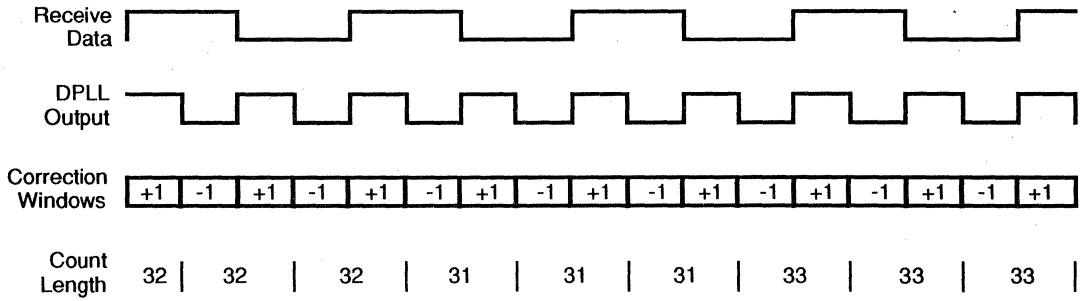


Figure 3-6. DPLL Operating Example (NRZI Mode)

3.5.2 DPLL Operation in the FM Modes

To operate in FM mode, the DPLL must be supplied with a clock that is 16 times the data rate. The DPLL uses this clock, along with the receive data, to construct receive and transmit clock outputs that are phased to receive and transmit data properly.

In FM mode one cycles of the counter in the DPLL is a count from 0 to 31, but now each cycle corresponds to 2-bit cells. To make adjustments to remain in phase with the receive data, the DPLL divides a pair of bit cells into 5 regions, making the adjustment to the counter dependent upon which region the transition on the receive data input occurred. This is shown in Figure 3-7.

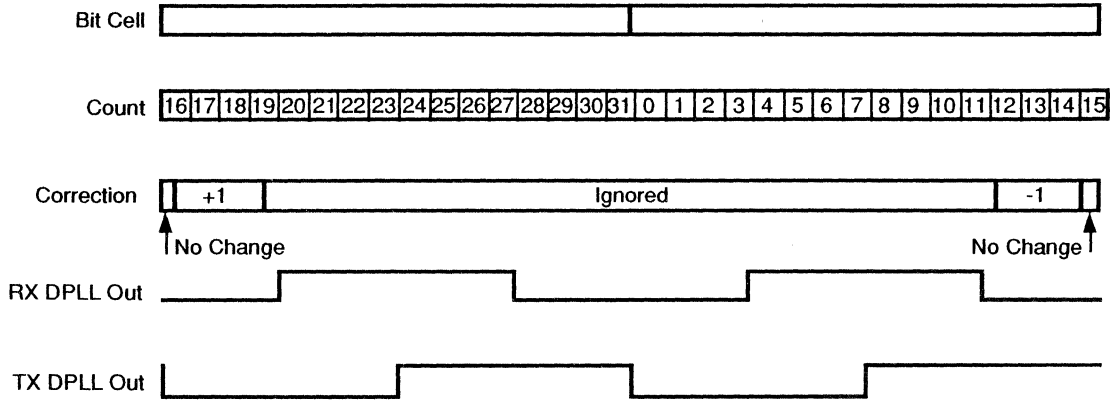


Figure 3-7. DPLL Operation in the FM Mode

In FM mode, the transmit clock and receive clock outputs from the DPLL are not in phase. This is necessary to make the transmit and receive bit cell boundaries coincide, since the receive clock must sample the data one-fourth and three-fourths of the way through the bit cell.

Ordinarily, a bit cell boundary will occur between count 15 or count 16, and the DPLL receive output will cause the data to be sampled at one-fourth and three-fourths of the way through the bit cell.

However, four variations may happen:

1. The DPLL actually allows the transition marking a bit-cell boundary to occur anywhere during the second half of count 15 or the first half of count 16, without making a correction to its count cycle.

2. If the transition marking a bit cell boundary occurs between the middle of count 16 and the middle of count 19, the DPLL is sampling the data too early in the bit cell. In response to this, the DPLL extends its count by 1 during the next 0 to 31 counting cycle, which effectively moves the receive clock edges closer to where they should be.

Any transitions occurring between the middle of count 19 in one cycle and the middle of count 12 during the next cycle are ignored by the DPLL. This is necessary to guarantee that any data transitions in the bit cells will not cause an adjustment to the counting cycle.

3. If no transition occurs between the middle of count 12 and the middle of count 19, the DPLL is probably not locked onto the data properly. When the DPLL misses an edge, the One Clock Missing bit is RR10, it is set to "1" and latched. It will hold this value until a Reset missing Clock command is issued in WR14 or until the DPLL is disabled or programmed to enter the Search mode. Upon missing this one edge, the DPLL takes no other action and does not modify its count during the next counting cycle.

4. If the DPLL does not see an edge between the middle of count 12 and the middle of count 19 in two successive 0 to 31 count cycles, a line error condition is assumed. If this occurs, the Two Clocks Missing bit in RR10 is set to "1" and latched. At the same time, the DPLL enters the Search mode. The DPLL makes the decision to enter Search mode during count 2, where both the receive clock and transmit clock outputs are LOW. This prevents any glitches on the

clock outputs when search mode is entered. While in search mode, no clock outputs are provided by the DPLL. The Two Clocks Missing bit in RR10 is latched until a Reset Missing Clock command is issued in WR14, or until the DPLL is disabled or programmed to enter the Search mode.

While the DPLL is disabled, the transmit clock output of the DPLL may be toggled by alternately selecting FM and NRZI move in the DPLL. The same is true of the receive clock.

While the DPLL is in Search mode, the counter remains at count 16, where the receive output is LOW and the transmit output is LOW. This fact can be used to provide a transmit clock under software control since the DPLL is in Search mode while it is disabled.

As in NRZI mode, if an adjustment to the counting cycle is necessary, the DPLL modifies count 5, either deleting it or doubling it. If no adjustment is necessary, the count sequence proceeds normally.

From the above discussion, together with an examination of FM0 and FM1 data encoding, it should be obvious that only clock transitions should exist on the receive data pin when the DPLL is programmed to enter search mode. If this is not the case, the DPLL may attempt to lock on to the data transitions.

With FM0 encoding this requires continuous "1s" received when leaving Search. In FM1 encoding, it is continuous "0s"; with Manchester encoded data this means alternating "1s" and "0s." With all three of these data encoding methods there will always be at least one transition in every bit cell, and in FM mode the DPLL is designed to expect this transition.

3.5.3 DPLL Operation and Encoding in the Manchester Mode

The ISCC can encode Manchester data using the external logic shown in Figure 3-8, and it can decode Manchester data using the DPLL. Recall that Manchester encoded data contains a transition at the center of every bit cell; it is the direction of this transition that distinguishes a "1" from a "0." Hence, for Manchester data, the DPLL should be in FM mode, but the receiver should be set up to accept NRZ data. As with the FM modes, when in the Search Mode the data stream should contain only clock transitions.

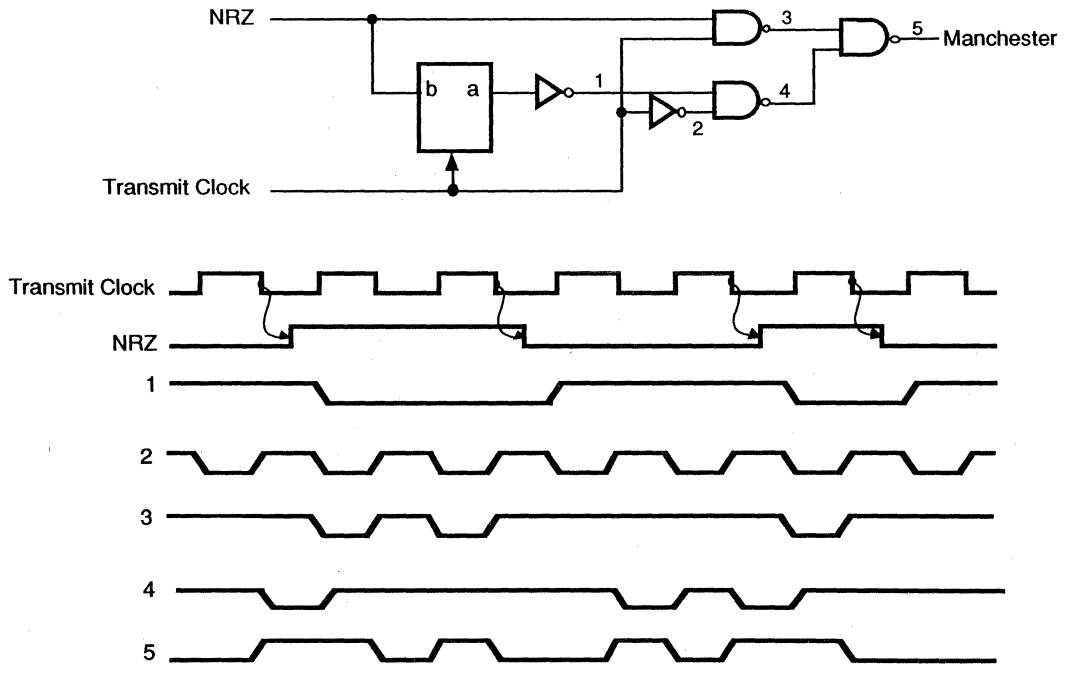


Figure 3-8. Encoding Manchester Data

3.6 CLOCK SELECTION

The ISCC can select several clock sources for internal and external use. Write Register 11 is the Clock Mode Control register for both the receive and transmit clocks. It determines the type of signal on the /SYNC and /RTxC pins and the direction of the /TRxC pin.

The ISCC may be programmed to select one of several sources to provide the receive and receive clocks.

The source of the receive clock is controlled by bits D6 and D5 of WR11. The receive clock may be programmed to come from the /RTxC pin, the /TRxC pin, the output of the baud rate generator, or the receive output of the DPLL.

The source of the transmit clock is controlled by bits D4 and D3 of WR11. The transmit clock may be programmed to come from the /RTxC pin, the /TRxC pin, the output of the baud rate generator, or the transmit output of the DPLL.

Ordinarily the /TRxC pin is an input, but it becomes an output if this pin has not been selected as the source for the transmitter or the receiver, and bit D2 of WR11 is set to "1." The selection of the signal provided on the /TRxC output pin is controlled by bits D1 and D0 of WR11. The /TRxC pin may be programmed to provide the output of the crystal oscillator, the output of the baud rate generator, the receive output of the DPLL or the actual transmit clock. If the output of the crystal oscillator is selected, but the crystal oscillator has not been enabled, the /TRxC pin will be driven HIGH. The option of placing the transmit clock signal on the /TRxC pin when it is an output allows access to the transmit output of the DPLL.

Figure 3-9 shows a simplified schematic diagram of the circuitry used in the clock multiplexing. It shows the inputs to the multiplexer section, as well as the various signal inversions that occur in the paths to the outputs.

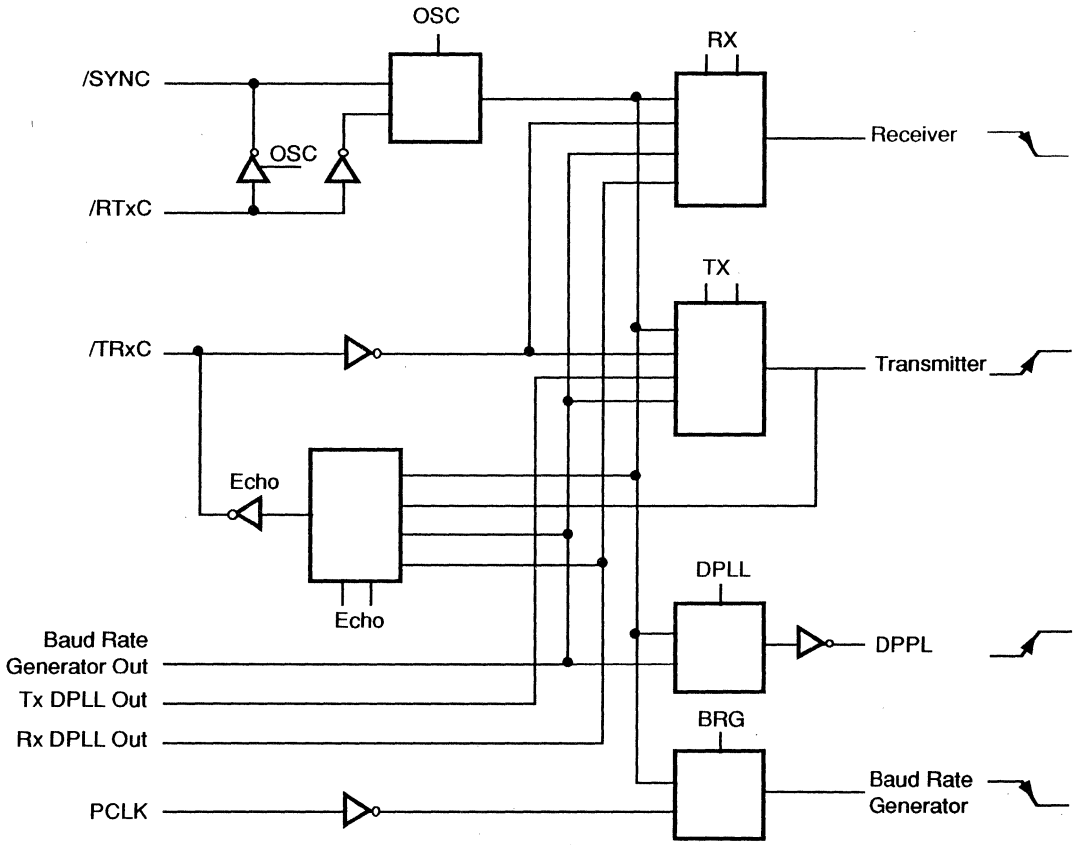


Figure 3-9. Clock Multiplexer

Selection of the clocking options may be done anywhere in the initialization sequence, but the final values must be selected before the receiver, transmitter, baud rate generator, or DPLL are enabled to prevent problems from arbitrarily narrow clock signals out of the multiplexers. The same is true of the crystal oscillator, in that the output should be allowed to stabilize before it is used as a clock source.

Also shown are the edges used by the receiver, transmitter, baud rate generator and DPLL to sample or send data or otherwise change state. For example, the receiver samples data on the falling edge, but since there is an inversion in the clock path between the /RTxC pin and the receiver, a rising edge of the /RTxC pin samples the data for the receiver.

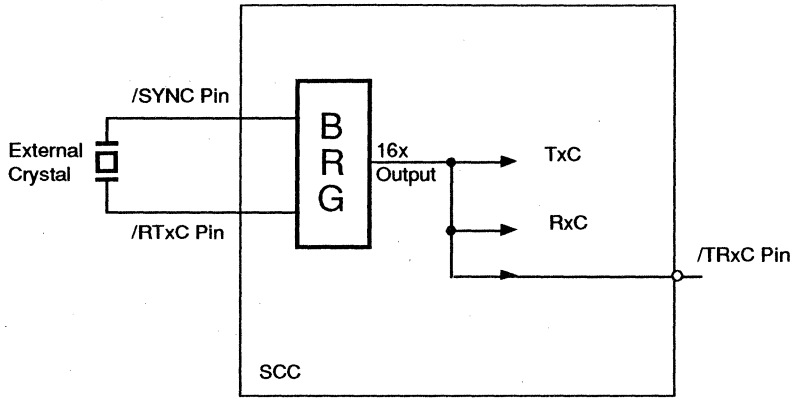


Figure 3-10a. Async Transmission, 16x Clock Mode Using External Crystal

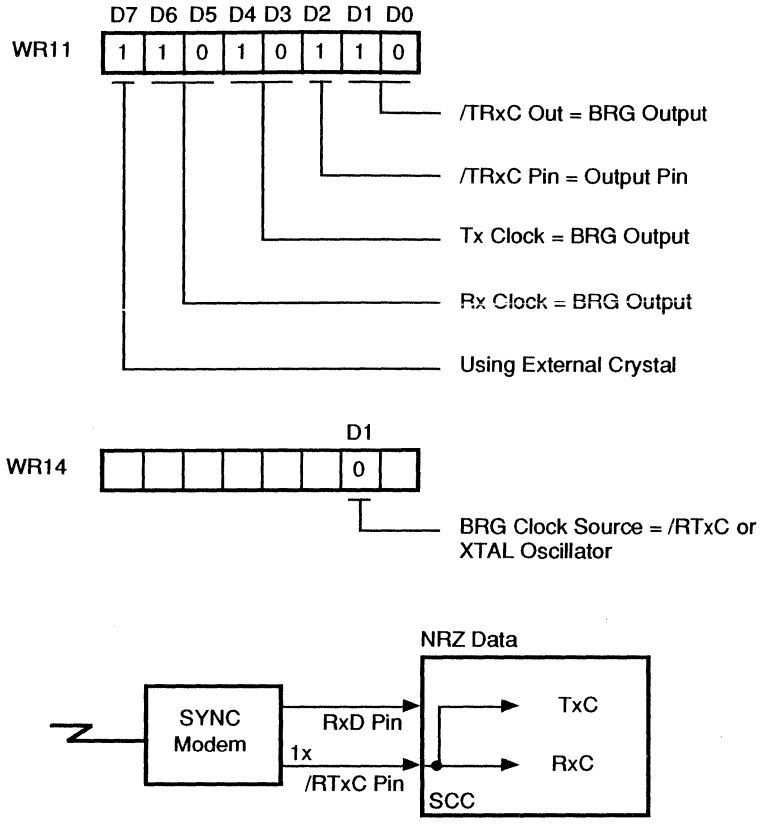


Figure 3-10b. Async Transmission, 1x Clock Rate, NRZ Data Encoding

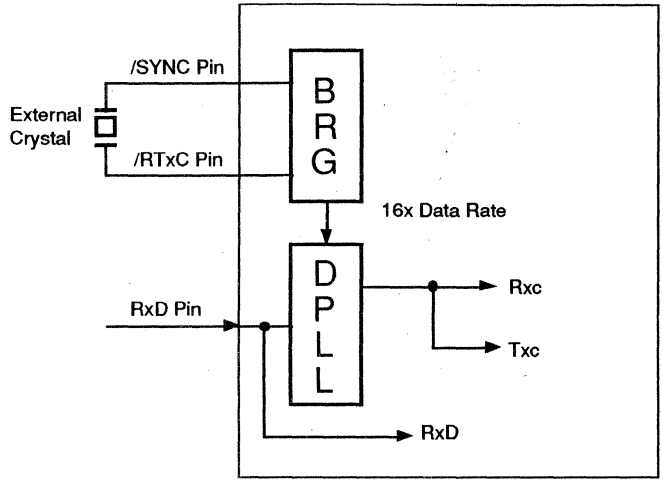


Figure 3-10c. Asynchronous Transmission, 1x Clock Rate, FM Data encoding

Figure 3-10 shows three examples of clock sources and selection. Part (a) of the figure shows the clock set up for asynchronous transmission, 16x clock mode using the on chip oscillator with an external crystal. The registers involved are WR11 and WR14 and the figure shows the programming in these registers. Part (b) of the figure shows asynchronous communication where a 1x clock is obtained from an external MODEM. The data encoding is NRZ.

Note that: The BRG is not used under this configuration.

The x1 mode in Asynchronous mode is a combination of both synchronous and asynchronous transmission. The data are clocked by a common timing base, but characters are still framed with Start and Stop bits. Because the receiver waits for one clock period after detecting the first High-to-Low transition before beginning to assemble characters, the data and clock must be synchronized externally. The x1 mode is the only mode in which a data encoding method other than NRZ may be used.

Part (c) of Figure 3-10 shows the use of the DPLL to derive a 1x clock from the data. In this example:

- The DPLL clock input = BRG output (x16 the data rate) WR14.
- The DPLL clock output = RxC (receiver clock) WR11.
- Set FM mode WR14.
- Set FM mode WR10.

3.7 CRYSTAL OSCILLATORS

For a given channel, if bit D7 of WR11 is set to 1, the crystal oscillator is enabled and a high-gain amplifier is connected between the /RTxC pin and the /SYNC pin. While the crystal oscillator is enabled, anything that has selected /RTxC as its clock source will automatically be connected to the output of the crystal oscillator. This also makes the /SYNC pin unavailable for other use.

In synchronous modes, no sync pulse is output, and the External Sync mode cannot be selected. In asynchronous modes, the state of the Sync/Hunt bit in RR0 is no longer controlled by the /SYNC pin. Instead, the Sync/Hunt bit is forced to "0." The crystal oscillator requires some finite time to stabilize and must be allowed to stabilize before it is used as a clock source. The External Crystal used should operate in parallel resonance.

CHAPTER 4
DATA COMMUNICATION MODES

4.1 INTRODUCTION

The ISCC provides two independent full-duplex channels programmable for use in any common asynchronous or synchronous data communication protocols. The data communication protocols handled by the SCC cell within the ISCC are:

- Asynchronous mode
- Character-Oriented mode
- Monosynchronous
- Bisynchronous
- External Synchronous
- Bit-Oriented mode
- SDLC
- SDLC Loop Mode

4.1.1 General Description of the Transmitter

A block diagram of the transmitter is given in Figure 4-1. The transmitter has an 8-bit Transmit Data register (WR8) loaded from the internal data bus and a Transmit Shift register loaded from either WR6, WR7, or the Transmit Data register. In byte-oriented modes, WR6 and WR7 can be programmed with sync characters. In Monosync mode, an 8-bit or 6-bit sync character is used (WR6), whereas a 16-bit sync character is used (WR6 and WR7) in Bisync mode. In bit-oriented synchronous modes, the flag contained in WR7 is loaded into the Transmit Shift register at the beginning and end of a message.

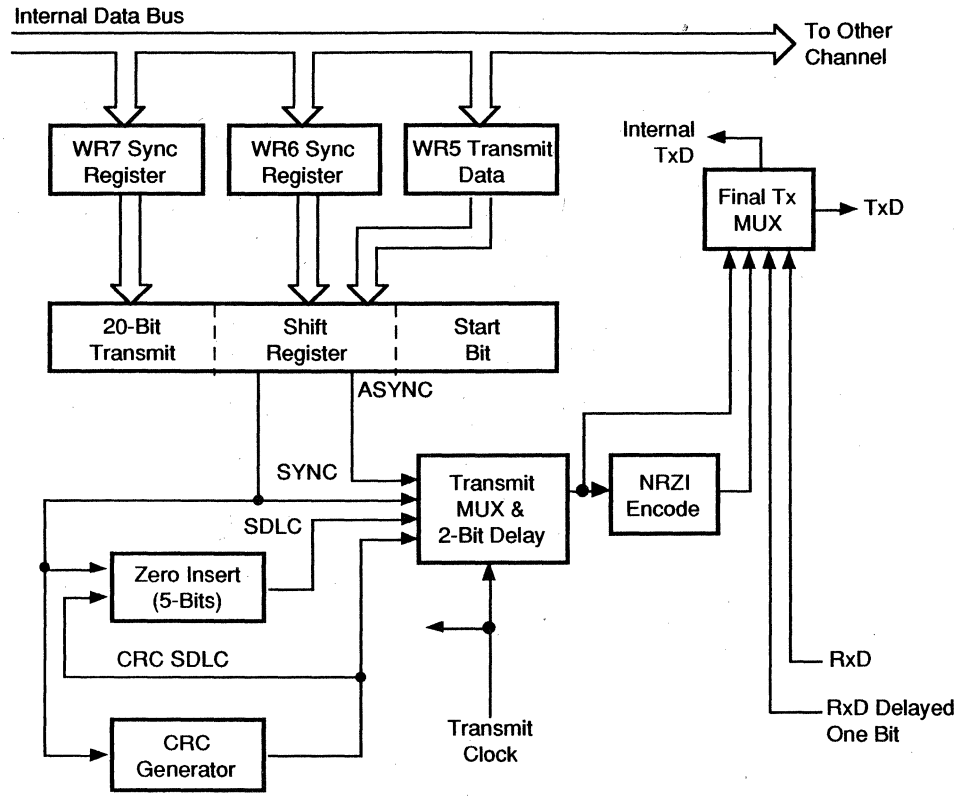


Figure 4-1. Transmitter Block Diagram

If asynchronous data is processed, WR6 and WR7 are not used and the Transmit Shift register is formatted with start and stop bits shifted out to the transmit multiplexer at the selected clock rate. Synchronous data (except SDLC/HDLC) is shifted to the CRC generator as well as to the transmit multiplexer.

SDLC/HDLC data is shifted to the CRC Generator and out through the zero insertion logic (which is disabled while the flags are being sent). A "0" is inserted in all address, control, information, and frame check fields following five contiguous "1s" in the data stream. The result of the CRC generator for SDLC data is also routed through the zero insertion logic and then to the transmit multiplexer.

4.1.2 General Description of the Receiver

The receiver has a three deep, 8-bit Data FIFO (paired with a three deep Error FIFO), and an 8-bit shift register. The receiver block diagram is shown in Figure 4-2. This arrangement creates a 3-character delay time, which allows the CPU time to service an interrupt at the beginning of a block of high-speed data. With each Receive Data FIFO, the Error FIFO stores parity and framing errors and other types of status information. The Error FIFO is readable in Read Register 1.

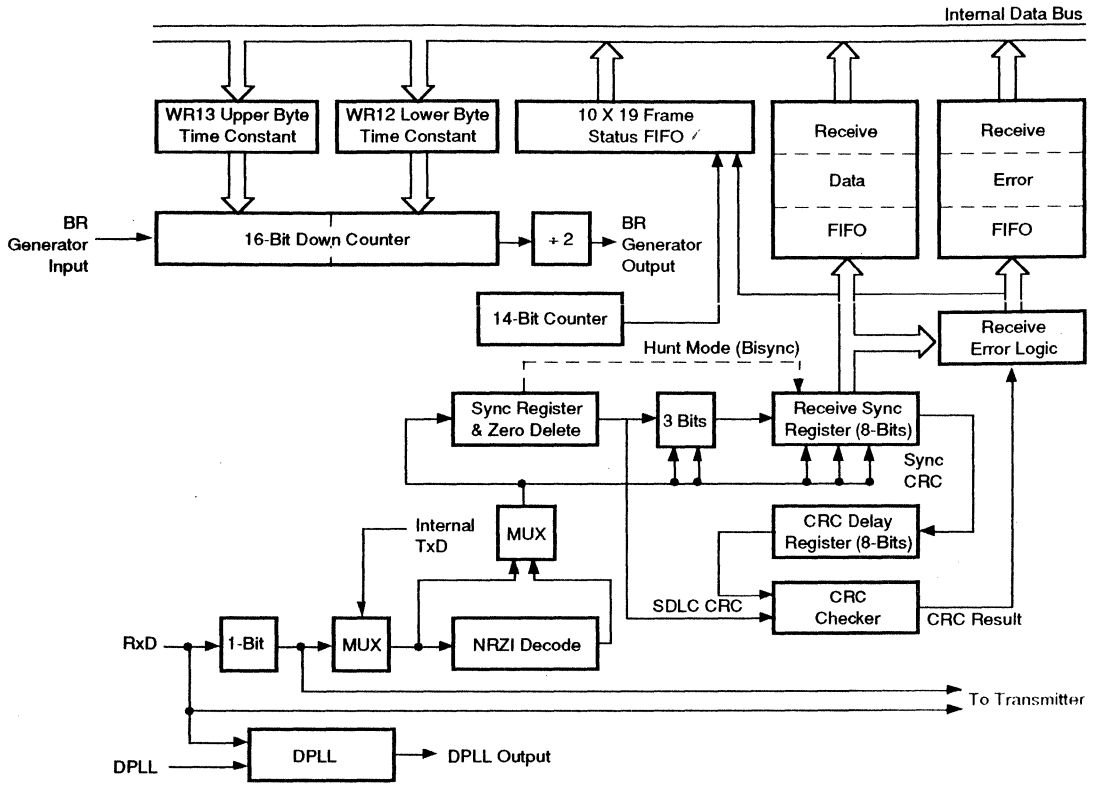


Figure 4-2. Receiver Block Diagram

Incoming data is routed through one of several paths depending on the mode and character length. In Asynchronous mode, serial data enters the 3-bit delay if the character length of seven or eight bits is selected. If a character length of five or six bits is selected, data enters the receive shift register directly.

In synchronous modes, the data path is determined by the phase of the receive process currently in operation. A synchronous receive operation begins with a hunt phase in which a bit pattern that matches the programmed sync characters (6-,8-, or 16-bit is searched).

The incoming data then passes through the Sync register and is compared to a sync character stored in WR6 or WR7 (depending on which mode it is in). The Monosync mode matches the sync character programmed in WR7 and the character assembled in the Receive Sync register to establish synchronization.

Synchronization is achieved differently in the Bisync mode. Incoming data is shifted to the Receive Shift register while the next eight bits of the message are assembled in the Receive Sync register. If these two characters match the programmed characters in WR6 and WR7, synchronization is established. Incoming data can then bypass the Receive Sync register and enter the 3-bit delay directly.

The SDLC mode of operation uses the receive Sync register to monitor the receive data stream and to perform zero deletion when necessary; i.e., when five continuous "1s" are received, the sixth bit is inspected and deleted from the data stream if it is "0". The seventh bit is inspected only if the sixth bit equals one. If the seventh bit is "0", a flag sequence has been received and the receiver is synchronized to that flag. If the seventh bit is a "1" an abort or an EOP (End Of Poll) is recognized, depending upon the selection of either the normal SDLC mode or SDLC Loop mode.

The same path is taken by incoming data for both SDLC modes. The reformatted data enters the 3-bit delay and is transferred to the Receive Shift register. The SDLC receive operation begins in the hunt phase by attempting to match the assembled character in the Receive Shift Register with the flag pattern in WR7. Then the flag character is recognized, subsequent data is routed through the same path, regardless of character length.

Either the CRC-16 or CRC-SDLC cyclic redundancy check (CRC) polynomial can be used for both Monosync and Bisync modes, but only the CRC-SDLC polynomial is used for SDLC operation. The data path taken for each mode is also different. Bisync protocol is a byte-oriented operation that requires the CPU to decide whether or not a data character is to be included in CRC calculation. An 8-bit delay in all synchronous modes except SDLC is allowed for this process. In SDLC mode, all bytes are included in the CRC calculation.

4.2 ASYNCHRONOUS MODE

In asynchronous communications data is transferred in the format shown in Figure 4-3.

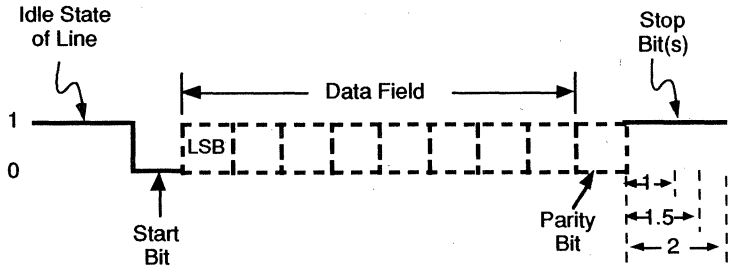


Figure 4-3. Asynchronous Message Format

The transmission of a character begins when the line makes a transition from the "1" state, or MARK condition to the "0" state or SPACE condition. This transition is the reference by which the character's bit cell boundaries are defined. Though the transmitter and receiver have no common clock signal, there must be an agreement as to the data rate so that the receiver can always sample the data in the center of the bit cell.

The character can be broken up into four fields:

Start bit - signals the beginning of a character frame.

Data field - typically 5-8 bits wide.

Parity bit - optional, provides mechanism for checking character validity, transmitter and receiver agree that:

Data + Parity bit contains odd number of 1s (odd parity) or Data + Parity bit contains even number of 1s (even parity).

Stop bit(s) - provides a minimum interval between the end of one character and the beginning of the next.

The ISCC supports Asynchronous mode with a number of programmable options including the number of bits per character, the number of stop bits, the clock factor, modem interface signals and break detect and generation.

Asynchronous mode is selected by programming the desired number of stop bits in D3 and D2 or WR4. Programming these two bits with other than "00" places both the receiver and transmitter in Asynchronous mode. In this mode, the ISCC ignores the state of bits D4, D3, and D2 of WR3, bits D5 and D4 of WR4, bits D2 and D0 of WR5, all of WR6 and WR7 and all of WR10 except D6 and D5. Bits that are ignored may be programmed with "1" or "0" or not at all. See Table 4-1 below.

Table 4-1. Write Register Bits Ignored in Asynchronous Mode

Register	D7	D6	D5	D4	D3	D2	D1	D0
WR3				x	x	x		
WR4			x	x				
WR5						x		x
WR6	x	x	x	x	x	x	x	x
WR7	x	x	x	x	x	x	x	x
WR10	x			x	x	x	x	x

4.2.1 Asynchronous Transmit

Characters are loaded from the transmit buffer to the shift register where they are given a start bit and a parity bit (if programmed), and are shifted out to the TxD pin. Each time the transmit buffer becomes empty the Tx Empty bit in RR0 is set to 1 and, optionally, an interrupt or DMA request can be generated.

The number of bits transmitted per character is controlled both by Bits D6 and D5 in WR5, and the way the data is formatted within the transmit buffer. The bits in WR5 allow the option of five, six, seven, or eight bits per character. When five bits per character is selected the data may be formatted before being written to the transmit buffer to allow transmission of from one to five bits per character.

This formatting is shown in Table 4-2.

Table 4-2. Transmit Bits per Character

Bit 7	Bit 6	
0	0	5 or less bits / character
0	1	7 bits / character
1	0	6 bits / character
1	1	8 bits / character

For five or less bits per character selection in WR5, the following encoding is used in the data sent to the transmitter. D is the data bit(s) to be sent.

D7	D6	D5	D4	D3	D2	D1	D0	
1	1	1	1	0	0	0	D	Sends one data bit
1	1	1	0	0	0	D	D	Sends two data bits
1	1	0	0	0	D	D	D	Sends three data bits
1	0	0	0	D	D	D	D	Sends four data bits
0	0	0	D	D	D	D	D	Sends five data bits

In all cases the data must be right-justified, with the unused bits being ignored except in the case of five bits or less per character.

An additional bit, carrying parity information, may be automatically appended to every transmitted character by setting bit D0 of WR4 to "1". This bit is sent in addition to the number of bits specified in WR4 or by the data format. The parity sense is selected by bit D1 of WR4. If this bit is set to "1", the transmitter sends even parity, if set to "0", the parity is odd.

The ISCC may be programmed to accept a transmit clock that is one, sixteen, thirty-two, or sixty-four times the data rate. This is selected by bits D7 and D6 of WR4, in common with the clock factor for the receiver. Note that the chosen clock factor may restrict the number of stop bits that may be transmitted. In particular, when the clock rate and data rate are identical, one-and-a-half stop bits are not allowed. If any length other than one stop bit is desired in the times one mode, only two stop bits may be used.

There are two modem control signals associated with the transmitter provided by the ISCC, namely /RTS and /CTS.

The /RTS pin is a simple output that carries the inverted state of the RTS bit (D1) in WR5, unless the Auto Enables bit (D5) is set in WR3. When Auto Enables is set, the /RTS pin will immediately go Low when the RTS bit is set. However, when the RTS bit is reset, the /RTS pin remains Low until the transmitter is completely empty and the last stop bit has left the TxD pin. Thus the /RTS pin may be used to disable external drivers for the transmit data.

The /CTS pin is ordinarily a simple input to the CTS bit in RR0. However, if Auto Enables mode is selected this pin becomes an enable for the transmitter. That is, if Auto Enables is on and the /CTS pin is High, the transmitter is disabled; the transmitter is enabled while the /CTS pin is Low.

The transmitter may be programmed to send a Break by setting bit D4 of WR5 to "1". The transmitter will send continuous "0s" from the first transmit clock edge after this command is issued, until the first transmit clock edge after this bit is reset. The transmit clock edges referred to here are those that define transmitted bit cell boundaries.

An additional status bit for use in Asynchronous mode is available in bit D0 or RR1. This bit, called All Sent, is set when the transmitter is completely empty and any previous data or stop bits have reached the TxD pin. The All Sent bit can be used by the processor as an indication that the transmitter may be safely disabled.

The initialization sequence for the transmitter in asynchronous mode is given in Table 4-3.

At this point other registers should be initialized according to the hardware design such as clocking, I/O mode, etc. When all this is completed, the transmitter may be enabled by setting WR5(3) = 1. Also note that the transmitter and receiver may be initialized at the same time.

The number of bits/char is selected by WR3, bits 6-7.

Table 4-3. Initialization Sequence for the Transmitter in Asynchronous Mode

Reg	Bit No	Description
WR4	3, 2	Select Async Mode and the number of stop bits*
	0, 1	Select parity*
	6, 7	Select clock mode*
WR3	5	Select Auto Enable Mode*
WR5	1	Select modem control (RTS)
	4	Select break generation
	6, 5	Select number of bits/char for transmitter

Note:
* initializes transmitter and receiver simultaneously

4.2.2 Asynchronous Reception

During reception, the start and stop bits are stripped away and checked for errors, leaving only the working data for CPU interaction.

The receiver always checks for one stop bit. If after character assembly the receiver finds this stop bit to be a "0", the Framing Error bit in the receive error FIFO is set at the same time that the character is transferred to the receive data FIFO. This error bit accompanies the data to the top of the FIFO, where it generates a special receive condition. The Framing Error bit is not latched, and so must be read in RR1 before the accompanying data is read.

The additional parity bit per character is transferred to the receive data FIFO along with the data if the data plus parity is eight bits or less. The Parity Error bit in the receive error FIFO may be programmed to cause a special receive condition interrupt by setting bit D2 of WR1 to "1". This error bit is latched and so will remain active, once set, until an Error Reset command has been issued. If interrupts are not used to transfer data, the Parity Error, Framing Error, and Overrun Error bits in RR1 should be checked before the data is removed from the receive data FIFO.

The ISCC may be programmed to accept a receive clock that is one, sixteen, thirty-two, or sixty-four times the data rate. This is selected by bits D7 and D6 in WR4. The 1X mode is used when bits are synchronized external to the receiver. The 1X mode is the only mode in which a data encoding method other than NRZ may be used. The clock factor is common to the receiver and transmitter.

The ISCC provides up to three modem control signals associated with the receiver.

The /SYNC pin is a general-purpose input whose state is reported in the Sync/Hunt bit in RR0. If the crystal oscillator is enabled, this pin is not available and the Sync/Hunt bit is forced to "0". Otherwise, the /SYNC pin may be used to carry the Ring Indicator signal.

The /DTR//REQ pin carries the inverted state of the DTR bit (D7) in WR5 unless this pin has been programmed to carry a DMA Request signal.

The /DCD pin is ordinarily a simple input to the DCD bit in RR0. However, if the Auto Enables mode is selected by setting D5 of WR3 to "1", this pin becomes an enable for the receiver. That is, if Auto Enables is on and the /DCD pin is High, the receiver is disabled. While the /DCD pin is Low, the receiver is enabled.

The break condition is continuous "0s", as opposed to the usual continuous ones during an idle. The ISCC recognizes the Break condition upon seeing a null character (all "0s") plus a framing error. Upon recognizing this sequence the Break bit in RR0 will be set and will remain set until a "1" is received. At this point the break condition is no longer present. At the termination of a break the receive data FIFO contains a single null character, which should be read and discarded. The Framing Error bit will not be set for this character, but if odd parity has been selected, the Parity Error bit will be set. Caution should be exercised if the receive data line contains a switch that is not debounced to generate breaks. Switch bounce may cause multiple breaks, recognized by the ISCC to be additional characters assembled in the receive data FIFO. It may also cause a receive overrun condition being latched.

Received characters are assembled, checked for errors, and moved to a three byte FIFO. When there is at least one character in the FIFO the Rx Character Available bit (in RR0) is set to 1 and, optionally, an interrupt or DMA request can be generated. Since errors apply to specific characters, it is necessary that error information moves along side the data that it refers to. This is implemented in the ISCC with a three entry error FIFO in parallel with the data FIFO. The three error conditions that the receiver checks for in asynchronous mode are:

1. Framing errors - when a character stop bit is found to be 0.

2. Parity errors - when parity is enabled and the parity of a character disagrees with the sense programmed in WR4.

3. Overrun errors - when the FIFO overflows.

The initialization sequence for the receiver in asynchronous mode is given in Table 4-4 below.

Table 4-4. Initialization Sequence for the Receiver in Asynchronous Mode

Reg	Bit No	Description
WR4	3, 2	Select Async Mode and the number of stop bits*
	0, 1	Select parity*
	6, 7	Select clock mode*
WR3	7, 6	Select number of bits / character
	5	Select Auto Enable Mode*
WR5	1	Select modem control (RTS)

Note:
* initializes transmitter and receiver simultaneously

At this point other registers should be initialized according to the hardware design such as clocking, I/O mode, etc. When all this is completed, the receiver may be enabled by setting WR3(0) = 1. Also note that the transmitter and receiver may be initialized at the same time.

4.3 BYTE - ORIENTED SYNCHRONOUS MODE

Three byte-oriented synchronous protocols supported by ISCC are monosync, bisync, and external sync.

In synchronous communications the bit cell boundaries are defined by a clock signal which is common to both the transmitter and receiver. Of course there must also be an agreement as to the location of the character boundaries so that the characters can be properly framed. This is normally accomplished by defining special SYNC patterns, or SYNC characters. The SYNC pattern serves as a reference; it signals the receiver that a character boundary occurs immediately after the last bit of the pattern. Another way of identifying the character boundaries (i.e. achieving synchronization) is with a logic signal that goes active just as the first character is about to enter the receiver. This method is referred to as "External Synchronization".

Figure 4-4 shows the character format for synchronous transmission. For example, bits 1-8 might be one character and bits 9-13 part of another character; or bit 1 might be part of a second character, and bits 10-13 part of a third character. The alignment of the received bytes to the byte assembly is accomplished by defining a synchronization character, commonly called a "sync character".

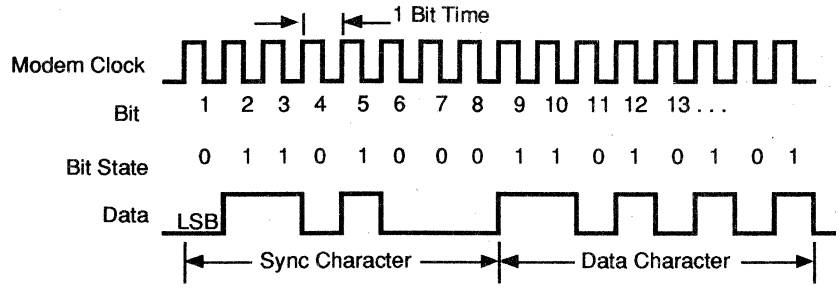


Figure 4-4. Monosync Data Character Format

Start and stop bits are not required in synchronous modes. All bits are used to transmit data. This eliminates the "waste" characteristic of asynchronous communication.

4.3.1 Byte Oriented Synchronous Transmit

Once Synchronous mode has been selected, any of three sync character lengths may be selected:

6 bit
8 bit
16 bit

The 6-bit option Sync character is selected by setting bits 4 and 5 of WR4 to zeros and bit 0 of WR10 to one. Only the least significant six bits of WR6 are transmitted.

The 8-bit sync character is selected by setting bits 4 and 5 of WR4 to zeros and bit 0 of WR10 to zero. With this option selected, the transmitter sends the contents of WR6 when it has no data to send.

Monosync and Bisync modes require clocking information to be transmitted along with the data either by a method of encoding data that contains clocking information, or by a modem that encodes or decodes clock information in the modulation process. Refer to the Monosync message format as shown in Figure 4-4.

The Bisync mode of operation is similar to the Monosync mode, except that two sync characters are provided instead of one. Bisync attempts a more structured approach to synchronization through the use of special characters as message "headers" or "trailers".

External Sync mode eliminates the use of sync characters in the serial data stream by providing an external sync signal to mark the beginning of a data field; i.e., an external input pin (Sync) waits for an active state change to indicate the beginning of an information field.

Character-oriented mode is selected by programming bits D3 and D2 of WR4 with zeros. This selects synchronous mode, as opposed to asynchronous mode, but this selection is further modified by bits 5 to 7 of WR4 as well as bits 1 and 0 of WR10. In sync character-oriented modes, except External Sync mode, the state of bits 7 and 6 of WR4 are always forced internally to zeros. In external sync mode, these two bits must be programmed as described in Section 5.4.5.

Table 4-5. Registers Used in Character-oriented Modes

Register	Bit No	Description
WR4	3 (=0)	Select sync mode
	2 (=0)	
	4 (=0)	Select monosync mode
	5 (=0)	(8 bit sync character)
	4 (=1)	Select bisync mode
	5 (=0)	(16 bit sync character)
	4 (=1)	Select external sync mode
	5 (=1)	(external sync signal required)
	6 (=0)	Select 1x clock mode
	7 (=0)	
WR6	7-0	Sync character (low byte)
WR7	7-0	Sync character (high byte)
WR10	1	Select sync character length

In character-oriented modes, a special bit pattern is used to provide character synchronization. The ISCC offers several options to support synchronous mode including various sync generation and checking, CRC generation and checking, as well as modem controls and a transmitter to receiver synchronization function.

For a 16-bit sync character, set bit D4 of WR4 to "1" and bit D5 of WR4 and bit D0 of WR10 to "0". In this mode the transmitter sends the concatenation of WR6 and WR7 as a time fill.

Because the receiver requires that sync characters be left-justified in the registers, while the transmiter requires them to be right justified, only the receiver will work with a 12-bit sync character. While the receiver is in External Sync mode, the transmitter sync length may be six or eight bits, as selected by bit D0 of WR10.

The number of bits per transmitted character is controlled by D6 and D5 of WR5 and the way the data is formatted within the transmit buffer. The bits in WR5 allow the option of five, six, seven, or eight bits per character. When five bits per character is selected the data may be formatted before being written to the transmit buffer to allow transmission of from one to five bits per character. This formatting is shown in Table 4-2. In all cases the data must be right-justified, with the unused bits being ignored except in the case of five bits per character.

An additional bit, carrying parity information, may be automatically appended to every transmitted character by setting bit D0 of WR4 to "1". This parity bit is sent in addition to the number of bits specified in WR4 or by the data format. If this bit is set to "1", the transmitter will send even parity, if set to "0", the transmitted parity will be odd.

Either of two CRC polynomials may be used in synchronous modes, selected by bit D2 in WR5. If this bit is set to "1", the CRC-16 polynomial is used and, if this bit is set to "0", the CRC-CCITT polynomial is used. This bit controls the selection for both the transmitter and receiver. The initial state of the generator and checker is controlled by bit D7 of WR10. When this bit is set to "1", both the generator and checker will have an initial value of all ones, if this bit is set to "0", the initial values will be all zeros.

The ISCC does not automatically preset the CRC generator, so this must be done in software. This is accomplished by issuing the Reset Tx CRC Generator command, which is encoded in bits D7 and D6 of WR0. For proper results this command must be issued while the transmitter is enabled and sending sync characters.

If CRC is to be used, the transmit CRC generator must be enabled by setting bit D0 of WR5 to "1". This bit may also be used to exclude certain characters from the CRC calculation. Sync characters are automatically excluded from the CRC calculation and any characters written as data may also be excluded from the calculation by using bit D0 of WR5. Internally, the CRC is enabled or disabled for a particular character at the same time as the character is loaded from the transmit buffer to the Transmit Shift register. Thus, to exclude a character from CRC calculation bit D0 of WR5 should be set to "0" before the character is written to the transmit buffer. This guarantees that the internal disable will occur when the character moves from the buffer to the shift register. Once the buffer becomes empty, the Tx CRC Enable bit may be written for the next character.

Enabling the CRC generator is not sufficient to control the transmission of CRC. In the ISCC this function is controlled by the Tx Underrun/EOM bit, which may be reset by the processor and set by the ISCC. When the transmitter underruns (both the transmit buffer and Transmit Shift register are empty) the state of the Tx Underrun/EOM bit determines the action taken by the ISCC. If the Tx Underrun/EOM bit is not set when the underrun occurs, the transmitter will send the accumulated CRC and set the Tx Underrun/EOM bit to indicate this. This transition may be programmed to cause an external/status interrupt, or the Tx Underrun/EOM is available in RR0.

The Reset Tx Underrun/EOM Latch command is encoded in bits D7 and D6 of WR0. For correct transmission of the CRC at the end of a block of data, this command must be issued after the first character is written to the ISCC but before the transmitter underruns after the last character written to the ISCC. The command is usually issued immediately after the first character is written to the ISCC so that CRC will be sent if an underrun occurs inadvertently during the block of data.

If the transmitter is disabled during transmission of a character, that character will be sent completely. This applies to both data and sync characters. However, if the transmitter is disabled during the transmission of CRC, the 16-bit transmission will be completed, but the remaining bits will come from the SYNC registers rather than the remainder of the CRC.

There are two modem control signals associated with the transmitter provided by the ISCC: /RTS and /CTS.

The /RTS pin is a simple output that carries the inverted state of the RTS bit (D1) in WR5.

The /CTS pin is ordinarily a simple input to the CTS bit in RR0. However, if Auto Enables mode is selected this pin becomes an enable for the transmitter. That is, if Auto Enables is ON and the /CTS pin is High the transmitter is disabled. While the /CTS pin is Low, transmitter is enabled.

The initialization sequence for the transmitter in character-oriented mode is shown in Table 4-6.

Table 4-6. Transmitter Initialization in Character Oriented Mode

Register	Bit No	Description
WR4	0,1	select parity
WR5	1	RTS
	2	select CRC generator
	5,6	select number of bits per character
WR10	7	CRC preset value

At this point, the other registers should be initialized as necessary. When all of this is completed the transmitter maybe enabled by setting bit 3 of WR5 to one. Now that the transmitter is enabled the CRC generator maybe initialized by issuing the Reset Tx CRC Generator command in WR0, bit 6-7.

4.3.2 Byte-Oriented Synchronous Receive

The CPU places the receiver in Hunt mode whenever transmission begins (or whenever a data dropout has occurred and the hardware determines that resynchronization is necessary). In Hunt mode, the receiver shifts a bit into the Receive Shift register and compares the contents of the Receive Shift register and with the sync character (stored in another register), repeating the process until a match occurs. When a match occurs, the receiver begins transferring bytes to the receive FIFO.

Once the sync character-oriented mode has been selected, any of the four sync character length maybe selected: 6-bits, 8-bits, 12-bits, or 16-bits.

The Table 4-7 shows the WR register bit setting for selecting sync character length.

Table 4-7. Sync Character Length Selection

Sync Length	WR4,D5	WR4,D4	WR10,D0
6 bits	0	0	1
8 bits	0	0	0
12 bits	0	1	1
16 bits	0	1	0

The arrangement of the sync character in WR6 and WR7 is shown in Figure 4-5.

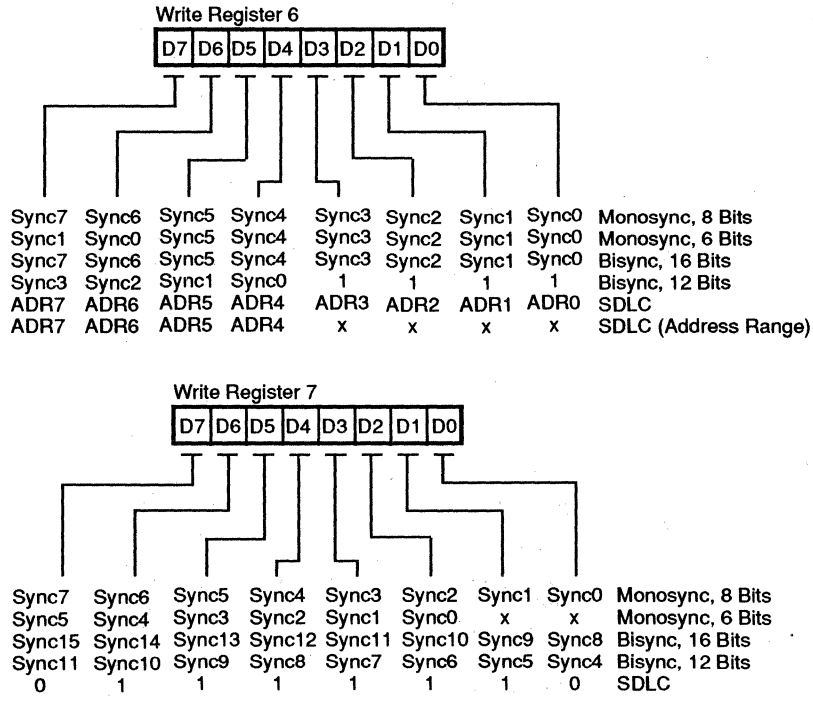


Figure 4-5. Sync Character Programming

For those applications requiring any other sync character length, the ISCC makes provision for an external circuit to provide a character synchronization signal on the /SYNC pin. This mode is selected by setting bits D5 and D4 of WR4 to "1". In this mode the Sync/Hunt bit in RR0 reports the state of the /SYNC pin but the receiver must still be placed in Hunt mode when the external logic is searching for a sync character match. When the receiver is in Hunt mode and the /SYNC pin is driven Low, two receive clock cycles after the last bit of the sync character is received, character assembly will begin on the rising edge of the receive clock immediately preceding the activation of /SYNC. This is shown in Figure 4-6. The receiver leaves Hunt mode when /SYNC is driven Low.

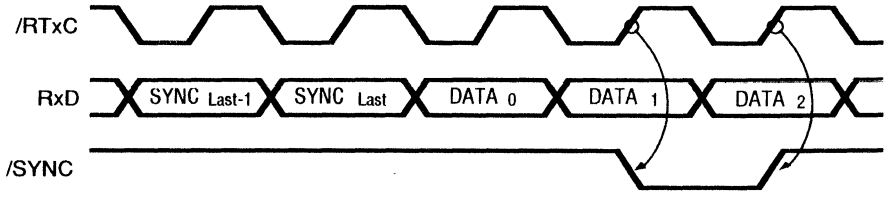


Figure 4-6. /SYNC as an Input

In all cases except External Sync mode the /SYNC pin is an output that is driven Low by the ISCC to signal that a sync character has been received. The /SYNC pin is activated regardless of character boundaries so any external circuitry using it should only respond the /SYNC pulse that occurs while the receiver is in Hunt mode. The timing for the /SYNC signal is shown in Figure 4-7.

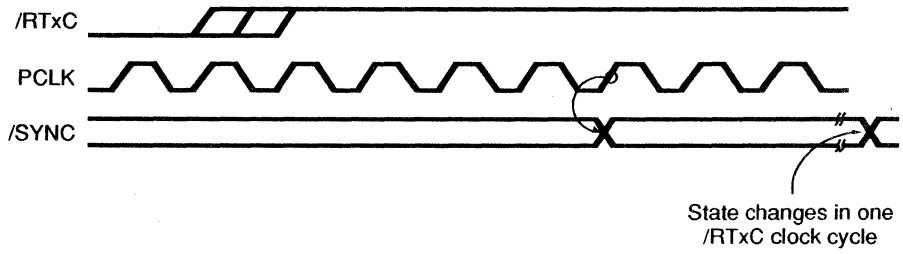


Figure 4-7. /SYNC as an Output

It is sometimes desirable to prevent sync characters from entering the receive data FIFO. This function is available in the ISCC by setting the Sync Character Load inhibit bit (D1) in WR3 to "1". While this bit is set to "1", the character about to be loaded into the receive data FIFO is compared with the contents of WR6. If all eight bits match the character, it is not loaded into the receive data FIFO. Because the comparison is across eight bits, this function works correctly only when the number of bits per character is the same as the sync character length. Thus it cannot be used with 12- or 16-bit sync characters. Both leading sync characters and sync characters embedded in the data may be properly removed in the case of a 8-bit sync character. Care must be exercised in using this feature because sync characters not transferred to the receive data FIFO will automatically be excluded from CRC calculation. This works properly only in the 8-bit case.

The receiver in the ISCC searches for character synchronization only while it is in Hunt mode. In this mode the receiver is idle having been first enabled, and may be placed in Hunt mode by command from the processor. this is accomplished by issuing the Enter Hunt Mode command in WR3. This bit (D4) is a command; writing a "0" to it has no effect. The Hunt status of the receiver is reported by the Sync/Hunt is one of the possible sources of external/status interrupts, with both transitions causing an interrupt. This is true even if the Sync/Hunt bit is set as a result of the processor issuing the Enter Hunt Mode command.

The number of bits per character is controlled by bits D7 and D6 of WR3. Five, six, seven, or eight bits per character may be selected via these two bits. The data is right-justified in the receive data buffer. The ISCC merely takes a snapshot of the receive data stream at the appropriate times so the "unused" bits in the receive buffer are only the bits following the character in the data stream.

An additional bit, carrying parity information, may be selected by setting bit D0 of WR4 to "1". If this bit is set to "1", the received character is checked for even parity, if set to "0", the received character is checked for odd parity. The additional bit per character is not visible when there are eight data bits per character. The Parity Error bit in the receive error FIFO may be programmed to cause a Special Receive Condition interrupt by setting bit D2 of WR1 to "1". This error bit is latched and so will remain active, once set, until an Error Reset command has been issued. If interrupts are not used to transfer data the Parity Error, CRC Error, and Overrun Error bits in RR1 should be checked before the data is removed from the receive data FIFO.

The character length may be changed at any time before the new number of bits has been assembled by the receiver, but, care should be exercised as unexpected results may occur. A representative example, switching from five bits to eight bits and back to five bits is shown in Figure 4-8.

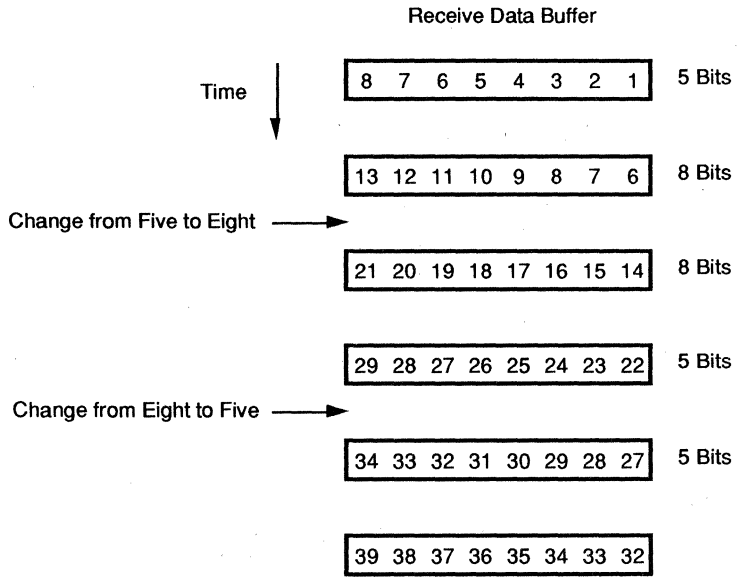


Figure 4-8. Changing Character Length

Either of two CRC polynomials may be used in synchronous modes, selected by bit D2 in WR5. If this bit is set to "1", the CRC-16 polynomial is used, if this bit is set to "0", the CRC-CCITT polynomial is used. This bit controls the polynomial selection for both the receiver and transmitter.

The initial state of the generator and checker is controlled by bit D7 of WR10. When this bit is set to "1", both the generator and checker will have an initial value of all ones, if this bit is set to "0", the initial values will be all "0s". The ISCC presets the checker whenever the receiver is in Hunt mode so a CRC reset command is not strictly necessary. However, there is a Reset CRC Checker command in WR0. This command is encoded in bits D7 and D6 of WR0. If CRC is to be used the CRC checker must be enabled by setting bit D0 of WR3 to "1".

If sync characters are being stripped from the data stream, this may be done at any time before the first non-sync character is received. If the sync strip feature is not being used, CRC must not be enabled until after the first data character has been transferred to the receive data FIFO. As previously mentioned, 8-bit sync characters stripped from the data stream are automatically excluded from CRC calculation.

Some synchronous protocols require that certain characters be excluded from CRC calculation. This is possible in the ISCC because CRC calculation may be enabled and disabled on the fly. To give the processor sufficient time to decide whether or not a particular character should be included in the CRC calculation, the ISCC contains an 8-bit time delay between the receive shift register and the CRC checker. The logic also guarantees that the calculation will only start or stop on a character boundary by delaying the enable or disable until the next character is loaded into the receive data FIFO.

To understand how this works refer to Figure 4-9 and the following explanation. Consider a case where the ISCC receives a sequence of eight bytes, called A, B, C, D, E, F, G and H with A received first. Now suppose that A is the sync character, that CRC is to be calculated on B, C, E, and F, and that F is the last byte of this message. A process is used to control the ISCC as described below.

The Receive Character-Operational Stages:

1. Before A is received the receiver is in Hunt mode and the CRC is disabled. When A is in the receive shift register it is compared with the contents of WR7. Since A is the sync character, the bit patterns match and receive leaves Hunt mode, but character A is not transferred to the receive data FIFO.

2. After 8-bit times, B is loaded into the receive data FIFO. The CRC remains disabled even though somewhere during the next eight bit times the processor reads B and enables CRC. At the end of this eight-bit time, B is in the 8-bit delay and C is in the receive shift register.

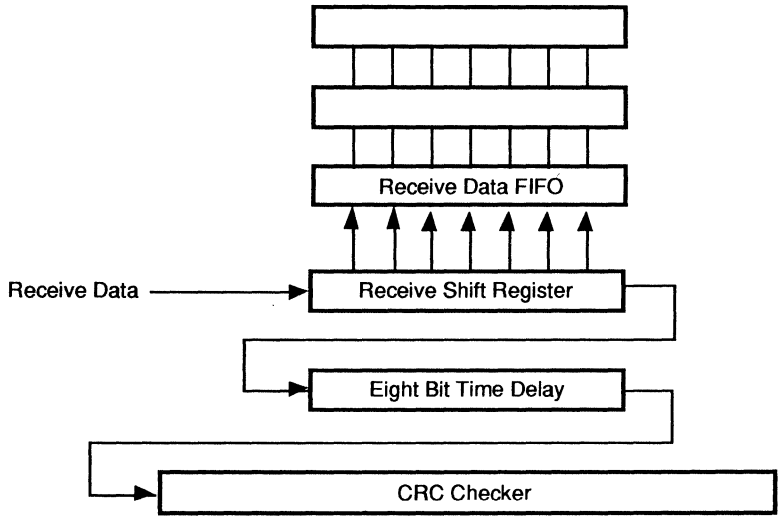


Figure 4-9. Receive CRC Data Path

3. Character C is loaded into the receive data FIFO and at the same time the CRC checker becomes enabled. During the next eight-bit-time, the processor reads C and since CRC is enabled within this period, the ISCC has calculated CRC on character B; character C is in the 8-bit delay and D is in the Receive Shift register. D is then loaded into the receive data FIFO and at some point during the next eight-bit-time the processor reads D and disables CRC. At the end of these eight-bit-times CRC has been calculated on C, character D is in the 8-bit delay and E is in the Receive Shift register.

4. Now E is loaded into the receive data FIFO. During the next eight-bit-times the processor reads E and enables the CRC. During this time E shifts into the 8-bit delay, F enters the Receive Shift register and CRC is not being calculated on D. After these eight-bit-times have elapsed, E is in the 8-bit delay, and F is in the Receive Shift register. Now F is transferred to the receive data FIFO and CRC is enabled. During the next eight-bit-times the processor reads F and leaves the CRC enabled. The processor is usually aware that this is the last character in the message and so prepares to check the result of the CRC computation. However, another sixteen bit-times are required before CRC has been calculated on all of character F.

5. At the end of eight-bit-times F is in the 8-bit delay and G is in the Receive Shift register. At this time G is transferred to the receive data FIFO. Character G must be read and discarded by the processor. Eight bit times later H is transferred to the receive data FIFO also. The result of a CRC calculation is latched in the receive error FIFO at the same time as data is written to the receive data FIFO. Thus the CRC result through character F accompanies character H in the FIFO and will be valid in RR1 until character H is read from the receive data FIFO. The CRC checker may be disabled and reset at any time after character H is transferred to the receive data FIFO. Recall, however, that internally CRC will not be disabled until after this occurs. A better alternative is to place the receiver in Hunt mode, which automatically disables and resets the CRC checker. See Table 4-8 for a condensed description.

Table 4-8. Enabling and Disabling CRC on the fly

A	B	C	D	E	F	G	H
(Sync)	(Data1)	(Data2)	(Data3)	(CRC1)	(CRC2)	(Data)	(Data)

Note: No CRC Calculation on "D"

Stage	Direction of Data Coming into SCC →	Shift Register	Receive Data FIFO	Delay Register	CRC	Notes
1	H G F E D C B A				d	
	H G F E D C B	A			d	
2	H G F E D C CPU Reads CPU Enables CRC	B	B		d	
3	H G F E D CPU Reads	C	C	B	e	
	H G F E CPU Reads CPU Disables CRC	D	D	C	e	→ CRC Calc on B → CRC Calc on C
4	H G F CPU Reads CPU Enables CRC	E	E	D	d	
	H G CPU Reads	F	F	E	e	→ CRC Calc is Disabled on D → CRC Calc on E
5	H CPU Reads & Discard	G	G	F	e	→ CRC Calc on F
		H		G *	e	→ CRC Calc on F * Result latched in Error FIFO †
	Read RR1 D6 Read H & Discard		H H			

Legend:

* Usually G is a end-of-message character indicator.

† The status is latched on the Error FIFO for each received byte. In the calculation of F, the CRC error flag in the Error FIFO will be 0 for an error free message.

d = disabled

e = enabled

A B C D E F G H
A = SYNC
B - F = Data with E = CRC1 and F = CRC2
G and H are arbitrary data

Up to two modem control signals associated with the receiver are available in synchronous modes: DTR/REQ and DCD. The /DTR//REQ pin carries the inverted state of the DTR bit (D7) in WR5 unless this pin has been programmed to carry a DMA Request signal. The /DCD pin is ordinarily a simple input to the DCD bit in RR0. However, if the Auto Enables mode is selected by setting D5 of WR3 to "1", this pin becomes an enable for the receiver. Then if Auto Enables is ON and the /DCD pin is High the receiver is disabled; while the /DCD pin is Low the receiver is enabled.

The initialization sequence for the receiver in character-oriented mode is WR4 first, to select the mode, then WR10 to modify it if necessary, WR6 and WR7 to program the sync characters and then WR3 and WR5 to select the various options. At this point the other registers should be initialized as necessary. When all this is completed the receiver is enabled by setting bit 0 of WR3 to a one. A summary is shown in Table 4-9.

Table 4-9. Initializing the Receiver in Character Oriented Mode

Register	Bit No	Description
WR4	4-5	Select sync character
WR10	0	Length
WR4	4-5	Select external sync
WR6	0-7	Sync character, lower byte
WR7	0-7	Sync character, upper byte
WR3	1	Sync character inhibit
	4	Enter hunt mode
WR3	6-7	Number of bits / character
WR4	0-1	Select parity
WR5	2	Select CRC
WR10	7	CRC generator initial state
WR0	7-6	Reset CRC generator
WR3	0	CRC enable
WR5	7	DTR/REQ
WR3	5	Auto enable

4.3.3 Transmitter/Receiver Synchronization

The ISCC contains a transmitter-to-receiver synchronization function that may be used to guarantee that the character boundaries for the received and transmitted data are the same. In this mode the receiver is in Hunt and the transmitter is idle, sending either all "1s" or all "0s". When the receiver recognizes a sync character, it leaves Hunt mode and one character time later the transmitter is enabled and begins sending sync characters. Beyond this point the receiver and transmitter are again completely independent, except that the character boundaries are now aligned. This is shown in Figure 4-10.

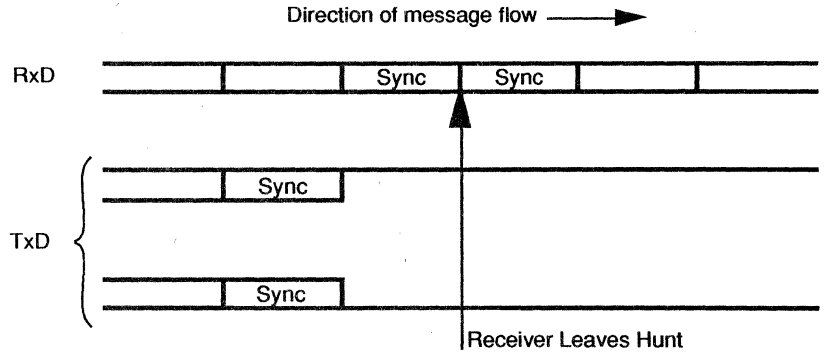


Figure 4-10. Transmitter to Receiver Synchronization

There are several restrictions on the use of this feature in the ISCC. First, it will only work with 6-bit, 8-bit or 16-bit sync characters, and the data character or eight bits with an 8-bit or 16-bit sync character. Of course, the receive and transmit clocks must have the same rate as well as the proper phase relationship.

A specific sequence of operations must be followed to synchronize the transmitter to the receiver. Both the receiver and transmitter must have been initialized for operation in Synchronous mode sometime in the past, although this initialization need not be redone each time the transmitter is synchronized to the receiver. The transmitter is disabled by setting bit D3 of WR5 to "0". At this point the transmitter will send continuous "1s". If it is desired that continuous "0s" be transmitted, the Send Break bit (D4) in WR5 should be set to "1". The transmitter is now idling but must still be placed in the transmitter to receiver synchronization mode. This is accomplished by setting the Loop Mode bit (D1) in WR10 and then enabling the transmitter by setting bit D3 to WR5 to "1". At this point the processor should set the Go Active on Poll bit (D4) in WR10. The final step is to force the receiver to search for sync characters. If the receiver is currently disabled the receiver will enter Hunt mode when it is enabled by setting bit D0 of WR3 to "1". If the receiver is already enabled it may be placed in Hunt mode by setting bit D4 of WR3 to "1". Once the receiver leaves Hunt mode the transmitter is activated on the following character boundary.

4.4 BIT-ORIENTED SYNCHRONOUS MODE

Synchronous Data Link Control mode (SDLC) uses synchronization characters similar to Bisync and Monosync modes (such as flags and pad characters), but it is a bit-oriented protocol instead of byte-oriented protocol. High-Level synchronous Data Link Communication (HDLC) protocol is identical to SDLC except for differences in framing and can be handled by the ISCC using the SDLC mode. The discussions on SDLC which follow are equally applicable to HDLC.

Any data communication link involves at least two stations. The station that is responsible for the data link and issues the commands to control the link is called the "primary station". The other station is a "secondary station". Not all information transfers need to be initiated by a primary station. In SDLC mode, a secondary station can be the initiator.

The basic format for SDLC is a "frame" (Figure 4-11). The information field is not restricted in format or content and can be of any reasonable length (including zero). Its maximum length is that which can be expected to arrive at the receiver error-free most of the time. Hence, the determination of maximum length is a function of communication channel error rate.

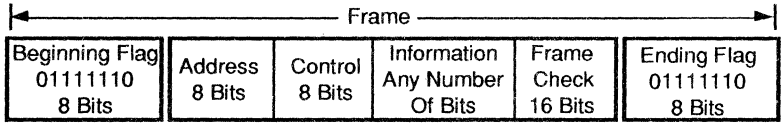


Figure 4-11. SDLC Message Format

Two flags that delineate the SDLC frame serve as reference points when positioning the address and control fields, and they initiate the transmission error check. The ending flag indicates to the receiving station that the 16-bits just received constitute the frame check. The ending flag could be followed by another frame, another flag, or an idle. This means that when two frames follow one another, the intervening flag may simultaneously be the ending flag of the first frame and the beginning flag of the next frame. Since the SDLC mode does not use characters of defined length, but rather works on a bit-by-bit basis, the 01111110 (7EH) flag can be recognized at any time.

To ensure that the flag is not sent accidentally, SDLC procedures require a binary "0" to be inserted by the transmitter after the transmission of any five contiguous "1s". The receiver then removes the "0" following a received succession of five "1s". Inserted and removed "0s" are not included in the CRC calculation.

There are two unique bit patterns in SDLC mode besides the flag sequence. They are the Abort and EOP (End of Poll) sequence. An Abort is a sequence of from seven to thirteen consecutive "1s" and is used to signal the premature termination of a frame. The EOP is the bit pattern

"11111110", which is used in loop applications as a signal to a secondary station that it may begin transmission.

The address field can consist of one or more octets and is used to designate the number of secondary station to which the commands or data are sent. A control field may follow the address. The control field is eight bits long and is used to initiate SDLC activities. Data follows the control field any may consist of any number of bits.

In the SDLC mode, the ISCC operates in the following way. In SDLC mode, frames of information are opened and closed by a flag. The Flag character has the unique bit pattern of "01111110". When transmitting data or CRC, the transmitter automatically performs zero insertion after five consecutive ones, irrespective of character boundaries. In turn, the receiver searches the receive data stream for five consecutive "1s" and deletes the next bit if it is a "0".

CRC may be used in SDLC mode but only with the CRC-CCITT polynomial. In the SDLC Mode, the transmitter in the SCC cell automatically inverts the CRC before transmission Because of this inversion, the receiver CRC check results in a non-zero, but fixed remainder for errorless data. The fixed remainder for this mode is "0001110100001111" and this is the pattern automatically checked for in the receiver in this mode. This is consistent with bit-oriented protocols such as SDLC, HDLC, and ADCCP.

SDLC mode is selected by setting bit D5 of WR4 to "1" and bits D4, D3, and D2 of WR4 to "0". In addition, the flag sequence must be written to WR7. Additional control bits for SDLC mode are located in WR10.

4.4.1 SDLC Transmit

In SDLC mode the transmitter moves characters from the transmit buffer to the shift register, through the zero inserter, and out the TxD pin. The transmitter does not automatically send the address byte; it merely encapsulates the data supplied by the processor with flags and CRC. Also, the processor must load the flag into WR7 as the ISCC does not have a default flag pattern.

Ordinarily, a frame will be terminated by the ISCC with CRC and a flag but the ISCC may be programmed to send an abort and a flag in place of the CRC. This option allows the ISCC to abort a frame transmission in progress if the transmitter is accidentally allowed to underrun. This is controlled by the Abort/Flag on Underrun bit (D2) in WR10. When this bit is set to "1" the transmitter will send an abort and a flag in place of the CRC when an underrun occurs. The frame will be terminated normally, with CRC and a flag, if this bit is set to "0", and the Tx Underrun /EOM latch is reset.

The ISCC is also able to send an abort by command of the processor. The Send Abort command, issued in WR0, will send eight consecutive "1s" and then the transmitter will idle. The Send Abort command also empties the transmit buffer register. Since up to five consecutive "1s" may have been sent prior to the Send Abort command being issued, the command will cause a sequence of from eight to thirteen "1s" to be transmitted (five ones of data followed by eight ones of the abort).

After the abort when the transmitter enters the idle condition, the ISCC permits sending continuous 1's instead of idle flags. This option is envoked by setting the Mark/Flag idle bit (D3) in WR10 to "1". Note that the closing flag will be transmitted correctly even if this mode is selected.

Before a new frame is transmitted, the Mark/Flag idle bit must be set to "0" to allow an opening flag to be transmitted. The Mark/Flag Idle bit must be set to "0" before data is written to the transmit buffer. Care must be exercised in doing this because the continuous "1s" are transmitted, eight at a time (as bytes) by the transmit shift register. After setting the Mark/Flag Idle bit to "0", the software must allow time for eight continuous ones to have left the Transmit Shift register before the first data byte is written to the transmit buffer. This allows the transmitter to recognize that the Flag Idle option has been invoked then, seeing an empty transmit buffer, the transmitter will load the flag into the shift register for transmission. Once the flag load has been done, the data may be placed in the transmit buffer without disturbing the transmission of the flag. (Note that when using the transmitter in SDLC mode, all data passes through the zero inserter, which adds an extra five bit times of delay between the Transmit Shift register and the Transmit Data pin.)

The number of bits per transmitted character is controlled by bits D6 and D5 of WR5 and the way the data is formatted within the transmit buffer. The bits in WR5 allow the option of five, six, seven, or eight bits per character. When "five bits per character" is selected, the data must be specially formatted before being written to the transmit buffer. This formatting is shown in Table 4-2. In all cases the data must be right-justified, with the unused bits being programmed as per the table (three zeros to the left of the data followed by 1's to the left of the zeros to complete the byte).

An additional bit, carrying parity information, may be automatically appended to every transmitted character by

setting bit D6 of WR4 to "1". This bit is sent in addition to the number of bits specified in WR4 or by the data format. The parity sense is selected by bit D1 of WR4. Parity is not normally used in SDLC mode.

The character length may be changed on the fly, but the desired length must be selected before the character is loaded into the transmit shift register from the transmit buffer. The easiest way to ensure this is to write to WR5 to change the character length before writing the data to the transmit buffer.

Only the CRC-CCITT polynomial may be used in SDLC mode. This is selected by setting bit D2 in WR5 to "0". This bit controls the selection for both the transmitter and receiver. The initial state of the generator and checker is controlled by bit D7 of WR10. When this bit is set to "1", both the generator, and checker will have an initial value of all "1s" and, if this bit is set to "0", the initial values will be all "0s".

The ISCC does not automatically preset the CRC generator so this must be done in software. This is accomplished by issuing the Reset Tx CRC generator command, which is encoded in bits D7 and D6 of WR0. For proper results, this command must be issued while the transmitter is enabled and idling. If CRC is to be used the transmit CRC generator must be enabled by setting bit D0 of WR5 to "1". CRC is normally calculated on all characters between opening and closing flags, so this bit is usually set to "1" at initialization and never changed.

Enabling the CRC generator is not sufficient to control the transmission of CRC. In the ISCC this function is controlled by the Tx Underrun/EOM bit, which may be reset by the processor and set by the ISCC.

When the transmitter underruns (both the transmit buffer and transmit shift register are empty) the state of the Tx Underrun EOM bit determines the action taken by the ISCC.

If the Tx Underrun/EOM bit is set to "1" when the underrun occurs, the transmitter will send flags.

The Reset Tx Underrun/EOM Latch command is encoded in bits D7 and D6 of WR0.

If this bit is reset to "0" when the underrun occurs, the transmitter will send either the accumulated CRC followed by flags, or an abort followed by flags, depending on the state of the Abort/Flag on Underrun bit in the WR10, Bit 1. A summary is shown in Table 4-10.

Table 4-10. Underrun EOM Bit

Tx Underrun /EOM Latch Bit	Abort/Flag	Action taken by ISCC upon transmit underrun
0	0	Sends CRC followed by flag
0	1	Sends abort followed by flag
1	x	Sends flag

The ISCC sets the Tx Underrun/EOM Latch when the CRC or abort is loaded into the shift register for transmission. This event can cause an interrupt, and the status of the Tx Underrun Latch can be read in RR0. The Tx Underrun Latch may be reset by the processor via WR0.

For correct transmission of the CRC at the end of a frame, the Reset Tx Underrun / EOM Latch command must be issued after the first character is written to the ISCC but before the transmitter underruns after the last character written to the ISCC. The command is usually issued immediately after the first character is written to the ISCC so that the abort or CRC is sent if an underrun occurs inadvertently. The Abort/Flag on Underrun bit (D2) in WR10 is usually set to "1" at the same time as the Tx Underrun/EOM bit is reset so that an abort can still be sent if the transmitter underruns. The Abort/Flag on Underrun bit is then set to "0" near the end of the frame to allow the correct transmission of CRC.

In this paragraph the term "completely sent" means shifted out of the Transmit Shift register, not shifted out of the zero inserter, which is an additional five bit times of delay. In SDLC mode, if the transmitter is disabled during transmission of a character, that character will be "completely sent". This applies to both data and flags. However, if the transmitter is disabled during the transmission of CRC, 16 total bits corresponding to the two CRC bytes will be transmitted but part of the bits will be from the CRC generator and the latter part of the bits will be from the Flag register rather than form the CRC generator. Thus part of the CRC bytes will not be transmitted.

There are two modem control signals associated with the transmitter provided by the ISCC.

The /RTS pin is a simple output that carries the inverted state of the RTS bit (D1) in WR5.

The /CTS pin is ordinarily a simple input to the CTS bit in RR0. However, if Auto Enables mode is selected this pin becomes and enable for the transmitter. That is, if Auto Enables is ON and the /CTS pin is High the transmitter is disabled. If the /CTS pin is Low, the transmitter is enabled.

The initialization sequence for the transmitter in SDLC mode is: WR4 first, to select the mode, then WR10 to modify it if necessary, WR7 to program the flag, and then WR3 and WR5 to select the various options. At this point the other registers should be initialized as necessary. When all of this is complete, the transmitter may be enabled by setting bit D3 of WR5 to "1". Now that the transmitter is enabled, the CRC generator may be initialized by issuing the Reset Tx CRC Generator command in WR0. A summary is shown in Table 4-11.

Table 4-11. Initializing the Transmitter in SDLC Mode

Register	Bit No	Description
WR5	5-6	Number of bits per character
WR4	1-0	Select parity
WR5	2	Select CRC-CCITT
WR10	7	Select CRC preset value
WR0	6-7	Reset Tx CRC
WR10	1	Abort / flag on underrun
WR0	6-7	Tx underrun
WR7	6-7	Flag

4.4.2 SDLC Receive

The receiver in the ISCC always searches the receive data stream for flag characters in SDLC mode. Ordinarily, the receiver transfers all received data between flags to the receive data FIFO. However, if the receiver is in Hunt mode no flag is received. The receiver is in Hunt mode when first enabled, or the receiver may be placed in Hunt mode by the processor issuing the Enter Hunt mode command in WR3. this bit (D4) is a command, and writing a "0" to it has no effect. The Hunt status of the receiver is reported by the Sync/Hunt bit in RR0.

Sync/Hunt is one of the possible sources of external/status interrupts, with both transitions causing an interrupt. This is true even if the Sync/Hunt bit is set as a result of the processor issuing the Enter Hunt mode command.

The receiver will automatically enter Hunt mode if an abort is received. Because the receiver always searches the receive data stream for flags and automatically enters Hunt Mode when an abort is received, the receiver will always handle frames correctly, and the Enter Hunt Mode command should never be needed. The ISCC will drive the SYNC pin Low to signal that a flag has been recognized. the timing for the SYNC signal is shown in Figure 4-12.

The first byte in an SDLC frame is assumed by the ISCC to be the address of the secondary station for which the frame is intended. The ISCC provides several options for handling this address.

If the Address Search Mode bit (D2) in WR3 is set to "0" the address recognition logic is disabled and all received frames are transferred to the receive data FIFO. In this mode the software must perform any address recognition.

If the Address Search Mode bit is set to "1", only those frames whose address matches the address programmed in WR6 or the global address (all "1s") will be transferred to the receive data FIFO.

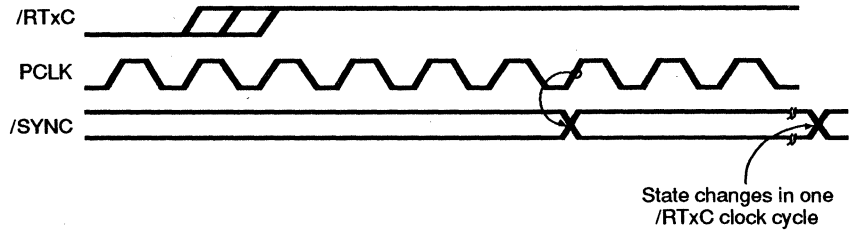


Figure 4-12. /SYNC as an Output

The address comparison will be across all eight bits of WR6 if the Sync Character Load inhibit bit (D1) in WR3 is set to "0". The comparison may be modified so that only the four most significant bits of WR6 must match the received address. This mode is selected by setting the Sync Character Load inhibit bit to "1". In this mode, however, the address field is still eight bits wide. The address field is transferred to the receive data FIFO in the same manner as data. It is not treated differently than data.

The number of bits per character is controlled by bits D7 and D6 of WR3. Five, six, seven, or eight bits per character may be selected via these two bits. The data is right-justified in the receive buffer. The ISCC merely takes a snapshot of the receive data stream at the appropriate times, so the "unused" receive buffer are only the bits following the character.

An additional bit carrying parity information may be selected by setting bit D6 of WR4 to "1". This also enables parity in the transmitter. The parity sense is selected by bit D1 of WR4. Parity is not normally used in SDLC mode. The character length may be changed at any time before the new number of bits have been assembled by the receiver. Care should be exercised, however, as unexpected results may occur. A representative example, switching from five bits to eight bits and back to five bits is shown in Figure 4-13.

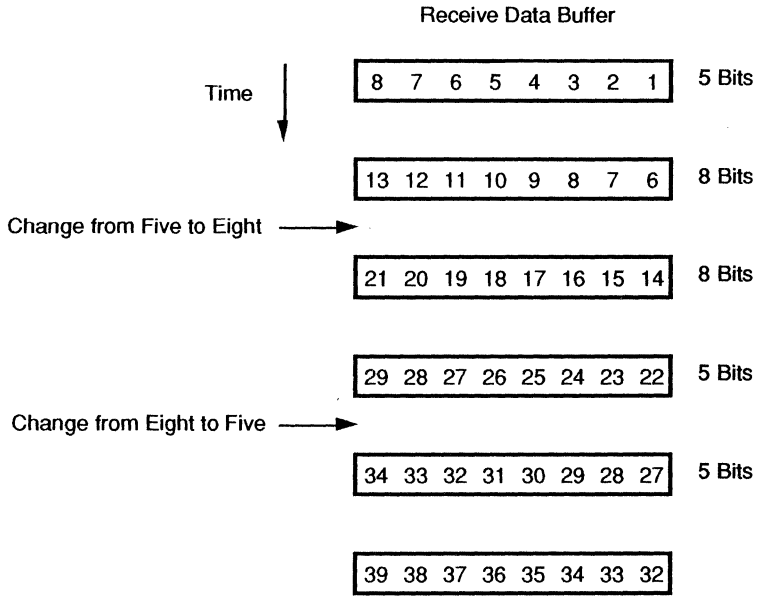


Figure 4-13. Changing Character Length

Most bit-oriented protocols allow an arbitrary number of bits between opening and closing flags. The ISCC allows for this by providing three bits of Residue Code in RR1 that indicates which bits in the last three bytes transferred from the receive data FIFO by the processor are actually valid data bits (and not part of the frame check sequence or CRC). Table 4-12 gives the meanings of the different codes for the four different character length options. The valid data bits are right-justified, that is to say if the number of valid bits given by the table is less than the character length, then the bits that are valid are the right-most or least significant bits. It should also be noted that the Residue Code is only valid at the time when the End of Frame bit in RR1 is set to 1.

Table 4-12. Residue Codes

Residue Code 2 1 0	Bits in Previous Byte 8B/C	7B/C	6B/C	5B/C	Bits in Second Previous Byte 8B/C	7B/C	6B/C	5B/C	Bits in Third Previous Byte 8B/C	7B/C	6B/C	5B/C
100	0	0	0	0	3	1	0	0	8	7	5	2
010	0	0	0	0	4	2	0	0	8	7	6	3
110	0	0	0	0	5	3	1	0	8	7	6	4
001	0	0	0	0	6	4	2	0	8	7	6	5
101	0	0	0	0	7	5	3	1	8	7	6	5
011	0	0	0		8	6	4		8	7	6	
111	1	0			8	7			8	7		
000	2				8				8			

As indicated in the table, these bits allow the processor to determine those bits in the information (and not CRC) field. This allows transparent retransmission of the received frame. The Residue Code bits do not go through a FIFO so they change in RR1 when the last character of the frame is loaded into the receive data FIFO. If there are any characters already in the receive data FIFO the Residue Code will be updated before they are read by the processor.

As an example of how the codes are interpreted, consider the case of eight bits per character and a residue code of 101. The number of valid bits for the previous, second previous, and third previous bytes are 0, 7, and 8 respectively. This indicates that the information field (I-field) boundary falls on the second previous byte as shown in Figure 4-14.

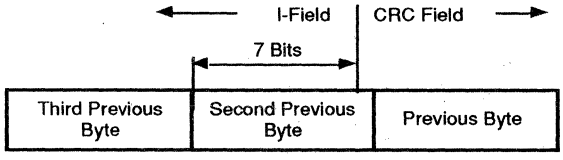


Figure 4-14. Residue Code 101 Interpretation

A frame is terminated by the detection of a closing flag. Upon detection of the flag the following actions take place: the contents of the Receive Shift Register are transferred to the receive data FIFO, the Residue Code is latched, the CRC Error bit is latched and the End of Frame upon reaching the top of the FIFO can cause a special receive condition. The processor can then read RR1 to determine the result of the CRC calculation as well as the Residue Code.

Only the CRC-CCITT polynomial may be used for CRC calculation in SDLC mode, although the generator and checker may be preset to all "1s" or all "0s". The CRC-CCITT polynomial is selected by setting bit D2 of WR5 to "0", bit D7 of WR10 controls the preset value. If this bit is set to "1", the generator and checker are preset to "1s", if this bit is reset, the generator and checker are present to all "0s".

The receiver expects the CRC to be inverted before transmission and so checks the CRC result against the value "0001110100001111". The ISCC presets the CRC checker whenever the receiver is in Hunt mode or whenever a flag is received so a CRC reset command is not strictly necessary. However, the CRC checker may be preset by issuing the Reset CRC Checker command in WR0.

The CRC checker is automatically enabled for all data between the opening and closing flags by the SCC cell in SDLC mode, and the Rx CRC Enable bit (D3) in WR3 is ignored. The result of the CRC calculation for the entire frame is valid in RR1 only when accompanied by the End of Frame bit being set in RR1. At all other times the CRC Error bit in RR1 should be ignored by the processor.

Care must be exercised so that the processor does not attempt to use the CRC bytes that are transferred as data because not all of the bits are transferred properly. The last two bits of CRC are never transferred to the receive data FIFO and are not recoverable.

Note the following about ISCC CRC operation:

The normal CRC checking mechanism involves checking over data and CRC characters. If the division remainder is 0, there is no CRC error.

SDLC is different. The CRC generator, when receiving a correct frame, will have a fixed, non-zero remainder. The actual remainder in the receive CRC calculation must be checked against this fixed value to determine if a CRC error exists.

A frame is terminated by a closing flag. When the ISCC recognizes this flag:

The contents of the Receive Shift register are transferred to the receive data FIFO.

The Residue Code is latched, and the CRC Error bit is latched in the status FIFO and the End of Frame bit is set in the receive status FIFO.

The End of Frame bit, upon reaching the top of the FIFO, will cause a special receive condition. The processor may then read RR1 to determine the result of the CRC calculation as well as the Residue Code. If either the Rx Interrupt or Special Condition Only or the Rx Interrupt on First Character or Special Condition modes are selected, the FIFO will be locked, and the processor must issue an Error Reset command in WR0 to unlock the receive FIFO.

In addition to searching the data stream for flags, the receiver in the ISCC also watches for seven consecutive "1s", which is the abort condition. The presence of seven consecutive "1s" is reported in the Break/Abort bit in RR0. This is one of the possible external/status interrupts, so transitions of this status may be programmed to cause interrupts. Upon receipt of an abort the receiver is forced into Hunt mode where it looks for flags. The Hunt status is also a possible external/status condition whose transition may be programmed to cause an interrupt. The transitions of these two bits occur very close together but either one or two external/status interrupts may result. The abort condition is terminated when a "0" is received, either by itself or as the leading "0" of a flag. The receiver does not leave Hunt mode until a flag has been received so two discrete external/status conditions will occur at the end of an abort. An abort received in the middle of a frame terminates the frame reception, but not in an orderly manner, because the character being assembled is lost.

Up to two modem control signals associated with the receiver are available in SDLC mode:

The /DTR//REQ pin carries inverted state of the DTR bit (D7) in WR5 unless this pin has been programmed to carry a DMA Request signal.

The /DCD pin is ordinarily a simple input to the DCD bit in RR0. However, if the Auto Enables mode is selected by setting bit D5 of WR3 to "1", this pin becomes an enable for the receiver. That is, if Auto Enable is on and the /DCD pin is High the receiver is disabled. While the /DCD pin is Low, the receiver is enabled.

The initialization sequence for the receiver in SDLC mode is WR4 first, to select the mode, then WR10 to modify it if necessary, WR6 to program the address, WR7 to program the flag and WR3 and WR5 to select the various options. At this point the other registers should be initialized as necessary. When all of this is completed the receiver may be enabled by setting bit 0 of WR3 to a one. A summary is shown in Table 4-13.

Table 4-13. Initializing the Receiver in SDLC Mode

Register	Bit No	Description
WR3	6-7	Number of bits per character
WR4	0-1	Select parity
WR5	2	Select CRC-CCITT Generator
WR10	7	Select CRC preset value
	5-6	Select NRZ / NRZI encoding
WR5	7	DTR/REQ
WR6	0-7	Address
WR7	0-7	Flag
WR3	5	Auto enable

NOTE: The receiver searches for synchronization when it is in Hunt mode. In this mode the receiver is idle except that it is searching the data stream for a flag match.

When the receiver detects a flag match it achieves synchronization and interprets the following byte as the address field.

The SYNC/HUNT bit in RR0 reports the Hunt Status and an interrupt can be generated upon transitions between the Hunt state and the Sync state.

The ISCC will drive the /SYNC pin Low to signal that the flag has been received.

4.4.3 SDLC LOOP MODE

The ISCC supports SDLC Loop mode in addition to normal SDLC. SDLC Loop mode is very similar to normal SDLC but is usually used in applications where a point-to-point network is not appropriate (for example, Point-of-Sale terminals). In an SDLC Loop there is a primary controller that manages the message traffic flow on the loop and any number of secondary stations. In SDLC Loop mode, the ISCC operating in regular SDLC mode can act as the primary controller.

A secondary station in an SDLC Loop is always listening to the messages being sent around the loop, and in fact must pass these messages to the rest of the loop by retransmitting them with a one-bit-time delay.

The secondary station can place its own message on the loop only at specific times. The controller signals that secondary stations may transmit messages by sending a

special character, called an EOP (End of Poll), around the loop. The EOP character is the bit pattern 11111110.

When a secondary station has a message to transmit and recognizes an EOP on the line, it changes the last binary 1 of the EOP to a 0 before transmission. This has the effect of turning the EOP into a flag pattern. The secondary station now places its message on the loop and terminates its message with an EOP. Any secondary stations further down the loop with messages to transmit can append their messages to the message of the first secondary station by the same process.

All secondary stations without messages to send merely echo the incoming messages and are prohibited from placing messages on the loop, except upon recognizing an EOP.

SDLC Loop mode is quite similar to normal SDLC mode except that two additional control bits are used. Writing a 1 to the Loop Mode bit in WR10 configures the ISCC for Loop mode. Writing a 1 to the Go Active on Poll bit in the same register normally causes the ISCC to change the next EOP into a flag and then begin transmitting on loop. However, when the ISCC first goes on loop it uses the first EOP as a signal to insert the one-bit delay, and doesn't begin transmitting until it receives the second EOP. There are also two additional status bits in RR10, the On Loop bit and the Loop Sending bit.

There are also restrictions as to when and how a secondary station physically becomes part of the loop.

A secondary station that has just powered up must monitor the loop, without the one-bit-time delay, until it recognizes an EOP. When an EOP is recognized the one-bit-time delay is switched on. This does not disturb the loop because the line is marking idle between the time that the controller sends the EOP and the time that it receives the EOP back. The secondary station that has gone on-loop cannot place a message on the loop until the next time that an EOP is issued by the controller. A secondary station goes off-loop in a similar manner. When given a command to go off-loop, the secondary station waits until the next EOP to remove the one-bit-time delay.

To operate the ISCC in SDLC Loop mode, the ISCC must first be programmed just as if normal SDLC were to be used. Loop mode is then selected by writing the appropriate control word in WR10; the ISCC is now waiting for the EOP so that it can go on loop. While waiting for the EOP, the ISCC ties TxD to RxD with only the internal gate delays in the signal path. When the first EOP is recognized by the ISCC, the Break/Abort/EOP bit is set in RR0, generating an External/Status interrupt (if so enabled). At the same time, the On-Loop bit in RR10 is set to indicate that the ISCC is indeed on-loop, and a one-bit time delay is inserted in the TxD to the RxD path.

The ISCC is now on-loop but cannot transmit a message until a flag and the next EOP are received. The requirement that a flag be received ensures that the ISCC cannot erroneously send messages until the controller ends the current polling sequence and starts another one.

If the CPU in the secondary station with ISCC needs to transmit a message, the Go-Active-On-Poll bit in WR10 must be set. If this bit is set when the EOP is detected, the ISCC changes the EOP to a flag and starts sending another flag. The EOP is reported in the Break/Abort/EOP bit in RR0 and the CPU should write its data bytes to the ISCC, just as in normal SDLC frame transmission. When the frame is complete and CRC has been sent, the ISCC closes with a flag and reverts to One-Bit-Delay mode. The last zero of the flag, along with the marking line echoed from the RxD pin, form an EOP for secondary stations further down the loop.

While the ISCC is actually transmitting a message, the loop-sending bit in R10 is set to indicate this.

If the Go-Active-On-Poll bit is not set at the time the EOP passes by, the ISCC cannot send a message until a flag (terminating the current polling sequence) and another EOP are received.

If SDLC loop is de-selected, the ISCC is designed to exit from the loop gracefully. When SDLC Loop mode is de-selected by writing to WR10; the ISCC waits until the next polling cycle to remove the one-bit time delay.

If a polling cycle is in progress at the time the command is written, the ISCC finishes sending any message that it may be transmitting, ends with an EOP, and disconnects TxD from RxD. If no message was in progress, the ISCC immediately disconnects TxD from RxD.

Once the ISCC is not sending on the loop, an exit from the loop is accomplished by setting the Loop Mode bit in WR10 to "0", and at the same time writing the Abort/Flag on Underrun and Mark/Flag idle bits with the desired values. The ISCC will revert to normal SDLC operation as soon as an EOP is received, or immediately, if the receiver is already in Hunt mode because of the receipt of an EOP.

To ensure proper loop operation after the ISCC goes off the loop, and until the external relays take the ISCC completely out of the loop, the ISCC should be programmed for Mark idle instead of Flag idle. When the ISCC goes off the loop, the On-Loop bit is reset.

NOTE: With NRZI encoding, removing the stations from the loop (removing the one-bit time delay) may cause

problems further down the loop because of extraneous transitions on the line. The ISCC avoids this problem by making transparent adjustments at the end of each frame it sends in response to an EOP. A response frame from the ISCC is terminated by a flag and EOP. Normally, the flag and the EOP share a zero, but if such sharing would cause the RxD and TxD pins to be of opposite polarity after the EOP, the ISCC adds another zero between the flag and the EOP. This causes an extra line transition so that RxD and TxD are identical after the EOP is sent. This extra zero is completely transparent because it only means that the flag and the EOP no longer share a zero. All that a proper loop exit needs, therefore, is the removal of the one-bit delay.

The ISCC allows the user the option of using NRZI in SDLC Loop mode by programming WR10 appropriately. With NRZI encoding, the outputs of secondary stations in the loop may be inverted from their inputs because of messages that they have transmitted.

The initialization sequence for the SCC cell in SDLC Loop mode is similar to the sequence used in SDLC mode, except that it is somewhat longer. The processor should program WR4 first, to select SDLC mode, and then WR10 to select the CRC preset value and program the Mark/Flag idle bit. The Loop Mode and Go Active On Poll bits in WR10 should not be set to "1" yet. The flag is written in WR7 and the various options are selected in WR3 and WR5. At this point the other registers should be initialized as necessary, as shown in Table 4-14.

Table 4-14. SDLC Loop Mode initialization

Register	Bit No	Description
WR4	5-4	Select SDLC mode
WR10	7	Select CRC preset value
	3	Select mark / flag idle bit
WR7		Flag
WR3	7-6	Select bits per character for receiver
	1	Sync character load inhibit
	2	Address search mode
	5	Auto enables
WR5	6-5	Select bits per character for transmitter
	4	Send break
	2	Select SDLC CRC
	7	
	1	
WR4	1-0	Select parity
	7-6	Select clock mode
WR6	0-7	Address
WR10	6-5	Select data encoding

Then the Loop Mode bit (D1) in WR10 should be set to "1". When all of this is complete the transmitter may be enabled by setting bit D3 of WR5 to "1". Now that the transmitter is enabled, the CRC generator may be initialized by issuing the Reset Tx CRC Generator command in WR0. The receiver is enabled by setting the Go Active on Poll bit (D4) in WR10 to "1". The ISCC will go on the loop when seven consecutive "1s" are received, and will signal this by setting the On Loop bit in RR10. Note that the seven consecutive "1s" will set the Break/Abort and Hunt bits in RR0 also. Once the ISCC is on the loop, the Go Active on Poll bit should be set to "0" until a message is to be transmitted on the loop. To transmit a message on the loop, the Go Active on Poll bit should be set to "1". At this point the processor may either write the first character to the transmit buffer and wait for a transmit buffer empty condition, or wait for the Break/Abort and Hunt bits to be set in RR10 and the Loop Sending bit to be set in RR10 before writing the first data to the transmitter. The Go Active On Poll bit should be set to "0" after the transmission of the frame has begun. To go off of the loop, the processor should set the Go Active On Poll bit in WR10 to "0" and then wait for the Loop Sending bit in RR10 to be set to "0". At this point the Loop Mode bit (D1) in WR10 is set to "0" to request an orderly exit from the loop. The ISCC will exit SDLC Loop mode when seven consecutive "1s" have been received; at the same time the Break/Abort and Hunt bits in RR0 will be set to "1", and the On Loop bit in RR10 will be set to "0".

4.4.4 SDLC Loop Mode Receive

SDLC Loop mode is quite similar to SDLC mode except that two additional control bits are used. They are the Loop Mode bit (D1) and the Go Active on Poll bit (D4) in WR10. In addition to these two extra control bits, there are also two status bits in RR10. They are the On Loop bit (D1) and the Loop Sending bit (D4).

Before Loop mode is selected both the receiver and transmitter must be completely initialized for SDLC operation. Once this is done, Loop mode is selected by setting bit D1 of WR10 to "1". At this point the ISCC connects TxD to RxD with only gate delays in the path. At the same time a flag is loaded into the Transmit Shift register, and is shifted to the end of the zero inserter, ready for transmission. The ISCC will remain in this state until the Go Active on Poll bit (D4) in WR10 is set to "1". When this bit is set to "1" the receiver begins looking for a sequence of seven consecutive "1s", indicating either an EOP or an idle line. When the receiver detects this condition the Break/Abort bit in RR0 is set to "1" and a one-bit time delay is inserted in the path from RxD to TxD. The On Loop bit in RR10 is also set to "1" at this time, and the receiver enters the Hunt mode. The ISCC cannot transmit on the loop until a flag is received, causing the receiver to leave Hunt mode, and another EOP (bit pattern "11111110") is received. The

ISCC is now on the loop and capable of transmitting on the loop. As soon as this status is recognized by the processor, the Go Active On Poll bit in WR10 should be set to "0" to prevent the ISCC from transmitting on the loop without the consent of the processor.

4.4.5 SDLC Loop Mode Transmit

To transmit a message on the loop, the Go Active On Poll bit in WR10 must be set to "1". Once this is done, the ISCC will change the next received EOP into a Flag and begin transmitting on the loop.

When the EOP is received, the Break/Abort and Hunt bits in RR0 will be set to "1", and the Loop Sending bit in RR10 will also be set to "1". Data to be transmitted may be written after the Go Active On Poll bit has been set or after the receiver enters Hunt mode.

If the data is written immediately after the Go Active On Poll bit has been set, the ISCC will only insert one flag after the EOP is changed into a flag. If the data is not written until after the receiver enters the Hunt mode, the flags will be transmitted until the data is written. If only one frame is to be transmitted on the loop in response to an EOP, the processor must set the Go Active on Poll bit to "0" before the last data is written to the transmitter. In this case the transmitter will close the frame with a single flag, and then revert to the one-bit delay. The Loop Sending bit in RR10 is set to "0" when the closing Flag has been sent. If more than one frame is to be transmitted, the Go Active On Poll bit should not be set to "0" until the last frame is being sent. If this bit is not set to "0" before the end of a frame, the transmitter will send Flags until either more data is written to the transmitter, or until the Go Active On Poll bit is set to "0". Note that the state of the Abort/Flag on Underrun and Mark/Flag idle bits in WR10 are ignored by the ISCC in SDLC Loop mode.

CHAPTER 5
REGISTER DESCRIPTIONS

5.1 INTRODUCTION

This section describes the function of the various bits in the registers of the device. Throughout this section the following conventions will be used:

Control bits may be written and read by the CPU and will not be modified by the device. Command bits may be written by the CPU to initiate an action in the device and will be read as zeros. Status bits are controlled by the device and may be read to check device status. Any writes to status bits are ignored by the device. Command/status bits are controlled by both the device and the CPU. They may be written and read by the CPU and may also be modified by the device.

Reserved bits are not used in this implementation of the device and may or may not be physically present in the device. Reserved bits that are physically present will be readable and writable but reserved bits that are not present will always be read as zero. To ensure compatibility with future versions of the device reserved bits should always be written with zeros. Reserved commands should not be used for the same reason.

5.2 REGISTER DESCRIPTIONS

Register can be accessed through either channel, the Interrupt Vector Read Register returns the interrupt vector with status if read from Channel B and without status if read from Channel A, and Channel A has an additional read register which contains all the Interrupt Pending bits.

5.2.1 Write Registers, SCC Cell

Ten write registers are used for control, two for sync character generation, and two for baud rate generation. In addition, there are two write registers which are shared by both channels; one is the interrupt vector register, and one is the master interrupt control and reset register. See Table 5-1 for a summary on write registers.

Table 5-1. SCC Cell Write Registers

Register	Description
WR0	Register Pointers, various initialization commands
WR1	Transmit and Receive interrupt enables, WAIT/DMA commands
WR2	Interrupt Vector
WR3	Receive parameters and control modes
WR4	Transmit and Receive modes and parameters
WR5	Transmit parameters and control modes
WR6	Sync Character or SDLC address
WR7	Sync Character or SDLC flag
WR8	Transmit buffer
WR9	Master Interrupt control and reset commands
WR10	Misc. transmit and receive control bits
WR11	Clock mode controls for receive and transmit
WR12	Lower byte of baud rate generator
WR13	Upper byte of baud rate generator
WR14	Miscellaneous control bits
WR15	External status interrupt enable control

5.2.2 Read Registers, SCC Cell

Four read registers indicate status information, two are for baud rate generation, and one for the receive buffer. In addition, there are two read registers which are shared by both channels: one for the interrupt pending bits and one for interrupt vector. See Table 5-2 for a summary on the SCC cell read registers.

Table 5-2. SCC Cell Read Registers

Register	Description
RR0	Transmit and Receive buffer status and external status
RR1	Special Receive Condition status
RR2	Modified interrupt vector (Channel B only), Unmodified interrupt vector (Channel A only)
RR3	Interrupt pending bits (Channel A only)
RR6	SDLC FIFO byte counter lower byte (only when enabled)
RR7	SDLC FIFO byte count and status (only when enabled)
RR8	Receive buffer
RR10	Miscellaneous status bits
RR12	Lower byte of baud rate generator time constant
RR13	Upper byte of baud rate generator time constant
RR15	External Status interrupt information

5.2.3 DMA Registers

The DMA cell contains 16 read write registers for control of the DMA channels. The DMA possesses its own interrupt vector register and interrupt control registers which are independent of the SCC cell. The DMA cell also includes the Bus Configuration Register (BCR) for the ISCC. The addresses, names and descriptions of these registers are given in Table 5-3.

Table 5-3. DMA Cell Register Description

Address	Name	Description
xxxxx	BCR	Bus Configuration Register
00000	CCAR	Channel Command/Address Register (Write)
00000	DSR	DMA Status Register (Read)
00001	ICR	Interrupt Control Register
00010	IVR	Interrupt Vector Register
00011	ICSR	Interrupt Command Register (Write)
00011	ISR	Interrupt Status Register (Read)
00100	DER	DMA Enable/Disable Register
00101	DCR	DMA Control Register
00110		Reserved Address
00111		Reserved Address
01000-01001	RDCRA	Receive DMA Count Register Channel A (Low-high byte)
01010-01011	TDCRA	Transmit DMA Count Register Channel A
01100-01101	RDCRB	Receive DMA Count Register Channel B
01110-01111	TDCRB	Transmit DMA Count Register Channel B
10000-10011	RDARA	Receive DMA Address Register Channel A
10100-10111	TDARA	Transmit DMA Address Register Channel A
11000-11011	RDARB	Receive DMA Address Register Channel B
11100-11111	TDARB	Transmit DMA Address Register Channel B

5.3 SCC CELL REGISTER OVERVIEW

The SCC cell write register set in each channel includes ten control registers (among them is the transmit buffer), two sync character registers and two baud rate time constant registers. The interrupt control register and the master interrupt control and reset register are shared by both channels.

The only variation in register definition is between the multiplexed and non-multiplexed bus mode programming of the ISCC. The variation exists in the command decode structure; register WR0. The following sections describe in detail each write register and the associated bit configuration for each.

5.4 WRITE REGISTERS

The following sections describe WR registers in detail.

5.4.1 Write Register 0 (Command Register)

WR0 is the command register and the CRC reset code register. WR0 takes on slightly different forms depending upon whether the ISCC is in the multiplexed or non-multiplexed bus mode of operation. Figure 5-1 shows the bit configuration for the non-multiplexed mode and includes register select bits in addition to command and reset codes.

Figure 5-2 shows the bit configuration for the multiplexed mode and includes (in Channel B only) the address decoding select described later.

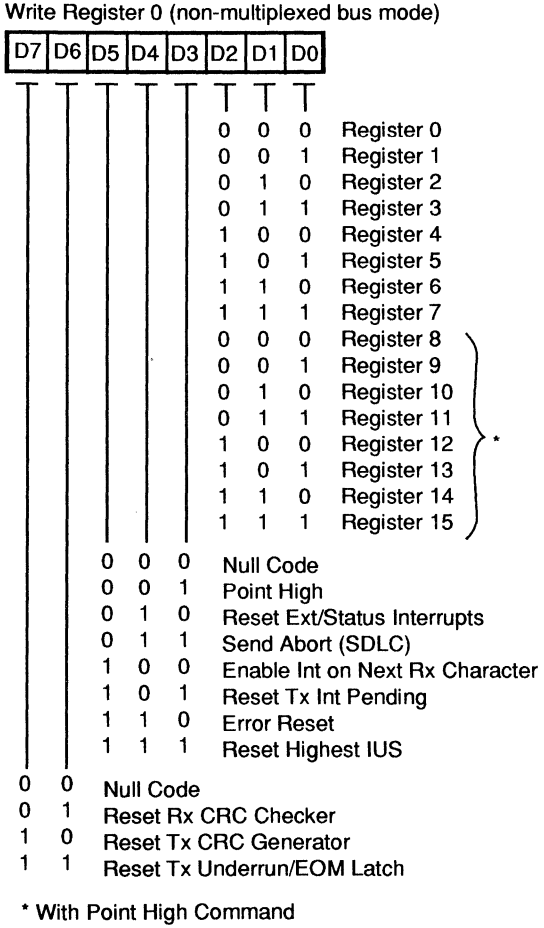


Figure 5-1. WR0 in the Non-multiplexed Bus Mode

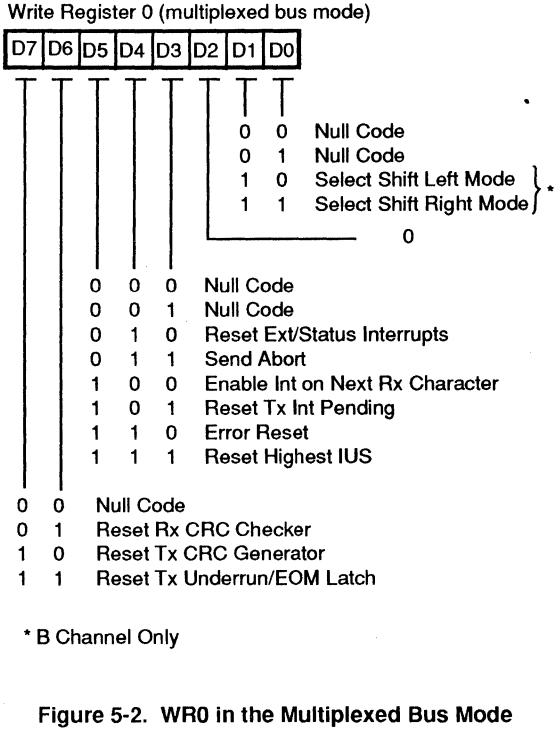


Figure 5-2. WR0 in the Multiplexed Bus Mode

The following bit description for WR0 is identical for both versions except where specified.

Bits D7 and D6 are the CRC Reset Codes 1 and 0.

Bit combination 00 is a Null Command.

This command has no effect on the ISCC SCC cell and is used when a write to WR0 is necessary for some reason other than a CRC Reset command.

Bit combination 01 is the Reset Receive CRC Checker Command.

This command is used to initialize the receive CRC circuitry. It is necessary in synchronous modes (except SDLC) if the Enter Hunt Mode command in Write Register 3 is not issued between received messages. Any action that disables the receiver initializes the CRC circuitry. Resetting the Receive CRC Checker command is accomplished automatically in SDLC mode.

Bit combination 10 is the Reset Transmit CRC Generator Command.

This command initializes the CRC generator. It is usually issued in the initialization routine and after the CRC has been transmitted. A Channel Reset will not initialize the generator and this command should not be issued until after the transmitter has been enabled in the initialization routine.

Bit combination 11 is the Reset Transmit Underrun/EOM Latch Command.

This command controls the transmission of CRC at the end of transmission (EOM). If this latch has been reset, and a transmit underrun occurs, the SCC cell automatically appends CRC to the message. In SDLC mode with Abort on Underrun selected, the SCC cell sends an abort, and Flag on underrun if the TX Underrun/EOM latch has been reset.

At the start of the CRC transmission, the Tx Underrun/EOM latch is set. The Reset command can be issued at any time during a message. If the transmitter is disabled, this command will not reset the latch. However, if no External Status interrupt is pending, or if a Reset External Status interrupt command accompanies this command while the transmitter is disabled, an External/Status interrupt is generated with the Tx Underrun/EOM bit reset in RR0.

Bits D5-D3 are the Command Codes for the SCC Cell.

Bit combination 000 is a Null Command.

The Null command has no effect on the SCC.

Bit combination 001 is the Point High Command.

This command effectively adds eight to the Register Pointer (D2-D0) by allowing WR8 through WR15 to be accessed. The Point High command and the Register Pointer bits are written simultaneously. This command is used when the ISCC is configured to be in the non-multiplexed bus mode. Note that WR0 changes form depending upon the bus mode selection.

Bit combination 010 is the Reset External/Status Interrupts Command.

After an External/Status interrupt (a change on a modem line or a break condition, for example), the status bits in RR0 are latched. This command re-enables the bits and allows interrupts to occur again as a result of a status change. Latching the status bits captures short pulses until the CPU has time to read the change.

The SCC cell contains simple queueing logic associated with most of the external status bits in RR0. If another External/Status condition changes while a previous condition is still pending (Reset External/Status Interrupts has not yet been issued) and this condition persists until after the command is issued, this second change causes another External/Status interrupt. However, if this second status change does not persist (there are two transitions), another interrupt is not generated. Exceptions to this rule are detailed in the RR0 description.

Bit combination 011 is the Send Abort Command.

This command is used in SDLC mode to transmit a sequence of eight to thirteen "1s." This command always empties the transmit buffer and sets Tx Underrun/EOM bit in Read Register 0.

Bit combination 100 is the Enable Interrupt On Next Rx Character Command.

If the interrupt on First Received Character mode is selected, this command is used to reactivate that mode after each message is received. The next character to enter the receive FIFO causes a Receive interrupt. Alternatively, the first previously stored character in the FIFO will cause a Receive interrupt.

Bit combination 101 is the Reset Tx Interrupt Pending Command.

This command is used in cases where there are no more characters to be sent; e.g. at the end of a message. This command prevents further transmit interrupts until after the next character has been loaded into the transmit buffer or until CRC has been completely sent. This command is

necessary to prevent the transmitter from requesting an interrupt when the transmit buffer becomes empty (with Transmit Interrupt Enabled).

Bit combination 110 is the Error Reset Command.

This command resets the error bits in RR1. If interrupt on first Rx Character or interrupt on Special Condition modes are selected and a special condition exists, the data with the special condition is held in the receive FIFO until this command is issued. If either of these modes is selected and this command is issued before the data has been read from the receive FIFO, the data is lost.

Bit combination 111 is the Reset Highest IUS Command.

This command resets the highest priority Interrupt Under Service (IUS) bit, allowing lower priority conditions to request interrupts. This command allows the use of the internal daisy-chain (even in systems without an external daisy-chain) and should be the last operation in an interrupt service routine.

Bits 2 through 0 are the Register Selection Code when the device is programmed to be in the non-multiplexed bus mode. These three bits select Registers 0 through 7. With the Point High command, Registers 8 through 15 are selected.

In the multiplexed bus mode, bits D2 through D0 have the following function.

Bit D2 must be programmed as "0." Bits D1 and D0 select Shift Left/Right; that is WR0(1-0)=10 for shift left and WR0(1-0)=11 for shift right.

5.4.2 Write Register 1 (Transmit/Receive Interrupt and Data Transfer Mode Definition)

Write Register 1 is the control register for the various SCC cell interrupt and Wait/Request modes. Figure 5-3 shows the bit assignments for WR1.

Table 5-4. SCC Cell Register Address Map Using Pointer (Non-multiplexed Bus Mode)

Using Null Command

A0/A//B	Address D2 D1 D0	Write Register	Read Register
0	000	WR0B	RR0B
0	001	WR1B	RR1B
0	010	WR2	RR2B
0	011	WR3B	RR3B
0	100	WR4B	(RR0B)
0	101	WR5B	(RR1B)
0	110	WR6B	(RR2B)
0	111	WR7B	(RR3B)
1	000	WR0A	RR0A
1	001	WR1A	RR1A
1	010	WR2	RR2A
1	011	WR3A	RR3A
1	100	WR4A	(RR0A)
1	101	WR5A	(RR1A)
1	110	WR6A	(RR2A)
1	111	WR7A	(RR3A)

Using Point High Command

A0/A//B	Address D2 D1 D0	Write Register	Read Register
0	000	WR8B	RR8B
0	001	WR9	(RR13B)
0	010	WR10B	RR10B
0	011	WR11B	(RR15B)
0	100	WR12B	RR12B
0	101	WR13B	RR13B
0	110	WR14B	(RR10B)
0	111	WR15B	RR15B
1	000	WR8A	RR8A
1	001	WR9A	(RR13A)
1	010	WR10A	RR10A
1	011	WR11A	(RR15A)
1	100	WR12A	RR12A
1	101	WR13A	RR13A
1	110	WR14A	(RR10A)
1	111	WR15A	RR15A

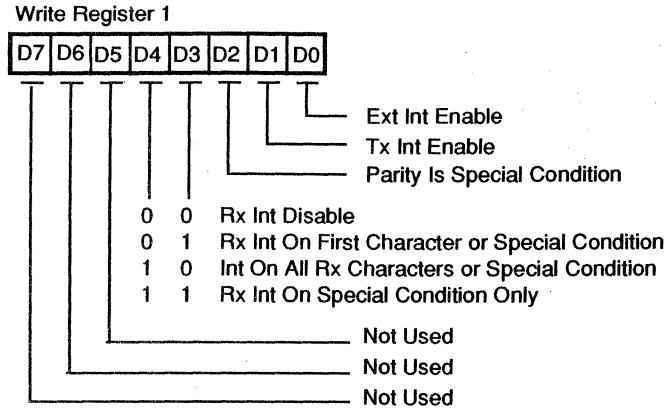


Figure 5-3. Write Register 1

Bit 7, 6, and 5 are not used in the ISCC. These bits were used in the SCC cell to control the action of the /WAIT//REQUEST pin but they have no function in the ISCC since this pin does not exist in this device. For code compatibility purposes, there is no restriction on how these bits are programmed.

Bit 4 and 3 specify the various character-available conditions that may cause interrupt requests.

Bit combination 00 programs Receive Interrupts Disabled. This mode prevents the receiver from requesting an interrupt and is normally used in a polled environment where either the status bits in RR0 or the modified vector in RR2 (Channel B) can be monitored to initiate a service routine. Although the receiver interrupts are disabled, a special condition can still provide a unique vector status in RR2.

Bit combination 01 programs Receive Interrupt on First Character or Special Condition. The receiver requests an interrupt in this mode on the first available character (or stored FIFO character) or on a special condition. Sync characters to be stripped from the message stream do not cause interrupts.

Special receive conditions are: receiver overrun, framing error, end of frame, or parity error (if selected). If a special receive condition occurs, the data containing the error is stored in the receive FIFO until an Error Reset command is issued by the CPU.

This mode is usually selected when a Block Transfer mode is used. In this interrupt mode, a pending special receive condition remains set until either an Error Reset Command, a channel or hardware reset, or until receive interrupts are disabled.

The Receive Interrupt on First Character or Special Condition mode can be re-enabled by the Enable Rx Interrupt on Next Character command in WR0.

Bit combination 10 programs Interrupt on All Receive Characters or Special Condition. This mode allows an interrupt for every character received (or character in the receive FIFO) and provides a unique vector when a special condition exists. The Receiver Overrun bit and the Parity Error bit in RR1 are two special conditions that are latched. These two bits must be reset by the Error Reset command. Receiver overrun is always a special receive condition, and parity can be programmed to be a special condition.

Data characters with special receive conditions are not held in the receive FIFO in the Interrupt On All Receive Characters or Special Conditions Mode as they are in the other receive interrupt modes.

Bit combination 11 programs Receive Interrupt on Special Condition. This mode allows the receiver to interrupt only on characters with a special receive condition. When an interrupt occurs, the data containing the error is held in the receive FIFO until an Error Reset command is issued. When using this mode in conjunction with a DMA, the DMA can be initialized and enabled before any characters have been received by the SCC. This eliminates the time-critical section of code required in the Receive Interrupt on First Character or Special Condition mode; i.e. all data can be transferred via the DMA so that the CPU need not handle the first received character as a special case.

Bit 2 selects Parity Is Special Condition.

If this bit is set to "1," any received characters with parity not matching the sense programmed in WR4 give rise to a Special Receive Condition. If parity is disabled (WR4), this bit is ignored. A special condition modifies the status of the interrupt vector stored in WR2. During an interrupt acknowledge cycle, this vector can be placed on the data bus.

Bit 1 is the Transmitter Interrupt Enable.

If this bit is set to "1," the transmitter requests an interrupt whenever the transmit buffer becomes empty.

Bit 0 is the External/Status Master Interrupt Enable.

This bit is the master enable for External/Status interrupts including /DCD, /CTS, /SYNC pins, break, abort, the beginning of CRC transmission when the Transmit/Underrun/EOM latch is set, or when the counter in the baud rate generator reaches "0." Write Register 15 contains the individual enable bits for each of these sources of External/Status interrupts. This bit is reset by a channel or hardware reset.

5.4.3 Write Register 2 (Interrupt Vector)

WR2 is the interrupt vector register. Only one vector register exists in the SCC cell, but it can be accessed through either channel. The interrupt vector can be modified by status information. This is controlled by the Vector Includes Status (VIS) and the Status High/Status Low bits in WR9. The bit positions for WR2 are shown in Figure 5-4. Note that the DMA cell has its own interrupt vector register.

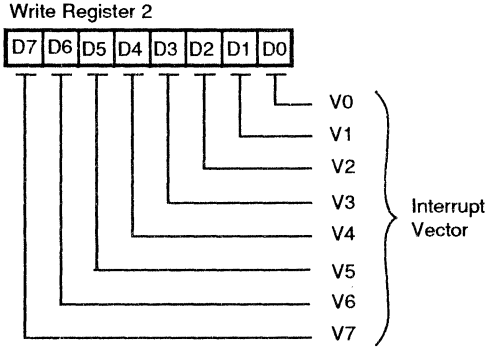


Figure 5-4. Write Register 2

5.4.4 Write Register 3 (Receive Parameters and Control)

This register contains the control bits and parameters for the receiver logic as illustrated in Figure 5-5.

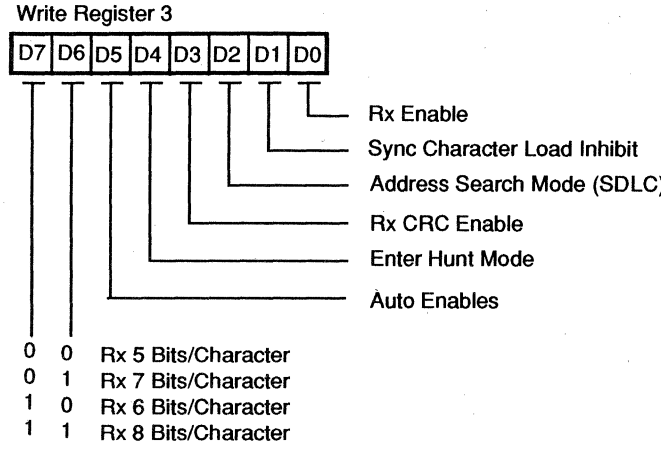


Figure 5-5. Write Register 3

Bit 7 and 6 select the Receiver Bits/Character.

The state of these two bits determines the number of bits to be assembled as a character in the received serial data stream. The number of bits per character can be changed while a character is being assembled but only before the number of bits currently programmed is reached. Unused bits in the Received Data Register (RR8) are set to "1" in asynchronous modes. In synchronous modes and SDLC modes, the ISCC merely transfers an 8-bit section of the serial data stream to the receive FIFO at the appropriate time. Table 5-5 lists the number of bits per character in the assembled character format.

Table 5-5. Receive Bits per Character

D7	D6	Bits/Character
0	0	5
0	1	7
1	0	6
1	1	8

Bit 5 selects Auto Enables.

This bit programs the function for both the /DCD and /CTS pins. /CTS becomes the transmitter enable and /DCD becomes the receiver enable when this bit is set to "1." However, the Receiver Enable and Transmit Enable bits must be set before the /DCD and /CTS pins can be used in this manner. When the Auto Enables bit is set to "0," the /DCD and /CTS pins are merely inputs to the corresponding status bits in Read Register 0. The state of /DCD is ignored in the Local Loopback mode. The state of /CTS is ignored in both Auto Echo and Local Loopback modes.

Bit 4 forces the SCC cell to Enter Hunt Mode.

This command forces the comparison of sync characters or flags to assembled receive characters for the purpose of synchronization. After reset, the ISCC cell automatically enters the Hunt mode (except asynchronous). Whenever a flag or sync character is matched, the Sync/Hunt bit in Read Register 0 is reset and, if External/Status Interrupt Enable is set, an interrupt sequence is initiated. The ISCC automatically enters the Hunt mode when an abort condition is received or when the receiver is enabled.

Bit 3 is the Receiver CRC Enable.

This bit is used to initiate CRC calculation at the beginning of the last byte transferred from the Receiver Shift register to the receive FIFO. This operation occurs independently of the number of bytes in the receive FIFO. When a particular byte is to be excluded from CRC calculation, this bit should be reset before the next byte is transferred to the receive FIFO. If this feature is used, care must be taken to ensure that eight bits per character is selected in the receiver because of an inherent delay from the Receive Shift register to the CRC checker.

This bit is internally set to "1" in SDLC mode and the ISCC calculates CRC on all bits except inserted zeros between the opening and closing character flags. This bit is ignored in asynchronous modes.

Bit 2 selects the Address Search Mode (SDLC).

Setting this bit in SDLC mode causes messages with addresses not matching the address programmed in WR6 to be rejected. No receiver interrupts can occur in this mode unless there is an address match. The address that the ISCC attempts to match can be unique (1 in 256) or multiple (16 in 256), depending on the state of Sync Character Load Inhibit bit. The Address Search mode bit is ignored in all modes except SDLC.

Bit 1 is the SYNC Character Load Inhibit.

If this bit is set to "1" in any mode except SDLC, the ISCC compares the byte in WR6 with the byte about to be stored in the FIFO, and it inhibits this load if the bytes are equal. (Caution this also occurs in the asynchronous mode if the received character matches the contents of WR6.) The ISCC does not calculate the CRC on bytes stripped from the data stream in this manner. If the 6-bit sync option is selected while in Monosync mode, the compare is still across eight bits, so WR6 must be programmed for proper operation.

If the 6-bit sync option is selected with this bit set to "1," all sync characters except the one immediately preceding the data are stripped from the message. If the 6-bit sync option is selected while in the Bisync mode, this bit is ignored.

The address recognition logic of the receiver is modified in SDLC mode if this bit is set to "1," i.e. only the four most significant bits of WR6 must match the receiver address. This procedure allows the ISCC to receive frames from up to 16 separate sources without programming WR6 for each source (if each station address has the four most significant bits in common). The address field in the frame is still eight bits long.

The bit is ignored in SDLC mode if Address Search mode has not been selected.

Bit 0 is the Receiver Enable.

When this bit is set to "1," receiver operation begins. This bit should be set only after all other receiver parameters are established and the receiver is completely initialized. This bit is reset by a channel or hardware reset command, and it disables the receiver.

5.4.5 Write Register 4 (Transmit/Receiver Miscellaneous Parameters and Modes)

WR4 contains the control bits for both the receiver and the transmitter. These bits should be set in the transmit and receiver initialization routine before issuing the contents of WR1, WR3, WR6, and WR7. Bit positions for WR4 are shown in Figure 5-6.

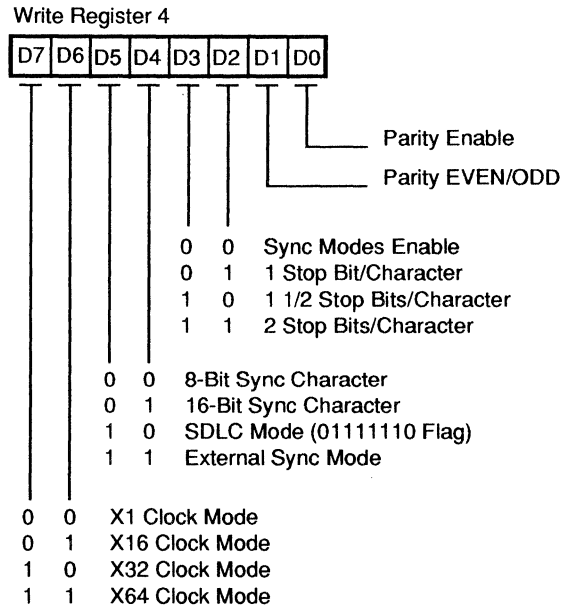


Figure 5-6. Write Register 4

Bit 7 and 6 are the Clock Mode, Bits 1 And 0.

These bits specify the multiplier between the clock and data rates. In synchronous modes, the 1X mode is forced internally and these bits are ignored unless External Sync mode has been selected.

Bit combination 00 selects the 1X Mode. The clock rate and data rate are the same. In External Sync mode, this bit combination specifies that only the /SYNC pin can be used to achieve character synchronization.

Bit combination 01 selects the 16X Mode. The clock rate is 16 times the data rate. In External Sync mode, this bit combination specifies that only the /SYNC pin can be used to achieve character synchronization.

Bit combination 10 selects the 32X Mode. The clock rate is 32 times the data rate. In External Sync mode, this bit combination specifies that either the /SYNC pin or a match with the character stored in WR7 will signal character synchronization. The sync character can be either six or eight bits long as specified by the 6-bit/8-bit Sync bit in WR10.

Bit combination 11 selects the 64X Mode. The clock rate is 64 times the data rate. With this bit combination in External Sync mode, both the receiver and transmitter are placed in SDLC mode. The only variation from normal SDLC operation is that the /SYNC pin can be used to start or stop the reception of a frame by forcing the receiver to act as though a flag had been received.

Bits 5 and 4 are the SYNC Mode selection Bits 1 And 0.

These two bits select the various options for character synchronization. They are ignored unless synchronous modes are selected in the stop bits field of this register.

Bit combination 00 selects the Monosync mode. In this mode, the receiver achieves character synchronization by matching the character stored in WR7 with an identical character in the received data stream. The transmitter uses the character stored in WR6 as a time fill. The sync character can be either six or eight bits, depending on the state of the 6-bit/8-bit Sync bit in WR10. if the Sync Character Load Inhibit bit is set, the receiver strips the contents of WR6 from the data stream if received within character boundaries.

Bit combination 01 selects the Bisync mode. The concatenation of WR7 with WR6 is used for receiver synchronization and as a time fill by the transmitter. The sync character can be 12 or 16 bits in the receiver, depending on the state of the 6-bit/8-bit Sync bit in WR10. The transmitted character is always 16 bits.

Bit combination 10 selects the SDLC Mode. In this mode, SDLC is selected and requires a Flag (01111110) to be written to WR7. The receiver address field should be written to WR6. The SDLC CRC polynomial must also be selected (WR5) in SDLC mode.

Bit combination 11 selects the External Sync Mode. In this mode, the ISCC expects external logic to signal character synchronization via the /SYNC pin. If the crystal oscillator option is selected (in WR11), the internal /SYNC signal is forced to "0." In this mode, the transmitter is in Monosync mode using the contents of WR6 as the time fill with the sync character length specified by the 6-bit/8-bit Sync bit in WR10.

Bits 3 and 2 are the Stop Bits selection, Bits 1 and 0.

These bits determine the number of stop bits added to each asynchronous character that is transmitted. The receiver always checks for one stop bit in Asynchronous mode. A Special mode specifies that a Synchronous mode is to be selected. D2 is always set to "1" by a channel or hardware reset to ensure that the /SYNC pin is in a known state after a reset.

Bit combination 00 selects Synchronous Modes Enable. This bit combination selects one of the synchronous modes specified by bits D4, D5, D6, and D7 of this register and forces the 1X Clock mode internally.

Bit combination 01 selects 1 Stop Bit/Character. This bit combination selects Asynchronous mode with one stop bit per character.

Bit combination 10 selects 1 1/2 Stop Bits/Character. These bits select Asynchronous mode with 1-1/2 stop bits per character. This mode can not be used with the 1X clock mode.

Bit combination 11 selects 2 Stop Bits/Character. These bits select Asynchronous mode with two stop bits per transmitted character and check for one received stop bit.

Bit 1 is the Parity Even//Odd select bit.

This bit determines whether parity is checked as an even or odd. A "1" programmed here selects even parity, and a "0" selects odd parity. This bit is ignored if the Parity enable bit is not set.

Bit 0 is the Parity Enable.

When this bit is set, an additional bit position beyond those specified in the bits/character control is added to the transmitted data and is expected in the receive data. The Received Parity bit is transferred to the CPU as part of the data unless eight bits per character is selected in the receiver.

5.4.6 Write Register 5 (Transmit Parameter and Controls)

WR5 contains control bits that affect the operation of the transmitter. B2 affects both the transmitter and the receiver. Bit positions for WR5 are shown in Figure 5-7.

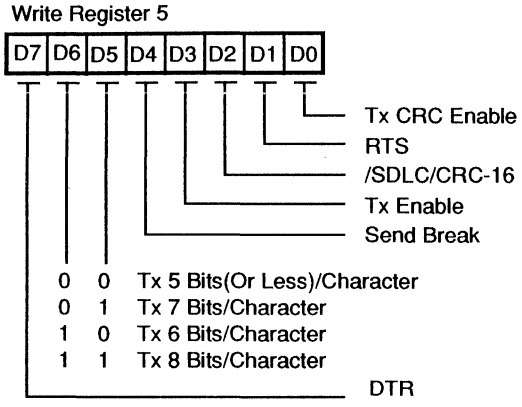


Figure 5-7. Write Register 5

Bit 7 is the Data Terminal Ready control bit.

This is the control bit for the /DTR//REQ pin while the pin is in the DTR mode (selected in WR14). When set, /DTR is Low; when reset, /DTR is High. This bit is ignored when /DTR//REQ is programmed to act as a /REQUEST pin. This bit is reset by a channel or hardware reset. Refer to the description of Bit 2 in Write Register 14.

Bits 6 and 5 are the Transmit Bits/Character select bits 1 and 0.

These bits control the number of bits in each byte transferred to the transmit buffer. Bits sent must be right justified with least significant bits first.

The Five Or Less mode allows transmission of one to five bits per character; however, the CPU should format the data character as shown below in Table 5-6. In the Six or Seven Bits/Character modes, unused data bits are ignored.

Table 5-6. Transmit Bits per Character

Bit 7	Bit 6	Bits/Character
0	0	5 or less bits / character
0	1	7 bits / character
1	0	6 bits / character
1	1	8 bits / character

For five or less bits per charater selection in WR5, the following encoding is used in the data sent to the transmitter. D is the data bit(s) to be sent.

D7	D6	D5	D4	D3	D2	D1	D0	Description
1	1	1	1	0	0	0	D	Sends one data bit
1	1	1	0	0	0	D	D	Sends two data bits
1	1	0	0	0	D	D	D	Sends three data bits
1	0	0	0	D	D	D	D	Sends four data bits
0	0	0	D	D	D	D	D	Sends five data bits

Bit 4 is the Send Break control bit.

When set, this bit forces the TxD output to send continuous "0s" beginning with the following transmit clock, regardless of any data being transmitted at the time. This bit functions whether or not the transmitter is enabled. When reset, TxD continues to send the contents of the Transmit Shift register, which might be syncs, data, or all "1s." If this bit is set while in the X21 mode (Monosync and Loop mode selected) and character synchronization is achieved in the receiver, this bit is automatically reset and the transmitter begins sending syncs or data. This bit can also be reset by a channel or hardware reset.

Bit 3 is Transmit Enable.

Data is not transmitted until this bit is set, and the TxD output sends continuous "1s" unless Auto Echo mode or SDLC Loop mode is selected. If this bit is reset after transmission started, the transmission of data or sync characters is completed. If the transmitter is disabled during the transmission of a CRC character, sync or flag characters are sent instead of CRC. This bit is reset by a channel or hardware reset.

Bit 2 is the SDLC/CRC-16 polynomial select bit.

This bit selects the CRC polynomial used by both the transmitter and receiver. When set, the CRC-16 polynomial is used; when reset, the SDLC polynomial is used. The SDLC/CRC polynomial must be selected when SDLC mode is selected. The CRC generator and checker can be preset to all "0s" or all "1s," depending on the state of the Preset 1/Preset 0 bit in WR10.

Bit 1 is the Request To Send control bit.

This is the control bit for the /RTS pin. When the RTS bit is set, the /RTS pin goes Low; when reset, /RTS goes High. When Auto Enable is set, the /RTS pin will immediately go Low when the RTS bit is set. However, when the RTS bit is reset, the /RTS pin remains Low until the transmitter is completely empty and the last stop bit has left the TxD pin. In synchronous modes or the Asynchronous mode with auto enables off, the pin directly follows the state of this bit. This bit is reset by a channel or hardware reset.

Bit 0 is Transmit CRC Enable.

This bit determines whether or not CRC is calculated on a transmit character. If this bit is set at the time the character is loaded from the transmit buffer to the Transmit Shift register, CRC is calculated on that character. CRC is not automatically sent unless this bit is set when the transmit underrun exists.

5.4.7 Write Register 6 (Sync Characters or SDLC Address Field)

WR6 is programmed to contain the transmit sync character in the Monosync mode, the first byte of a 16-bit sync character in the External Sync mode. WR6 is not used in asynchronous modes. In the SDLC modes, it is programmed to contain the secondary address field used to compare against the address field of the SDLC Frame. In SDLC mode, the ISCC does not automatically transmit the stations address at the beginning of a response frame. Bit positions for WR6 are shown in Figure 5-8.

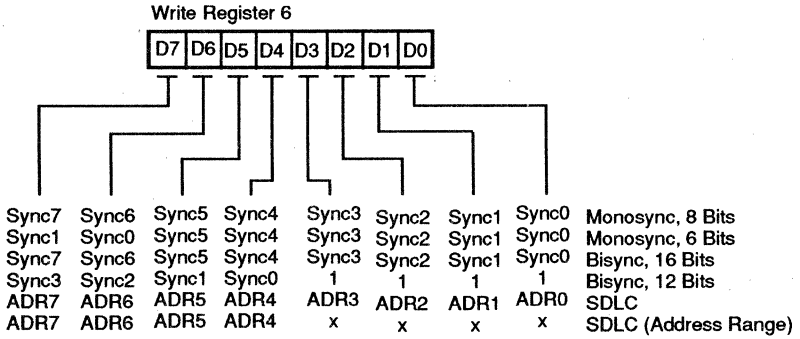


Figure 5-8. Write Register 6

5.4.8 Write Register 7 (SYNC Character or SDLC Flag)

WR7 is programmed to contain the receive sync character in the Monosync mode, a second byte (the last eight bits) of a 16-bit sync character in the Bisync mode, or a Flag character (01111110) in the SDLC modes. WR7 may hold the receive sync character or a flag if one of the special versions of the External Sync mode is selected. WR7 is not used in Asynchronous mode. Bit positions for WR7 are shown in Figure 5-9.

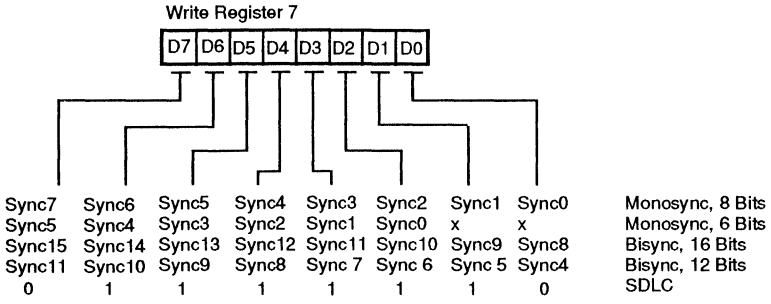


Figure 5-9. Write Register 7

5.4.9 Write Register 8 (Transmit Buffer)

WR8 is the transmit buffer register.

5.4.10 Write Register 9 (Master Interrupt Control)

WR9 is the Master Interrupt Control register and contains the Reset command bits. Only one WR9 exists in the ISCC and can be accessed from either channel. The Interrupt control bits can be programmed at the same time as the Reset command because these bits are only reset by a hardware reset. Bit positions for WR9 are shown in Figure 5-10.

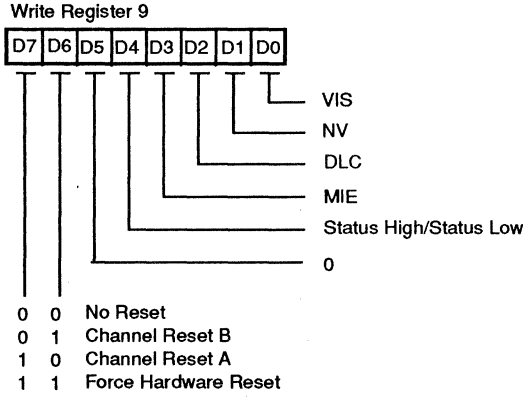


Figure 5-10. Write Register 9

Bit 7 and 6 are the Reset Command Bits.

Together, these bits select one of the reset commands for the SCC cell. Setting either of these bits to "1" disables both the receiver and the transmitter in the corresponding channel, forces TxD for that channel marking, forces the modem control signals High in that channel, resets all IPs and IUSs and disables all interrupts in that channel. Four extra PCLK cycles must be allowed beyond the usual cycle time after any of the active reset commands is issued before any additional commands or controls are written to the channel affected. In the non-multiplexed bus mode, four extra PCLK cycles must be allowed beyond the usual cycle time before any additional command or controls are written to the SCC cell.

Bit combination 00 is a Null Command. This command has no effect. It is used when a write to WR9 is necessary for some reason other than an SCC cell Reset command.

Bit combination 01 is the Channel Reset B Command. Issuing this command causes a channel reset to be performed on Channel B.

Bit combination 10 is the Channel Reset A Command. Issuing this command causes a channel reset to be performed on Channel A.

Bit combination 11 is the Force Hardware Reset Command. The effects of this command are identical to those of a hardware reset, except that the Shift Right/Shift Left bit is not changed and the MIE, Status High/Status Low and DLC bits take the programmed values that accompany this command.

Bit 5 is not used and must be programmed "0."

Bit 4 is the Status High//Status Low control bit.

This bit controls which vector bits the SCC cell will modify to indicate status. When set to "1," the SCC cell modifies bits V6, V5, and V4 according to Table 5-7. When set to "0," the SCC cell modifies bits V1, V2, and V3 according to Table 5-5. This bit controls status in both the vector returned during an interrupt acknowledge cycle and the status in RR2B. This bit is reset by a hardware reset.

Table 5-7. Interrupt Vector Modification

V3 V2 V1	Status High/Status Low =0
V4 V5 V6	Status High/Status Low =1
0 0 0	Ch B Transmit Buffer Empty
0 0 1	Ch B External/Status Change
0 1 0	Ch B Receive Char. Available
0 1 1	Ch B Special Receive Condition
1 0 0	Ch A Transmit Buffer Empty
1 0 1	Ch A External/Status Change
1 1 0	Ch A Receive Char. Available
1 1 1	Ch A Special Receive Condition

Bit 3 is the Master Interrupt Enable.

This bit is set to 1 to globally enable interrupts, and cleared to zero to disable interrupts. Clearing this bit to zero forces the IEO pin to follow the state of the IEI pin unless there is an IUS bit set in the SCC cell. No IUS bit can be set after the MIE bit is cleared to zero. This bit is reset by a hardware reset.

Bit 2 is the Disable Lower Chain control bit.

The Disable Lower Chain bit can be used by the CPU to control the interrupt daisy-chain. Setting this bit to "1" forces the IEO pin Low, preventing lower priority devices on the daisy-chain from requesting interrupts. This bit is reset by a hardware reset. **(Note that in the ISCC this will also prevent the DMA cell from requesting interrupts.)**

Bit 1 is the No Vector select bit.

The No Vector bit controls whether or or not the ISCC will respond to an interrupt acknowledge cycle by placing a vector on the data bus if the ISCC is the highest priority device requesting an interrupt. If this bit is set, no vector is returned; i.e. AD7-AD0 remain three-stated during an interrupt acknowledge cycle, even if the ISCC is the highest priority device requesting an interrupt.

Bit 0 is the Vector Includes Status control bit.

The Vector Includes Status Bit controls whether or not the SCC cell will include status information in the vector it places on the bus in response to an interrupt acknowledge cycle. If this bit is set, the vector returned is variable, with the variable field depending on the highest priority IP that is set. Table 5-5 shows the encoding of the status information. This bit is ignored if the No Vector (NV) bit is set.

5.4.11 Write Register 10 (Miscellaneous Transmitter/Receiver Control Bits)

WR10 contains miscellaneous control bits for both the receiver and the transmitter. Bit positions for WR10 are shown in Figure 5-11.

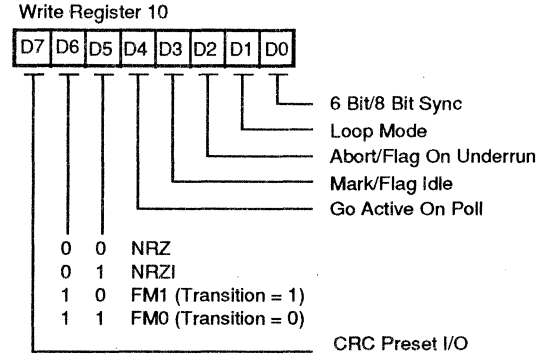


Figure 5-11. Write Register 10

Bit 7 is the CRC Presets 1//0 select bit.

This bit specifies the initialized condition of the receive CRC checker and the transmit CRC generator. If this bit is set to "1," the CRC generator and checker are preset to "1." If this bit is set to "0," the CRC generator and checker are preset to "0." Either option can be selected with either CRC polynomial. In SDLC mode, the transmitted CRC is inverted before transmission and the received CRC is checked against the bit pattern "0001110100001111." This bit is reset by a channel or hardware reset. This bit is ignored in Asynchronous mode.

Bits 6 and 5 are the Data Encoding select bits.

These bits control the coding method used for both the transmitter and the receiver, as illustrated in Table 5-8. All of the clocking options are available for all coding methods. The DPLL in the ISCC is useful for recovering clocking information in NRZI and FM modes. Any coding method can be used in X1 clock mode. A hardware reset forces NRZ mode. Timing for the various modes is shown in Figure 5-12.

Table 5-8. Data Encoding

Bit 6	Bit 5	Encoding
0	0	NRZ
0	1	NRZI
1	0	FM1 (transition = 1)
1	1	FM0 (transition = 0)

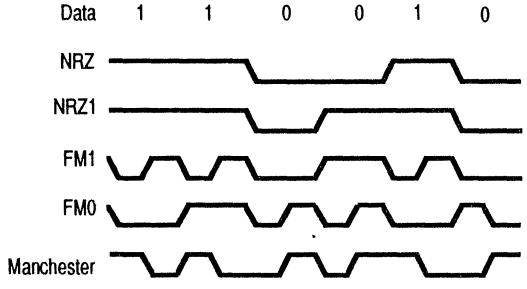


Figure 5-12. NRZ (NRZI), FM1 (FM0) Timing

Bit 4 is the Go Active On Poll control bit.

When Loop mode is first selected during SDLC operation, the ISCC connects RxD to TxD with only gate delays in the path. The ISCC does not go on-loop and insert the 1-bit delay between RxD and TxD until this bit has been set and an EOP received. When the ISCC is on-loop, the transmitter can not go active unless this bit is set at the time an EOP is received. The ISCC examines this bit whenever the transmitter is active in SDLC Loop mode and is sending a flag. If this bit is set at the time the flag is leaving the Transmit Shift register, another flag or data byte (if the transmit buffer is full) is transmitted. If the Go Active on Poll bit is not set at this time, the transmitter finishes sending the flag and reverts to the 1-Bit Delay mode. Thus, to transmit only one response frame, this bit should be reset after the first data byte is sent to the ISCC, but before CRC has been transmitted. If the bit is not reset before CRC is transmitted, extra flags are sent, slowing down response time on the loop. If this bit is reset before the first data is written, the ISCC completes the transmission of the present flag and reverts to the 1-Bit Delay mode. After gaining control of the loop, the ISCC is not able to transmit again until a flag and another EOP have been received. Though not strictly necessary, it is good practice to set this bit only upon receipt of a poll frame to ensure that the ISCC does not go on-loop without the CPU noticing it.

In synchronous modes other than SDLC with the Loop Mode bit set, this bit must be set before the transmitter can go active in response to a received sync character.

This bit is always ignored in Asynchronous mode and Synchronous modes unless the Loop Mode bit is set. This bit is reset by a channel or hardware reset.

Bit 3 is the Mark//Flag Idle line control bit.

This bit affects only SDLC operation and is used to control the idle line condition. If this bit is set to "0," the transmitter send flags as an idle line. If this bit is set to "1," the transmitter sends continuous "1s" after the closing flag of a frame. The idle line condition is selected byte by byte; i.e. either a flag or eight "1s" are transmitted. The primary station in an SDLC loop should be programmed for Mark Idle to create the EOP sequence. Mark Idle must be deselected at the beginning of a frame before the first data is written to the ISCC, so that an opening flag can be transmitted. This bit is ignored in Loop mode, but the programmed value takes effect upon exiting the Loop mode. This bit is reset by a channel or hardware reset.

Bit 2 is the Abort//Flag On Underrun select bit.

This bit affects only SDLC operation and is used to control how the ISCC responds to a transmit underrun condition. If this bit is set to "1" and a transmit underrun occurs, the ISCC sends an abort and a flag instead of CRC. If this bit is reset, the ISCC sends CRC on a transmit underrun. At the beginning of this 16-bit transmission, the Transmit Underrun/EOM bit is set, causing an External/Status interrupt. The CPU uses this status, along with the byte count from memory or the DMA, to determine whether the frame must be retransmitted. A transmit buffer Empty interrupt occurs at the end of this 16-bit transmission to start the next frame. If both this bit and the Mark/Flag Idle bit are set to "1," all "1s" are transmitted after the transmit underrun. This bit should be set after the first byte of data is sent to the ISCC and reset immediately after the last byte of data so that the frame will be terminated properly with CRC and a flag. This bit is ignored in Loop mode, but the programmed value is active upon exiting Loop mode. This bit is reset by a channel or hardware reset.

Bit 1 is the Loop Mode control bit.

In SDLC mode, the initial set condition of this bit forces the ISCC to connect TxD to RxD and to begin searching the incoming data stream so that it can go on loop. All bits pertinent to SDLC mode operation in other registers must be set before this mode is selected. The transmitter and receiver should not be enabled until after this mode has been selected. As soon as the Go Active On Poll bit is set and an EOP is received, the ISCC goes on-loop. If this bit is reset after the ISCC goes on-loop, the ISCC waits for the next EOP to go off-loop.

In synchronous modes, the ISCC uses this bit, along with the Go Active On Poll bit, to synchronize the transmitter to the receiver. The receiver should not be enabled until after this mode is selected. The TxD pin is held marking when this mode is selected unless a break condition is programmed. The receiver waits for a sync character to be received and then enables the transmitter on a character boundary. The break condition, if programmed, is removed. This mode works properly with sync characters of 6, 8, or 16 bits. This bit is ignored in Asynchronous mode and is reset by a channel or hardware reset.

Bit 0 is the 6 Bit//8 Bit SYNC select bit.

This bit is used to select a special case of synchronous modes. If this bit is set to "1" in Monosync mode, the receiver and transmitter sync characters are six bits long instead of the usual eight. If this bit is set to "1" in Bisync mode, the received sync will be 12 bits and the transmitter sync character will remain 16 bits long. This bit is ignored in SDLC and Asynchronous modes, but still has effect in the special external sync modes. This bit is reset by a channel or hardware reset.

5.4.12 Write Register 11 (Clock Mode Control)

WR11 is the Clock Mode Control register. The bits in this register control the sources of both the receive and transmit clocks, the type of signal on the /SYNC and /RTxC pins, and the direction of the /TRxC pin. Bit positions for WR11 are shown in Figure 5-13.

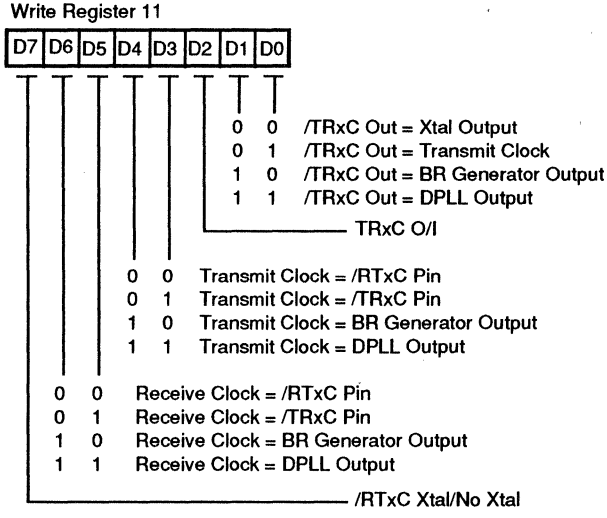


Figure 5-13. Write Register 11

Bit 7 is the RTxC-XTAL//NO XTAL select bit.

This bit controls the type of input signal the ISCC expects to see on the /RTxC pin. If this bit is set to "0," the ISCC expects a TTL-compatible signal as an input to this pin. If this bit is set to "1," the ISCC connects a high-gain amplifier between the /RTxC and /SYNC pins in expectation of a quartz crystal being placed across the pins.

The output of this oscillator is available for use as a clocking source. In this mode of operation, the /SYNC pin is unavailable for other use. The /SYNC signal is forced to "0" internally. A hardware reset forces /NO XTAL. (At least 20ms should be allowed after this bit is set to 1, to allow the oscillator to stabilize.)

Bits 6 and 5 are the Receiver Clock select bits 1 and 0.

These bits determine the source of the receive clock as shown in Table 5-9. They do not interfere with any of the modes of operation in the SCC cell, but simply control a multiplexer just before the internal receive clock input. A hardware reset forces the receive clock to come from the /RTxC pin.

Table 5-9. Receive Clock Source

Bit 6	Bit 5	Receive Clock
0	0	RTxC Pin
0	1	TRxC Pin
1	0	BR Output
1	1	DPLL Output

Bits 4 and 3 are the Transmit Clock select bits 1 and 0.

These bits determine the source of the transmit clock as shown in Table 5-10. They do not interfere with any of the modes of operation of the ISCC, but simply control a multiplexer just before the internal transmit clock input. The DPLL output that may be used to feed the transmitter in FM modes lags by 90 degrees the output of the DPLL used by the receiver. This makes the received and transmitted bit cells occur simultaneously, neglecting delays. A hardware reset selects the /TRxC pin as the source of the transmit clocks.

Table 5-10. Transmit Clock Source

Bit 4	Bit 3	Transmit Clock
0	0	RTxC Pin
0	1	TRxC Pin
1	0	BR Output
1	1	DPLL Output

Bit 2 is the TRxC Pin O//I control bit.

This bit determines the direction of the /TRxC pin. If this bit is set to "1," the TRxC pin is an output and carries the signal selected by D1 and D0 of this register. However, if either the receive or the transmit clock is programmed to come from the /TRxC pin, /TRxC will be an input, regardless of the state of this bit. The /TRxC pin is also an input if this bit is set to "0" A hardware reset forces this bit to "0."

Bits 1 and 0 are the /TRxC Output Source select bits 1 and 0.

These bits determine the signal to be echoed out of the ISCC via the /TRxC pin as given in Table 5-11. No signal is produced if /TRxC has been programmed as the source of either the receive or the transmit clock. If TRxC O//I (bit 2) is set to "0," these bits are ignored.

If the XTAL oscillator output is programmed to be echoed, and the Xtal oscillator has not been enabled, the /TRxC pin goes High. The DPLL signal that is echoed is the DPLL signal used by the receiver. Hardware reset selects the XTAL oscillator as the output source.

Table 5-11. Transmit External Control Selection

Bit 1	Bit 0	TRxC Pin Output
0	0	XTAL Oscillator Output
0	1	Transmit Clock
1	0	BR Output
1	1	DPLL Output (receive)

5.4.13 Write Register 12 (Lower Byte of Baud Rate Generator Time Constant)

WR12 contains the lower byte of the time constant for the baud rate generator. The time constant can be changed at any time, but the new value does not take effect until the next time the time constant is loaded into the down counter. No attempt is made to synchronize the loading of the time constant into WR12 and WR13 with the clock driving the down counter. For this reason, it is advisable to disable the baud rate generator while the new time constant is loaded into WR12 and WR13. Ordinarily, this is done anyway to prevent a load of the down counter between the writing of the upper and lower bytes of the time constant.

The formula for determining the appropriate time constant for a given baud is shown below with the desired rate in bits per second and the BR clock period in seconds. This formula is derived because the counter decrements from N down to "0"-plus-one-cycle for reloading the time constant and is then fed to a toggle flip-flop to make the output a square wave. Bit positions for WR12 are shown in Figure 5-14.

Time constant = [1/2 * desired rate * BR clock period] - 2

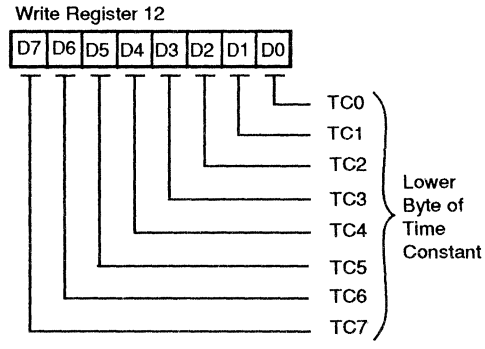


Figure 5-14. Write Register 12

5.4.14 Write Register 13 (Upper Byte of Baud Rate Generator Time Constant)

WR13 contains the upper byte of the time constant for the baud rate generator. Bit positions for WR13 are shown in Figure 5-15.

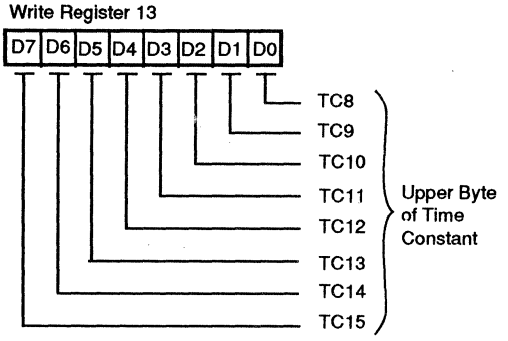


Figure 5-15. Write Register 13

5.4.15 Write Register 14 (Miscellaneous Control Bits)

WR14 contains some miscellaneous control bits. Bit positions for WR14 are shown in Figure 5-16.

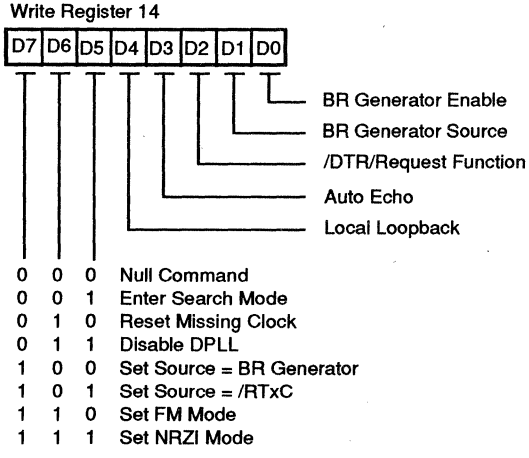


Figure 5-16. Write Register 14

Bit D7 and D5 are the Digital Phase-Locked Loop Command Bits.

These three bits encode the eight commands for the Digital Phase-Locked Loop. A channel or hardware reset disables the DPLL, resets the missing clock latches, sets the source to the /RTxC pin and selects NRZI mode. The Enter Search Mode command enables the DPLL after a reset.

Bit combination 000 is the Null Command. This command has no effect on the DPLL.

Bit combination 001 is the Enter Search Mode Command. Issuing this command causes the DPLL to enter the Search mode, where the DPLL searches for a locking edge in the incoming data stream. The action taken by the DPLL upon receipt of this command depends on the operating mode of the DPLL.

In NRZI mode, the output of the DPLL is High while the DPLL is waiting for an edge in the incoming data stream. After the Search mode is entered, the first edge the DPLL sees is assumed to be a valid data edge, and the DPLL begins the clock recovery operation from that point. The DPLL clock rate must be 32x the data rate in NRZI mode. Upon leaving the Search mode, the first sampling edge of the DPLL occurs 16 of these 32x clocks after the first data edge and the second sampling occurs 48 of these 32x clocks after the first data edge. Beyond this point, the DPLL begins normal operation, adjusting the output to remain in sync with the incoming data.

In FM mode, the output of the DPLL is Low while the DPLL is waiting for an edge in the incoming data stream. The first edge the DPLL detects is assumed to be a valid clock edge. for this to be the case, the line must contain only clock edges; i.e. with FM1 encoding, the line must be continuous "0s." With FM0 encoding the line must be continuous "1s," whereas Manchester encoding requires alternating "1s" and "0s" on the line. The DPLL clock rate must be 16 times the data rate in FM mode. The DPLL output causes the receiver to sample the data stream in the nominal center of the two halves of the bit cell to decide whether the data was a "1" or a "0." After this command is issued, as in NRZI mode, the DPLL starts sampling immediately after the first edge is detected. (In FM mode, the DPLL examines the clock edge of every other bit cell to decide what correction must be made to remain in sync.) If the DPLL does not see an edge during the expected window, the one clock missing bit in RR10 is set. If the DPLL does not see an edge after two successive attempts, the two clocks missing bit in RR10 is set and the DPLL automatically enters the Search mode. This command resets both clock missing latches.

Bit combination 010 is the Reset Clock Missing Command. Issuing this command disables the DPLL, resets the clock missing latches in RR10, and forces a continuous Search mode state.

Bit combination 001 is the Disable DPLL Command. Issuing this command disables the DPLL, resets the clock missing latches in RR10, and forces a continuous Search mode state.

Bit combination 100 is the Set Source = BR Gen Command. Issuing this command forces the clock for the DPLL to come from the output of the baud rate generator.

Bit combination 101 is the Set Source = /RTxC Command. Issuing the command forces the clock for the DPLL to come from the /RTxC pin or the crystal oscillator, depending on the state of the XTAL/NO XTAL bit in WR11. This mode is selected by a channel or hardware reset.

Bit combination 110 is the Set FM Mode Command. This command forces the DPLL to operate in the FM mode and is used to recover the clock from FM or Manchester-encoded data. (Manchester is decoded by placing the receiver in NRZ mode while the DPLL is in FM mode.)

Bit combination 111 is the Set NRZI Mode Command. Issuing this command forces the DPLL to operate in the NRZI mode. This mode is also selected by a hardware or channel reset.

Bit 4 is the Local Loopback select bit.

Setting this bit to "1" selects the Local Loopback mode of operation. In this mode, the internal transmitted data is routed back to the receiver, as well as to the TxD pin. The /CTS and /DCD inputs are ignored as enables in Local Loopback mode, even if auto enables is selected. (If so programmed, transitions on these inputs still cause interrupts.) This mode works with any Transmit/Receive mode except Loop mode. For meaningful results, the frequency of the transmit and receive clocks must be the same. This bit is reset by a channel or hardware reset.

Bit 3 is the Auto Echo select bit.

Setting this bit to "1" selects the Auto Echo mode of operation. In this mode, the TxD pin is connected to RxD, as in Local Loopback mode, but the receiver still listens to the RxD input. Transmitted data is never seen inside or outside the ISCC in this mode, and /CTS is ignored as a transmit enable. This bit is reset by a channel or hardware reset.

Bit 2 is the DTR/Request Function select bit.

This bit selects the function of the /DTR//REQ pin. If this is set to "0," the /DTR//REQ pin follows the state of the DTR bit in WR5. If this bit is set to "1," the /DTR//REQ pin goes Low whenever the transmit buffer becomes empty and in any of the synchronous modes when CRC has been sent at the end of a message. The /DTR//REQ does not go inactive until the internal operation satisfying the request is complete, which occurs three to four PCLK cycles after the falling edge of /DS, /READ or /WRITE. This bit is reset by a channel or hardware reset. Note that the /REQUEST function of this pin is not related to the operation of the ISCC DMA cell. Since a DMA function is present on this device, the /REQUEST function would not normally be used.

Bit 1 is the Baud Rate Generator Source select bit.

This bit selects the source of the clock for the baud rate generator. If this bit is set to "0." the baud rate generator clock comes from either the /RTxC pin or the XTAL oscillator (depending on the state of the XTAL/NO XTAL bit). If this bit is set to "1," the clock for the baud rate generator is the ISCC's PCLK input. Hardware reset sets this bit to "0," selecting the /RTxC pin as the clock source for the baud rate generator.

Bit 0 is the Baud Rate Generator Enable.

This bit controls the operation of the baud rate generator. The counter in the baud rate generator is enabled for counting when this bit is set to "1," and counting is inhibited when this bit is set to "0." When this bit is set to "1," change in the state of this bit is not reflected by the output of the baud rate generator for two counts of the counter. This allows the command to be synchronized. However, when set to "0," disabling is immediate. This bit is reset by a hardware reset.

5.4.16 Write Register 15 (External/Status Interrupt Control)

WR15 is the External/Status Source Control register. If the External/Status interrupts are enabled as a group via WR1, bits in this register control which External/Status conditions can cause an interrupt. Only the External/Status conditions that occur after the controlling bit are sent to "1" will cause an interrupt. This is true, even if an External/Status condition is pending at the time the bit is set. Bit positions for WR15 are shown in Figure 5-17.

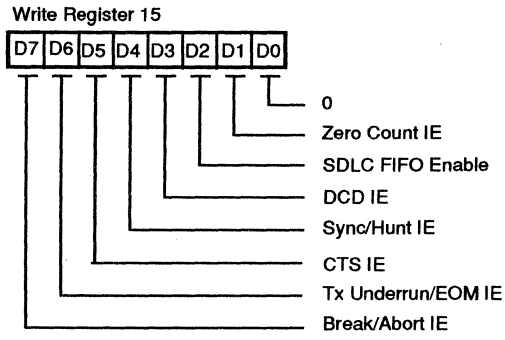


Figure 5-17. Write Register 15

Bit 7 is the Break/Abort Interrupt Enable.

If this bit is set to "1," a change in the Break/Abort status of the receiver causes an External/Status interrupt. This bit is set by a channel or hardware reset.

Bit 6 is the Transmit Underrun/EOM Interrupt Enable.

If this bit is set to "1," a change of state by the Tx Underrun/EOM latch in the transmitter causes an External/Status interrupt. This bit is set to "1" by a channel or hardware reset.

Bit 5 is the CTS Interrupt Enable.

If this bit is set to "1," a change of state on the /CTS pin causes an External/Status Interrupt. This bit is set by a channel or hardware reset.

Bit 4 is the SYNC/Hunt Interrupt Enable.

If this bit is set to "1," a change of state on the /SYNC pin causes an External/Status interrupt in Asynchronous mode, and a change of state in the Hunt bit in the receiver causes and External/Status interrupt in synchronous modes. This bit is set by a channel or hardware reset.

Bit 3 is the DCD Interrupt Enable.

If this bit is set to "1," a change of state on the /DCD pin causes an External/Status interrupt. This bit is set by a channel or hardware reset.

Bit 2 is not used and must be programmed "0."

Bit 1 is the Zero Count Interrupt Enable.

If this bit is set to "1," an External/Status interrupt is generated whenever the counter in the baud rate generator reaches "0." This bit is set to "0" by a channel or hardware reset.

Bit 0 is not used and must be programmed "0."

5.5 READ REGISTERS

The ISCC SCC cell contains seven read registers in each channel. In addition, there are two registers which are shared by both channels. The status of these registers is continually changing and depends on the mode of communication, received and transmitted data, and the manner in which this data is transferred to and from the CPU. The following description details the bit assignments for each register.

5.5.1 Read Register 0 (Transmit/receive buffer Status and External Status)

Read Register 0 contains the status of the receive and transmit buffers. RR0 also contains the status bits for the six sources of External/Status interrupts. The bit configuration is illustrated in Figure 5-18.

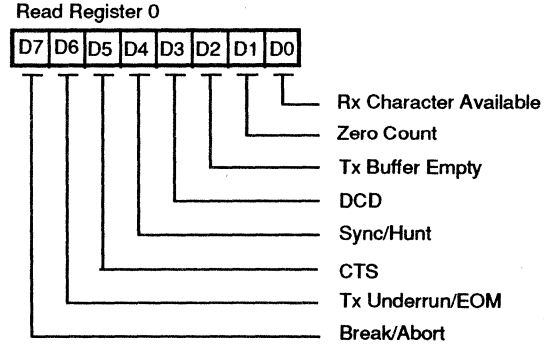


Figure 5-18. Read Register 0

Bit 7 is the Break/Abort status.

In the Asynchronous mode, this bit is set when a Break sequence (null character plus framing error) is detected in the receive data stream. This bit is reset when the sequence is terminated, leaving a single null character in the receive FIFO. This character should be read and discarded. In SDLC mode, this bit is set by the detection of an Abort sequence (seven or more "1s"), then reset automatically at the termination of the Abort sequence. In either case, if the Break/Abort IE bit is set, an External/Status interrupt is initiated. Unlike the remainder of the External/Status bits, both transitions are guaranteed to cause an External/Status interrupt, even if another External/Status interrupt is pending at the time these transitions occur. This procedure is necessary because Abort or Break conditions may not persist.

Bit 6 is the Transmit Underrun/EOM status.

This bit is set by a channel or hardware reset and when the transmitter is disabled or a Send Abort command is issued. This bit can only be reset by the reset Tx Underrun/EOM Latch command in WR0. When the Transmit Underrun occurs, this bit is set and causes an External/Status interrupt (if the Tx Underrun/EOM IE bit is set).

Only the 0-to-1 transition of this bit causes an interrupt. This bit is always "1" in Asynchronous mode, unless a reset Tx Underrun/EOM Latch command has been erroneously issued. In this case, the Send Abort command can be used to set the bit to one and at the same time cause an External/Status interrupt.

Bit 5 is the Clear to Send pin status.

If the CTS IE bit in WR15 is set, this bit indicates the state of the /CTS pin while no interrupt is pending latches the state of the /CTS pin and generates an External/Status interrupt. Any odd number of transitions on the /CTS pin, while another External/Status interrupt is pending, also causes an External/Status interrupt condition. If the CTS IE bit is reset, it merely reports the current unlatched state of the /CTS pin.

Bit 4 is the SYNC/Hunt status.

The operation of this bit is similar to that of the CTS bit, except that the condition monitored by the bit varies depending on the mode in which the ISCC is operating.

When the XTAL oscillator option is selected in asynchronous modes, this bit is forced to "0" (no External/Status interrupt is generated). Selecting the XTAL oscillator in synchronous or SDLC modes had no effect on the operation of this bit.

The XTAL oscillator should not be selected in External Sync mode.

In Asynchronous mode, the operation of this bit is identical to that of the CTS status bit, except that this bit reports the state of the /SYNC pin.

In External sync mode the /SYNC pin is used by external logic to signal character synchronization. When the Enter Hunt Mode command is issued in External Sync mode, the /SYNC pin must be held High by the external sync logic until character synchronization is achieved. A High on the /SYNC pin holds the Sync/Hunt bit in the reset condition.

When external synchronization is achieved, /SYNC must be driven Low on the second rising edge of the Receive Clock after the last rising edge of the Receive Clock on which the last bit of the receive character was received. Once /SYNC is forced Low, it is good practice to keep it Low until the CPU informs the external sync logic that synchronization has been lost or that a new message is about to start. Both transitions on the /SYNC pin cause External/Status interrupts if the Sync/Hunt IE bit is set to "1".

The Enter Hunt Mode command should be issued whenever character synchronization is lost. At the same time, the CPU should inform the external logic that character synchronization has been lost and that the ISCC is waiting for /SYNC to become active.

In the Monosync and Bisync Receive modes, the Sync/Hunt status bit is initially set to "1" by the Enter Hunt Mode command. The Sync/Hunt bit is reset when the ISCC established character synchronization. Both transitions cause External/Status interrupts if the Sync/Hunt IE bit is set. When the CPU detects the end of message or the loss of character synchronization, the Enter Hunt Mode command should be issued to set the Sync/Hunt bit and cause an External/Status interrupt. In this mode, the SYNC pin is an output, which goes Low every time a sync pattern is detected in the data stream.

In the SDLC modes, the Sync/Hunt bit is initially set by the Enter Hunt Mode command or when the receiver is disabled. It is reset when the opening flag of the first frame is detected by the ISCC. An External/Status interrupt is also generated if the Sync/Hunt IE bit is set. Unlike the Monosync and Bisync modes, once the Sync/Hunt bit is reset in SDLC mode, it does not need to be set when the end of the frame is detected. The ISCC automatically maintains synchronization. The only way the Sync/Hunt bit can be set again is by the Enter Hunt Mode command or by disabling the receiver.

Bit 3 is the Data Carrier Detect status.

If the DCD IE bit in WR15 is set, this bit indicates the state of the DCD pin the last time the Enabled External/Status bits changed. Any transition on the DCD pin while no interrupt is pending latches the state of the DCD pin and generates an External/Status interrupt. Any odd number of transitions on the DCD pin while another External/Status interrupt is pending will also cause an External/Status interrupt condition. If the DCD IE is reset, this bit merely reports the current, unlatched state of the DCD pin.

Bit 2 is the TX Buffer Empty status.

This bit is set to "1" when the transmit buffer is empty. It is reset while CRC is sent in a synchronous or SDLC mode and while the transmit buffer is full. The bit is reset when a character is loaded into the transmit buffer. This bit is always in the set condition after a hardware or channel reset.

Bit 1 is the Zero Count status.

If the Zero Count interrupt Enable bit is set in WR15, this bit is set to one while the counter in the baud rate generator is at the count of zero. If there is no other External/Status interrupt condition pending at the time this bit is set, an External/Status interrupt is generated. However, if there is another External/Status interrupt pending at this time, no interrupt is initiated until interrupt service is complete. If the Zero Count condition does not persist beyond the end of the interrupt service routine, no interrupt will be generated. This bit is not latched High, even thought the other External/Status latches close as a result of the Low-to-High transition on Zero Count. The interrupt routine should check the other External/Status conditions for changes. If none changed, Zero Count was the source. In polled applications, check the IP bit in RR3A for a status change and then proceed as in the interrupt service routine.

Bit 0 is Receive Character Available.

This bit is set to "1" when at least one character is available in the receive FIFO and is reset when the receive FIFO is completely empty. A channel or hardware reset empties the receive FIFO.

5.5.2 Read Register 1

RR1 contains the Special Receive Condition status bits and the residue codes for the I-field in SDLC mode. Figure 5-19 shows the bit positions for RR1.

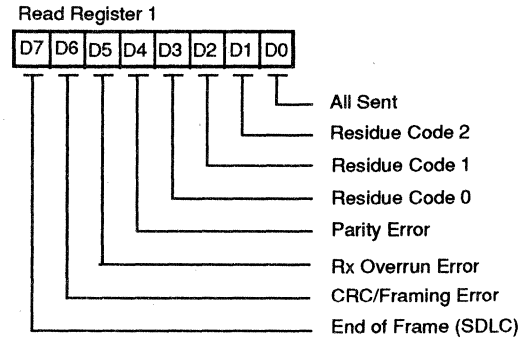


Figure 5-19. Read Register 1

Bit 7 is the End of Frame (SDLC) status.

This bit is used only in SDLC mode and indicates that a valid closing flag has been received and that the CRC Error bit and residue codes are valid. This bit can be reset by issuing the Error Reset command. It is also updated by the first character of the following frame. This bit is reset in any mode other than SDLC.

Bit 6 is the CRC/Framing Error status.

If a framing error occurs (in Asynchronous mode), this bit is set (and not latched) for the receive character in which the framing error occurred. Detection of a framing error adds an additional one-half bit to the character time so that the framing error is not interpreted as a new Start bit. In Synchronous and SDLC modes, this bit indicates the result of comparing the CRC checker to the appropriate check value. This bit is reset by issuing an Error Reset command, but the bit is never latched. Therefore, it is always updated when the next character is received. When used for CRC error status in Synchronous or SDLC modes, this bit is usually set since most bit combinations, except for a correctly completed message, result in a non-zero CRC.

Bit 5 is the Receiver Overrun Error status.

This bit indicates that the receive FIFO has overflowed. Only the character that has been written over is flagged with this error, and when the character is read, the Error condition is latched until reset by the Error Reset command. The overrun character and all subsequent characters received until the Error Reset command is issued causes a Special Receive Condition vector to be returned.

Bit 4 is the Parity Error status.

When parity is enabled, this bit is set for the characters whose parity does not match the programmed sense (even/odd). This bit is latched so that once an error occurs, it remains set until the Error Reset command is issued. If the parity in Special Condition bit is set, a parity error causes a Special Receive Condition vector to be returned on the character containing the error and on all subsequent characters until the Error Reset command is issued.

Bits 3, 2, and 1 are the Residue Codes, bits 2, 1, and 0.

In those cases in SDLC mode where the received I-Field is not an integral multiple of the character length, these three bits indicate the length of the I-Field and are meaningful only for the transfer in which the end of frame bit is set. This field is set to "011" by a channel or hardware reset and is forced to this state in Asynchronous mode. These three bits can leave this state only if SDLC is selected and a character is received. The codes signify the following (Reference Table 5-12) when a receive character length is eight bits per character.

I-Field bits are right-justified in all cases. If a receive character length other than eight bits is used for the I-Field, a table similar to Table 5-12 can be constructed for each different character length. Table 5-13 shows the residue codes for no residue (The I-Field boundary lies on a character boundary).

Table 5-12. I-Field Bit Selection (8 Bits Only)

Bit 3	Bit 2	Bit 1	I-Field Bits in Last Byte	I-Field Bits in Previous Byte
1	0	0	0	3
0	1	0	0	4
1	1	0	0	5
0	0	1	0	6
1	0	1	0	7
0	1	1	0	8
1	1	1	1	8
0	0	0	2	8

Table 5-13. Bits per Character Residue Decoding

Bits per Character	Bit 3	Bit 2	Bit 1
8	0	1	1
7	0	0	0
6	0	1	0
5	0	0	1

Bit 0 is the All Sent status.

In Asynchronous mode, this bit is set when all characters have completely cleared the transmitter pins. Most modems contain additional delays in the data path, which requires the modem control signals to remain active until after the data has cleared both the transmitter and the modem. This bit is always set in synchronous and SDLC modes.

5.5.3 Read Register 2

RR2 contains the interrupt vector written into WR2. When the register is accessed in Channel A, the vector returned is the vector actually stored in WR2. When this register is accessed in Channel B, the vector returned includes status information in bits 1, 2 and 3 or in bits 6, 5 and 4, depending on the state of the Status High/Status Low bit in WR9 and independent of the state of the VIS bit in WR9. The vector is modified according to Table 5-7 shown in the explanation of the VIS bit in WR9. If no interrupts are pending the status is V3,V2,V1 -011, or V6,V5,V4-110. Figure 5-20 shows the bit positions for RR2.

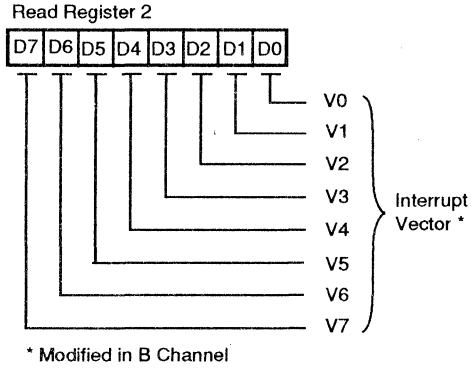


Figure 5-20. Read Register 2

5.5.4 Read Register 3

RR3 is the Interrupt Pending register. The status of each of the interrupt Pending bits in the SCC cell is reported in this register. This register exists only in Channel A. If this register is accessed in Channel B, all "0's" are returned. The two unused bits are always returned as "0". Figure 5-21 shows the bit positions for RR3.

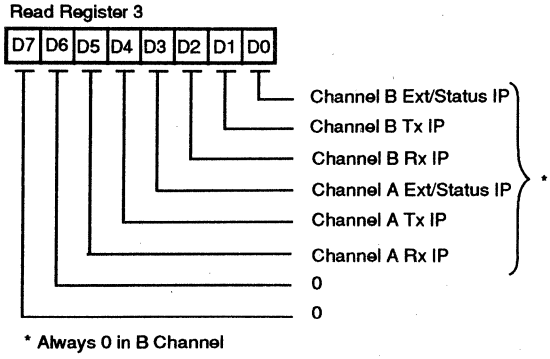


Figure 5-21. Read Register 3

5.5.5 Read Register 8

RR8 is the Receive Data register.

5.5.6 Read Register 10

RR10 contains some miscellaneous status bits. Unused bits are always "0". Bit position for RR10 are shown in Figure 5-22.

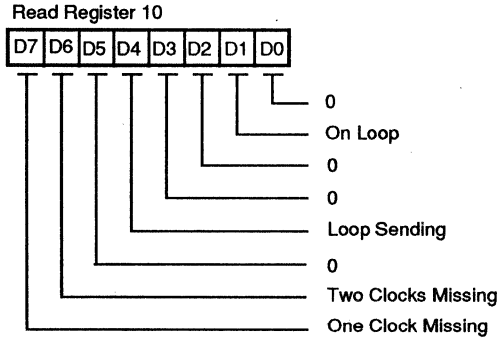


Figure 5-22. Read Register 10

Bit 7 is the One Clock Missing status.

While operating in the FM mode, the DPLL sets this bit to "1" when it does not see a clock edge on the incoming lines in the window where it expects one. This bit is latched until reset by a Reset Missing Clock or Enter Search Mode command in WR14. In the NRZI mode of operation and while the DPLL is disabled, this bit is always "0".

Bit 6 is the Two Clocks Missing status.

While operating in the FM mode, the DPLL sets this bit to "1" when it does not see a clock edge in two successive tries. At the same time the DPLL enters the Search mode. This bit is latched until reset by a Reset Missing Clock or Enter Search Mode command in WR14, bit 5-7. In the NRZI mode of operation and while the DPLL is disabled, this bit is always "0".

Bit 4 is the Loop Sending status.

This bit is set to "1" in SDLC Loop mode while the transmitter is in control of the Loop, that is, while the ISCC is actively transmitting on the loop. This bit is reset at all other times.

This bit can be polled in SDLC mode to determine when the closing flag has been sent.

Bit 1 is the On Loop status.

This bit is set to "1" while the ISCC is actually on loop in SDLC Loop mode. This bit is set to "1" in the X21 mode (Loop mode selected while in monosync) when the transmitter goes active. This bit is "0" at all other times. This bit can also be polled in SDLC mode to determine when the closing flag has been sent.

5.5.7 Read Register 12

RR12 returns the value stored in WR12, the lower byte of the time constant for the baud rate generator. Figure 5-23 shows the bit positions for RR12.

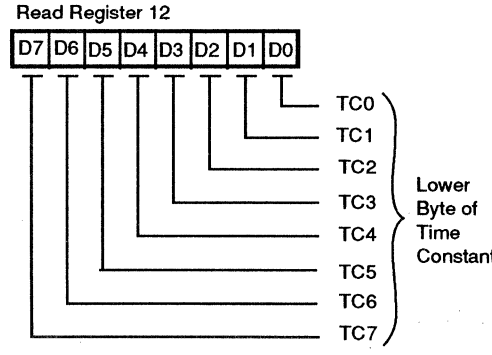


Figure 5-23. Read Register 12

5.5.8 Read Register 13

RR13 returns the value stored in WR13, the upper byte of the time constant for the baud rate generator. Figure 5-24 shows the bit positions for RR13.

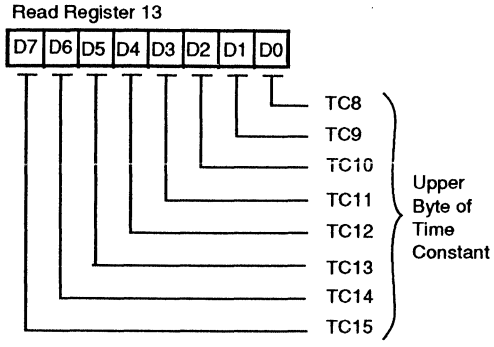


Figure 5-24. Read Register 13

5.5.9 Read Register 15

RR15 reflects the value stored in WR15, the External/Status IE bits. The two unused bits are always returned as "0s". Figure 5-25 shows the bits positions for RR15.

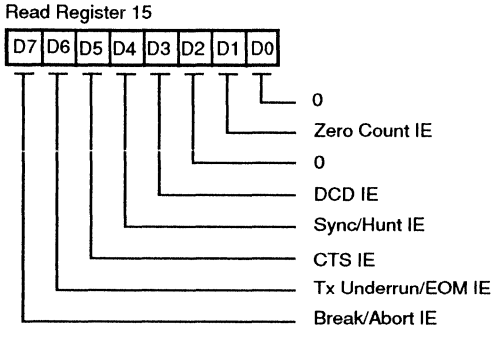


Figure 5-25. Read Register 15

5.6 DMA CELL REGISTER DESCRIPTIONS

5.6.1 Channel Command/Address Register

This register is a write only register and is at the same address as the DMA Status Register. Figure 5-26 shows the bit positions for this register.

Bits 7 through 5 are encoded with the commands for the DMA as shown below:

Bit combination 000 is a Null command and has no affect on the DMA.

Bit combination 001 is reserved.

Bit combination 010 is the DMA Reset Highest IUS command. This command resets only the highest priority IUS bit that is set in the DMA cell and occurs independent of the state of the IEI for the ISCC.

Bit combination 011 is the Reset command and is used to reset the DMA cell. All of the DMA channels are reset. The DMA channels remain reset until enabled.

Bit combination 100 is the command to enable the Transmitter B channel DMA. The DMA operation is not triggered by this command.

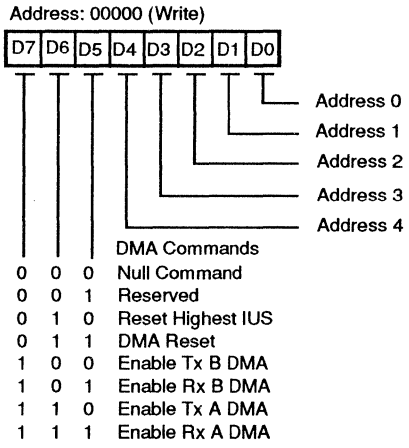


Figure 5-26. Channel Command/Address Register

Bit combination 101 is the command to enable the Receiver B channel DMA. The DMA operation is not triggered by this command.

Bit combination 110 is the command to enable the Transmitter A channel DMA. The DMA operation is not triggered by this command.

Bit combination 111 is the command to enable the Receiver B channel DMA. The DMA operation is not triggered by this command.

Bits 4 through 0 comprise the pointer to the internal registers. This pointer is used in the non-multiplexed bus modes to access the DMA cell internal registers. After reset, the internal pointer points to the Channel Command / Address Register. Access to other registers is accomplished by first writing the address of the desired register to this field. The next access to the DMA cell will be to the register so addressed; this access may be a read or a write. After this second access (the access to the desired register), the internal pointer latch is cleared and the pointer again points to the Channel Command / Address Register.

5.6.2 DMA Status Register

This register is a read only register and is at the same address as the Channel Command / Address Register. The individual bits indicate abort and terminal count of each of the four DMA channels. Figure 5-27 shows the bit positions for the DMA Status Register. The status in this register is automatically cleared after a read.

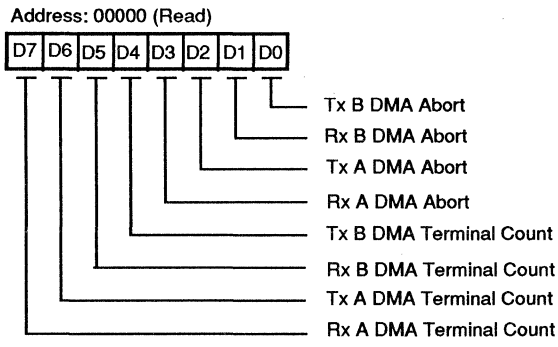


Figure 5-27. DMA Status Register

Bit 7, when set, indicates that the Receiver A DMA has reached terminal count.

Bit 6, when set, indicates that the Transmitter A DMA has reached terminal count.

Bit 5, when set, indicates that the Receiver B DMA has reached terminal count.

Bit 4, when set, indicates that the Transmitter B DMA has reached terminal count.

Bit 3, when set, indicates that the Receiver A DMA operation has been aborted.

Bit 2, when set, indicates that the Transmitter A DMA operation has been aborted.

Bit 1, when set, indicates that the Receiver B DMA operation has been aborted.

Bit 0, when set, indicates that the Transmitter B DMA operation has been aborted.

5.6.3 Interrupt Control Register

The Interrupt Control Register is used to enable the interrupts from the individual sources, and select the interrupt vector options. This register is read / write. The bit positions are shown in Figure 5-28.

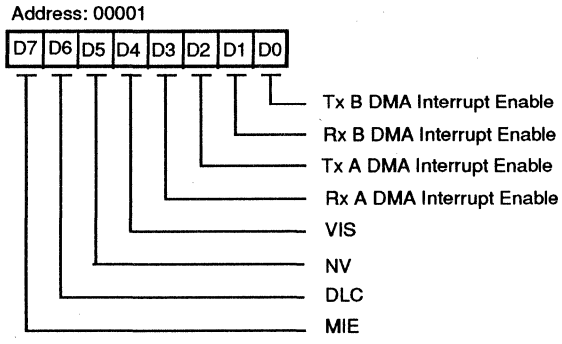


Figure 5-28. Interrupt Control Register

Bit 7 is the Master Interrupt Enable (MIE). when this bit is cleared, all interrupts from the DMA cell are disabled even though the individual enable bits are set. This bit must be set for any DMA interrupt source to cause an interrupt.

Bit 6 is the disable lower chain control bit (DLC). If this bit is set, the external lower chain of the daisy chained interrupt structure is disabled; IEO will not become active.

Bit 5 selects the no vector option. With this bit set, the DMA cell does not return an interrupt vector to the CPU. During the interrupt acknowledge cycle when the interrupt vector

is requested, the ISCC will not drive the bus. With this bit clear, an interrupt vector will be returned in the interrupt acknowledge cycle.

Bit 4 selects the vector include status option for the interrupt vector from the DMA cell. With this bit clear, a DMA interrupt vector will be returned which is the vector that has been programmed into the Interrupt Vector Register. With this bit set, the returned vector contains status information concerning the interrupt source. The status returned reflects the highest priority interrupt pending (IP bit is set and the corresponding Interrupt Enable bit is set). This status information is contained in bits 1, 2, and 3 of the interrupt vector. The other interrupt vector bits remain unmodified.

Table 5-14. Interrupt Vector Status Encoding

IV3	IV2	IV1	Interrupt
0	0	0	No Interrupt Pending
0	0	1	Not Possible
0	1	0	Not Possible
0	1	1	Not Possible
1	0	0	Rx A IP
1	0	1	Rx B IP
1	1	0	Tx A IP
1	1	1	Tx B IP

Bit 3, when set, enables the interrupt from the Receiver A DMA.

Bit 2, when set, enables the interrupt from the Transmitter A DMA.

Bit 1, when set, enables the interrupt from the Receiver B DMA.

Bit 0, when set, enables the interrupt from the Transmitter B DMA.

5.6.4 Interrupt Vector Register

This register holds the interrupt vector for the DMA cell. The value programmed into this register is returned during the interrupt response cycle as the interrupt vector when one of the DMA interrupt sources is the highest priority pending interrupt. Note that bits 1, 2, and 3 may be replaced by interrupt status information if the Vector Include Status option has been selected (see Interrupt Control Register). The bit positions are shown in Figure 5-29.

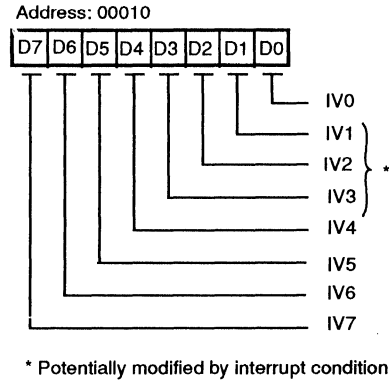


Figure 5-29. Interrupt Vector Register

5.6.5 Interrupt Command Register

This is a write only register and is used to command the DMA cell. It shares its address with the Interrupt Status Register. The bit positions for the Interrupt Command Register are shown in Figure 5-30.

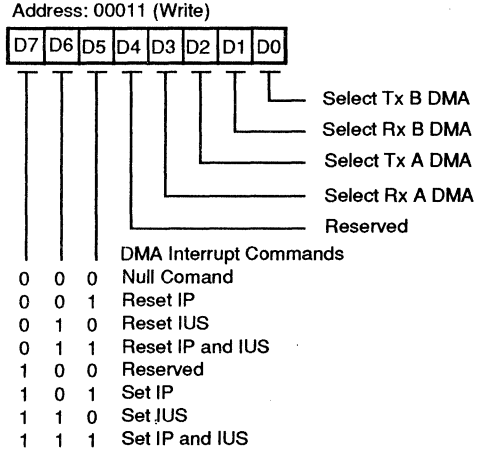


Figure 5-30. Interrupt Command Register

Bits 7 through 5 are encoded with the commands for the DMA cell as shown below:

Bit combination 000 is a Null command and has no affect on the DMA.

Bit combination 001 resets the Interrupt Pending (IP) bit in the selected DMA channel(s).

Bit combination 010 resets the Interrupt Under Service (IUS) bit in the selected DMA channel(s).

Bit combination 011 resets both the Interrupt Pending (IP) bit and the Interrupt Under Service (IUS) bit in the selected DMA channel(s).

Bit combination 100 is Reserved.

Bit combination 101 sets the Interrupt Pending (IP) bit in the selected DMA channel(s).

Bit combination 110 sets the Interrupt Under Service (IUS) bit in the selected DMA channel(s).

Bit combination 111 sets both the Interrupt Pending (IP) bit and the Interrupt Under Service (IUS) bit in the selected DMA channel(s).

Bit 4 is Reserved. (This bit should be programmed as a zero to avoid conflicts with future versions of this device.)

Bits 3 through 0 select the channel to which the command is to apply. More than one of these bits may be set for the command; the command is applied to all of the DMA channels whose bits are set in this field: (These bits are not stored and must be written with each command.)

Bit 3, when set, applies the command to the Receive A DMA.

Bit 2, when set, applies the command to the Transmit A DMA.

Bit 1, when set, applies the command to the Receive B DMA.

Bit 0, when set, applies the command to the Transmit B DMA.

5.6.6 Interrupt Status Register

This is a read only register which shares its address with the Interrupt Command Register. The bits in this register reflect the status of the Interrupt Pending (IP) and Interrupt Under Service (IUS) bits in the DMA channels. The bit positions for this register are shown in Figure 5-31.

Bit 7 reflects the Receive A DMA Interrupt Under Service status. This bit can be set or cleared through a command (see Interrupt Command Register). This bit is set to 1 automatically during an interrupt acknowledge if this is the highest priority interrupt pending. This is the highest priority pending interrupt if the corresponding Interrupt Pending bit is set to 1, if the Interrupt Enable bit for this interrupt is set to 1, if the IEI input to the ISCC is 1, if the DMA cell Master Interrupt Enable bit is set to 1, if there are no SCC cell interrupts pending, and if there is no other DMA channel with an interrupt pending that is at a higher priority level (see DMA Control Register for priority programming).

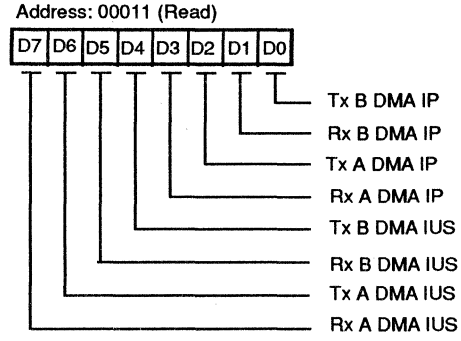


Figure 5-31. Interrupt Status Register

Bit 6 reflects the Transmit A DMA Interrupt Under Service status. The function of this bit is identical to that for bit 7.

Bit 5 reflects the Receive B DMA Interrupt Under Service status. The function of this bit is identical to that for bit 7.

Bit 4 reflects the Transmit B DMA Interrupt Under Service status. The function of this bit is identical to that for bit 7.

Bit 3 reflects the Receive A DMA Interrupt Pending (IP) status. This bit can be set or cleared through a command (see Interrupt Command Register). This bit will be set to 1 automatically when a Receive A DMA interrupt condition occurs. An interrupt will be requested if the corresponding Interrupt Enable bit is set to 1, if the DMA Master Interrupt Enable bit is set to 1, and if the ISCC IEI input is 1, and if the corresponding IUS bit is 0.

Bit 2 reflects the Transmit A DMA Interrupt Pending status. The function of this bit is identical to that for bit 3.

Bit 1 reflects the Receive B DMA Interrupt Pending status. The function of this bit is identical to that for bit 3.

Bit 0 reflects the Transmit B DMA Interrupt Pending status. The function of this bit is identical to that for bit 3.

5.6.7 DMA Enable Register

This register controls the enabling of the DMA channels and contains the enables for the DMA Abort Interrupt conditions. The bit positions for this register are shown in Figure 5-32.

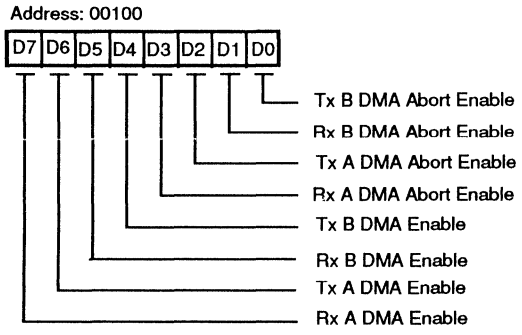


Figure 5-32. DMA Enable Register

Bit 7, when set to 1, enables the Receive A DMA.

Bit 6, when set to 1, enables the Transmit A DMA.

Bit 5, when set to 1, enables the Receive B DMA.

Bit 4, when set to 1, enables the Transmit B DMA.

Bit 3, when set to 1, enables the interrupt in the Receive A DMA Channel that is generated when a DMA operation in this channel is aborted.

Bit 2, when set to 1, enables the interrupt in the Transmit A DMA Channel that is generated when a DMA operation in this channel is aborted.

Bit 1, when set to 1, enables the interrupt in the Receive B DMA Channel that is generated when a DMA operation in this channel is aborted.

Bit 0, when set to 1, enables the interrupt in the Transmit B DMA Channel that is generated when a DMA operation in this channel is aborted.

5.6.8 DMA Control Register

This register controls DMA priorities, requests, and address generation. The bit positions for this register are shown in Figure 5-33.

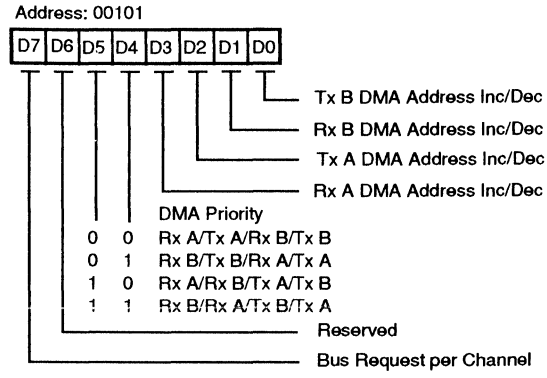


Figure 5-33. DMA Control Register

Bit 7, when set to 1, enables a bus request per channel. This means that if more than one DMA request is pending, after the completion of a DMA transfer from one DMA channel, the bus will be relinquished and susequently requested for the other channel DMA requests. If this bit is cleared (0), the DMA will hold the bus until there are no DMA requests pending, thus multiple channels may make DMA transfers without separate, intervening bus acquisitions.

Bit 6 is reserved and should be programmed zero.

Bits 5 and 4 control the DMA priority according to Table 5-15. If DMA requests arise simultaneously, the channel which is serviced first is the one with the highest priority as programmed. Note that the interrupt priorities are not affected by this programming and remain fixed in the order Rx A DMA (highest), Tx A DMA, Rx B DMA, Tx B DMA (lowest).

Table 5-15. DMA Priority

D5	D4	DMA Priority
0	0	Rx A / Tx A / Rx B / Tx B
0	1	Rx B / Tx B / Rx A / Tx A
1	0	Rx A / Rx B / Tx A / Tx B
1	1	Rx B / Rx A / Tx B / Tx A

Bit 3 selects if the DMA address for the Receive A DMA is to be incremented or decremented after each DMA byte transfer. Programming this bit to a 1 causes the address to increment; programming this bit to a 0 causes the address to decrement.

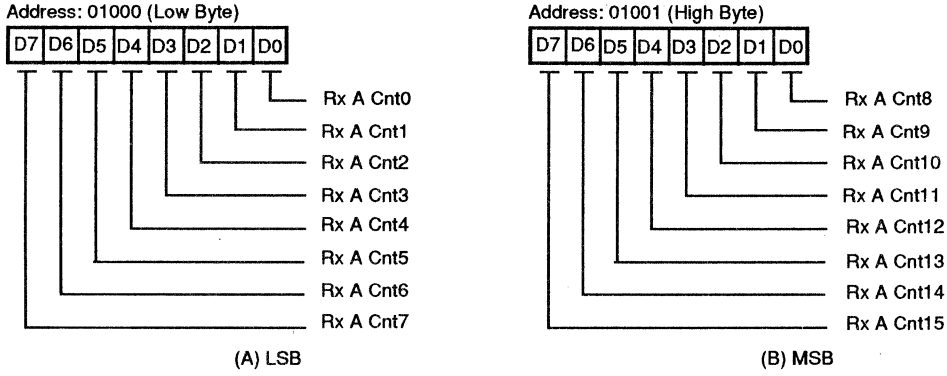
Bit 2 selects if the DMA address for the Transmit A DMA is to be incremented or decremented after each DMA byte transfer. Its operation is identical to bit 3.

Bit 1 selects if the DMA address for the Receive B DMA is to be incremented or decremented after each DMA byte transfer. Its operation is identical to bit 3.

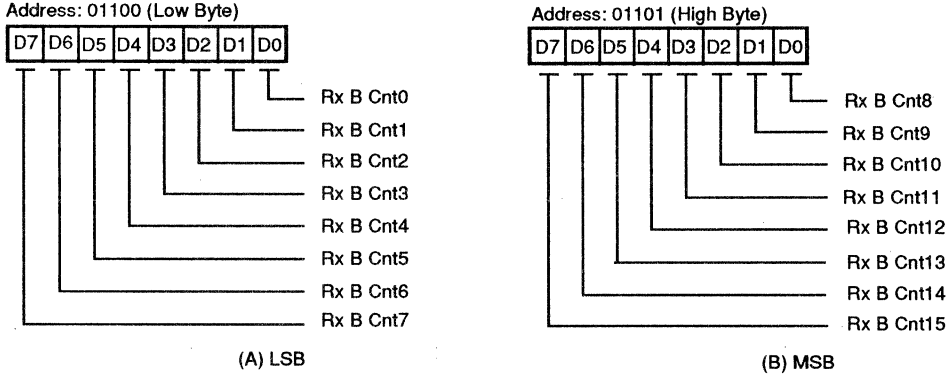
Bit 0 selects if the DMA address for the Transmit B DMA is to be incremented or decremented after each DMA byte transfer. Its operation is identical to bit 3.

5.6.9 Receive DMA Count Registers A, B

There are two sets of Receive DMA Count Registers, one set for Receive DMA Channel A and one set for Receive DMA Channel B. Each register set contains two registers, one for the low byte (bits 7 - 0) and one for the high byte (bits 15 - 8) as shown in Figure 5-34. These registers are read / write.



Receive DMA Count Register Channel A

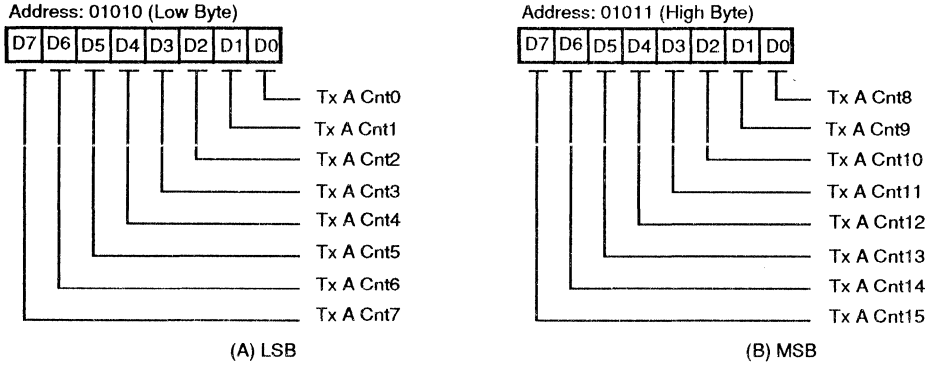


Receive DMA Count Register Channel B

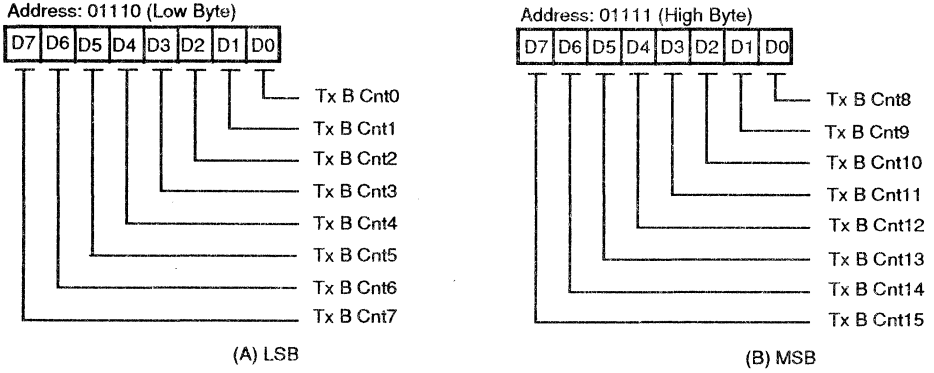
Figure 5-34. Receive DMA Count Registers

5.6.10 Transmit DMA Count Registers A, B

There are two sets of Transmit DMA Count Registers, one set for Transmit DMA Channel A and one set for Transmit DMA Channel B. Each register set contains two registers, one for the low byte (bits 7 - 0) and one for the high byte (bits 15 - 8) as shown in Figure 5-35. These registers are read / write.



Transmit DMA Count Register Channel A



Transmit DMA Count Register Channel B

Figure 5-35. Transmit DMA Count Registers

5.6.11 Receive DMA Address Registers A, B

There are two sets of Receive DMA Address Registers, one set for Receive DMA Channel A and one set for Receive DMA Channel B. Each set consists of four registers, one for address bits 7 - 0, one for address bits 15 - 8, one for address bits 23 - 16, and one for address bits 31 - 24 as shown in Figure 5-36. These registers are read/write.

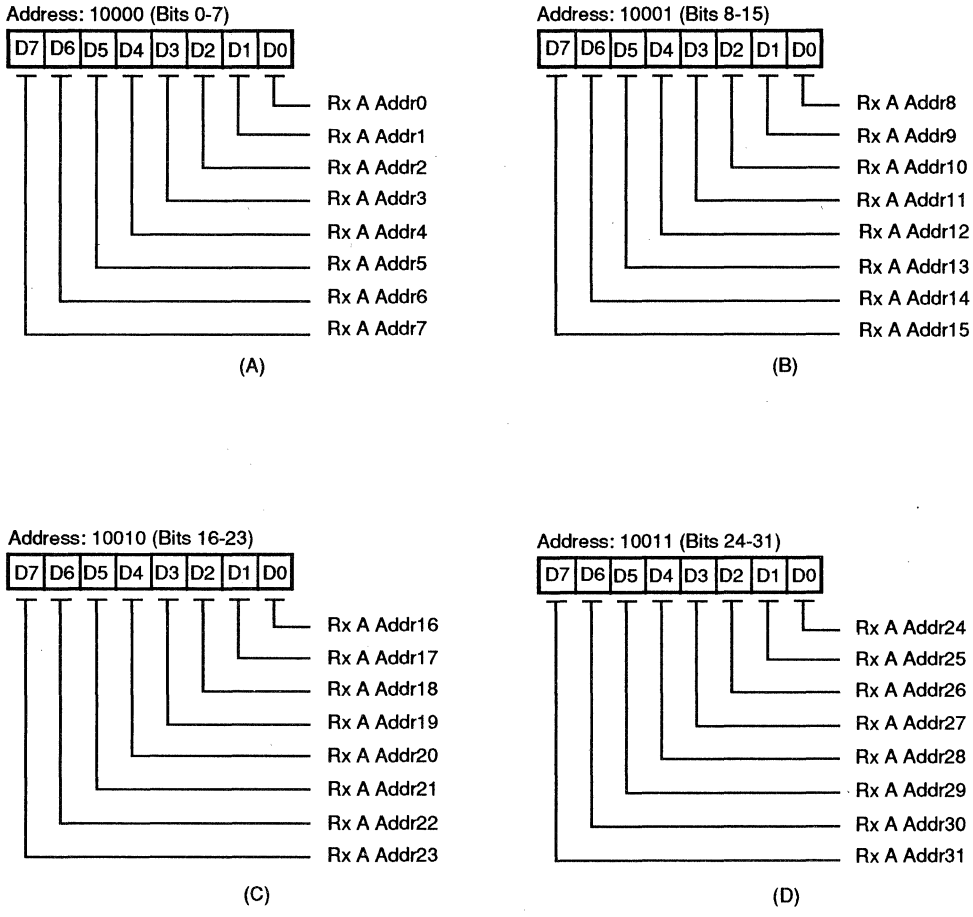


Figure 5-36. Receive DMA Address Registers

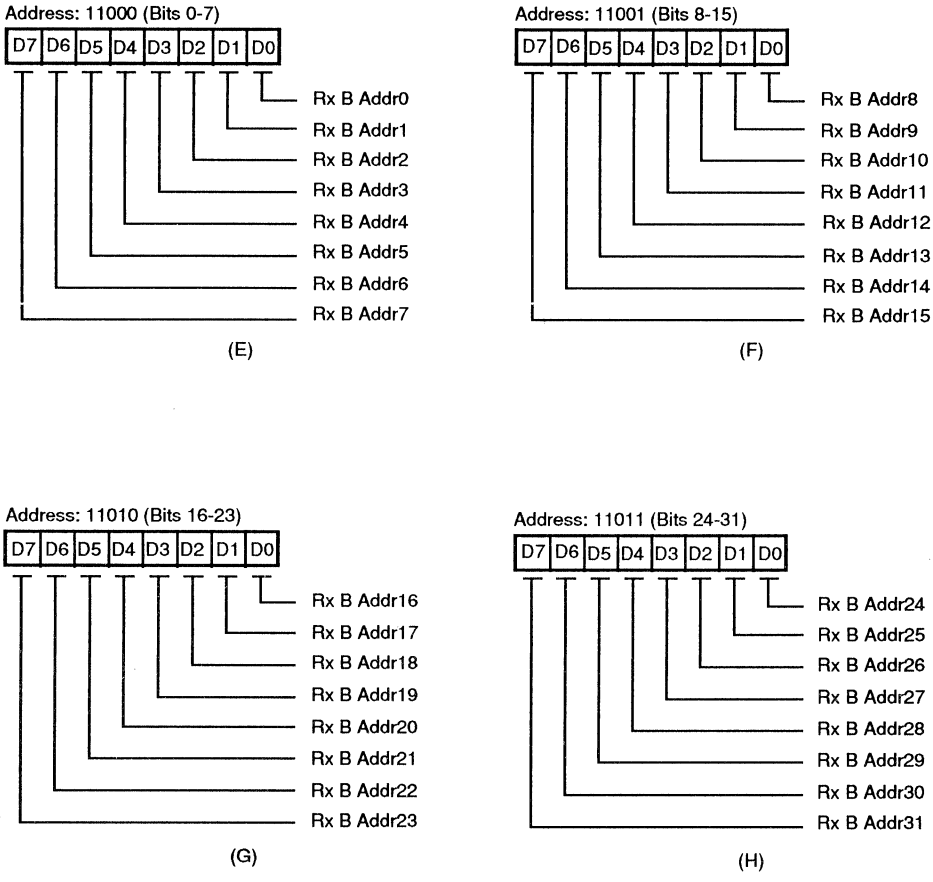


Figure 5-36. Receive DMA Address Registers (Continued)

5.6.12 Transmit DMA Address Registers A, B

There are two sets of Transmit DMA Address Registers, one set for Transmit DMA Channel A and one set for Transmit DMA Channel B. Each set consists of four registers, one for address bits 7 - 0, one for address bits 15 - 8, one for address bits 23 - 16, and one for address bits 31 - 24 as shown in Figure 5-37. These registers are read/write.

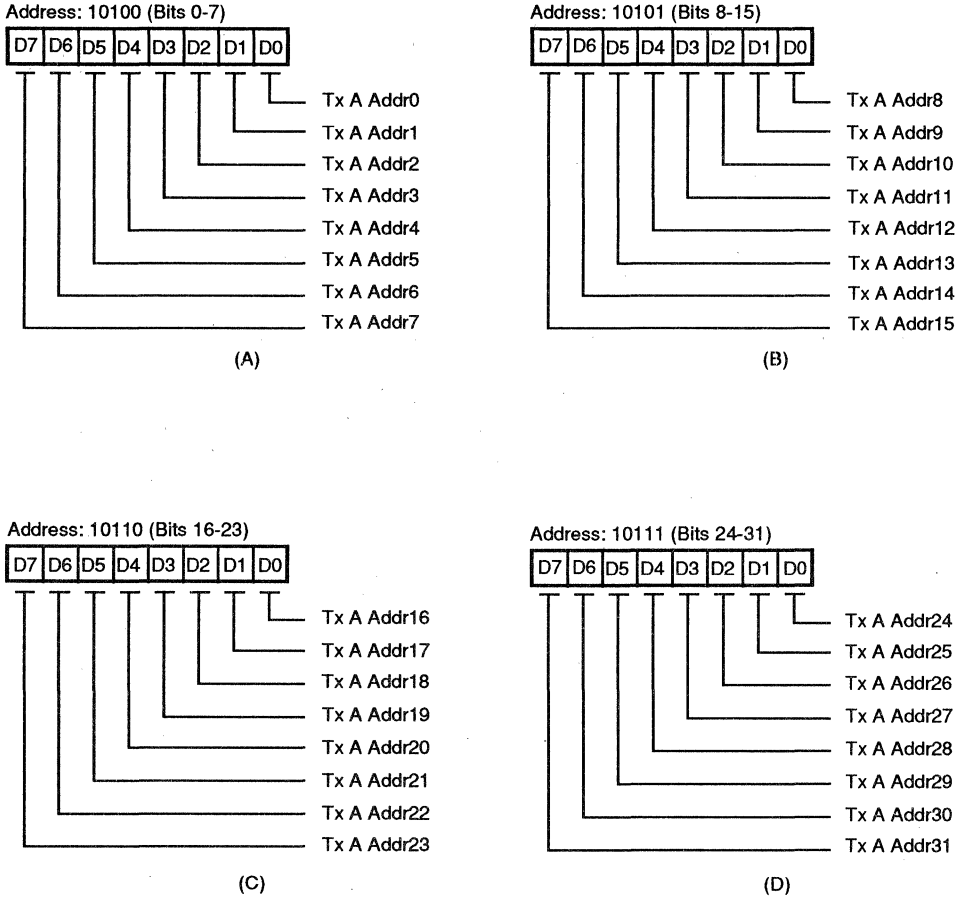


Figure 5-37. Transmit DMA Address Registers

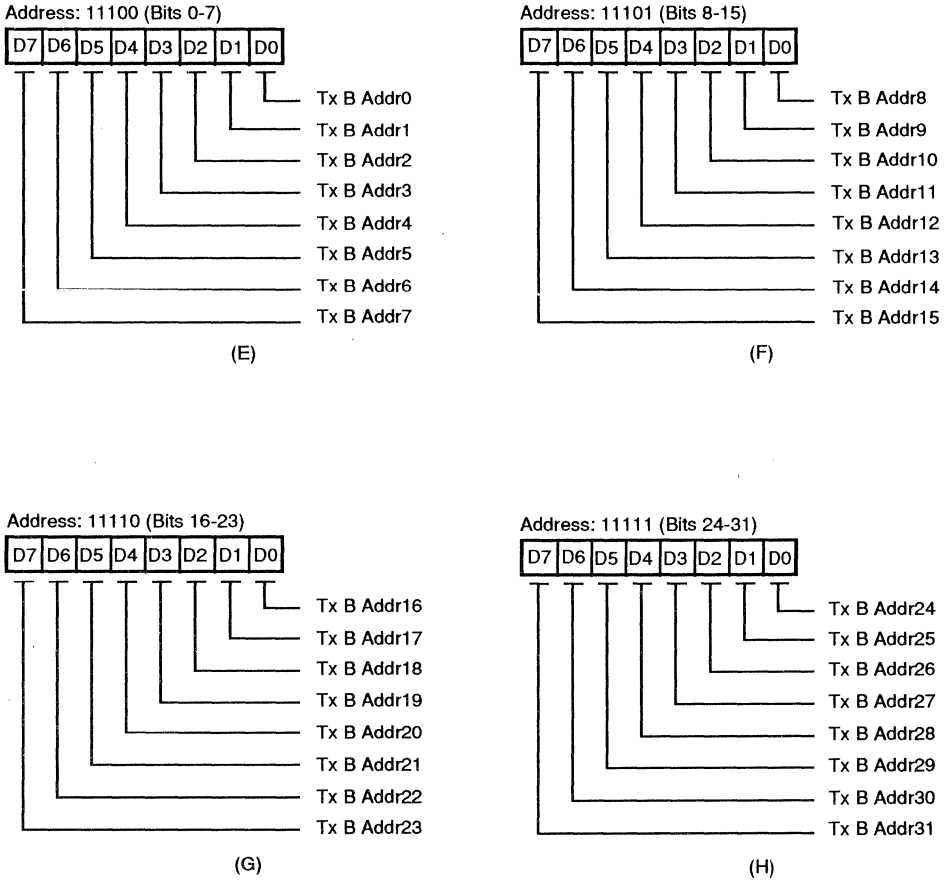


Figure 5-37. Transmit DMA Address Registers (Continued)

5.6.13 Bus Configuration Register

The first write to the ISCC after a hardware reset is always to the Bus Configuration Register. The register is shown in Figure 5-38. The Bus Configuration Register is not affected by any reset function other than a hardware reset and is accessible only after the hardware reset. Note that when writing to the Bus Configuration Register, /AS and A1/A//B are used to program certain bus interface features. Refer to the Bus Interface Unit description for details.

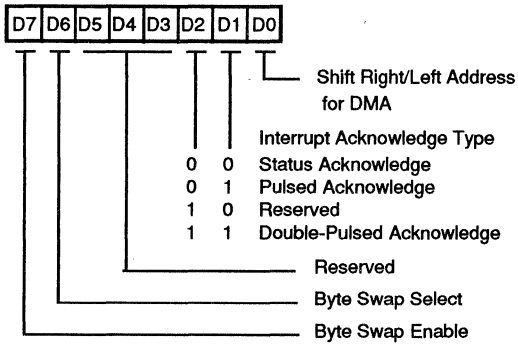


Figure 5-38. Bus Configuration Register

Bit D7 is the Byte Swap Enable.

A zero in this bit disables the byte swap feature. Thus the ISCC accepts DMA transferred data from memory on the lower eight bits of the address data bus (AD) and ignores data on the upper eight bits.

A one in this bit enables the byte swap feature and the ISCC accepts DMA transferred data from memory on either the upper or lower eight bits of the bus depending on the state of A0, the least significant address bit. Big endian or little endian selection is made through bit D6.

Note that whether or not this feature is enabled, when data is DMA transfered from the ISCC to memory, the ISCC replicates the same data on both the lower and upper eight bits of the bus. Writing to memory is controlled by the external generation of appropriate memory enable or strobe signals.

Bit D6 controls the odd / even byte selection when the Byte Swap feature is enabled. If Byte Swap Select is a 1, then even address bytes (transfers where the DMA address has A0 equal to 0) are accepted by the ISCC on the lower eight bits of the bus and odd address bytes (transfers where the DMA address has A0 equal to 1) are accepted on the upper eight bits of the bus. If Byte Swap Select is a 0, then even address bytes (transfers where the DMA address has A0 equal to 0) are accepted by the ISCC on the upper eight bits of the bus and odd address bytes (transfers where the DMA address has A0 equal to 1) are accepted on the lower eight bits of the bus.

Bits D5 through D3 are reserved and should be programmed zero.

Bits D2 and D1 program the Interrupt acknowledge type according to Table 5-16.

Table 5-16. Interrupt Acknowledge Programming

D2	D1	Interrupt Acknowledge Type
0	0	Status Acknowledge
0	1	Pulsed Acknowledge
1	0	Reserved
1	1	Double Pulse Acknowledge

The Status Acknowledge is compatible with the 68000 family of microprocessors and the Double Pulse Acknowledge is compatible withe the 8086 family or microprocessors.

Bit D0 selects the Shift Right / Shift Left address decoding mode for the DMA cell only. A 1 in this bit selects the Shift Right mode. In this mode, when the ISCC is in the multiplexed bus mode, the addresses to the DMA cell registers is decoded from address data lines AD4 through AD0. A 0 in this bit selects the Shift Left mode. In this mode, when the ISCC is in the multiplexed bus mode, the addresses to the DMA cell registers is decoded from address data lines AD5 through AD1.

APPENDIX A
ISCC INTERFACE TO THE 68000 AND 8086

INTRODUCTION

The ISCC uses its flexible bus to interface with a variety of microprocessors and microcontrollers; included are the 68000 and 8086.

The Z16C35 ISCC is a Superintegration form of the 85C30/80C30 Serial Communications Controller (SCC). Super integration includes four DMA channels, one for each receiver and transmitter and a flexible Bus Interface Unit (BIU). The BIU supports a wide variety of buses including the bus types of the 680x0 and the 8086 families of microprocessors.

This Application Note presents the details of BIU operation for both slave peripheral and DMA modes. Included are application examples of interconnecting an ISCC to a 68000 and a 8086 (These examples are currently under test).

ISCC BUS INTERFACE UNIT (BIU)

The following subsections describe and illustrate the functions and parameters of the ISCC Bus Interface Unit.

Overview

The ISCC contains a flexible bus interface that is directly compatible with a variety of microprocessors and microcontrollers. The bus interface unit adds to the chip by allowing ease of connection to several standard bus configurations; among others are the 68000 and the 8086 family microprocessors. This compatibility is achieved by initializing the ISCC after a reset to the desired bus configuration.

The device also configures to work with a variety of other 8- or 16-bit bus systems and is used with address/data multiplexed or non-multiplexed buses. In addition, the wait/ready handshake, the interrupt acknowledge, and the bus high byte/low byte selection are all programmable. Separate read/write, data strobe, write, read, and address strobe signals are available for direct system interface with a minimum of external logic.

Modes Description

There are basically two bus modes of operation: multiplexed and non-multiplexed. In the multiplexed bus mode, the ISCC internal registers are directly accessible as separate registers with their own unique hardware addresses. By contrast, in the non-multiplexed mode, all registers access through an internal pointer which first loads with the register address. Loading of the pointer is done as a data write. In either case, there are some external addressing signals.

Chip Enable (CE) allows external selection through the decode of upper order address bits like accessing separate chips. A separate input (not part of the AD15-0 bus connection) selects between the internal SCC and DMA sections of the chip. This input is A0/SCC/DMA and provides direct transfers to the appropriate chip subsystem; either multiplexed or non-multiplexed bus mode.

A second separate input (not part of the AD15-0 bus connection) provides for a selection between the internal SCC; both channels A and B (Table A-1). This input is A1/A/B and provides direct transfers to the appropriate SCC channel when A0/SCC/DMA selects the SCC; either multiplexed or non-multiplexed bus mode. Note that these two signals, A1/A/B and A0/SCC/DMA, are inputs when the ISCC is a slave peripheral; they become outputs when the ISCC is a bus master during DMA operations.

Table A-1. Accessing the ISCC Registers

A0/SCC/DMA	A1/A/B	ACCESS
1	1	SCC Channel A
1	0	SCC Channel B
0	x	DMA

The following discussions assume knowledge of the SCC Serial Communications Controller opertions and refer to internal register designations. For a detailed explanation, refer to the SCC Technical Manual.

Non-multiplexed Bus Operation

When the ISCC initializes for non-multiplexed operation, Write Register 0 (WR0) takes on the form of WR0 in the Z8530, Write Register Bit Functions (Figure A-1). Register addressing for the SCC section is (except for WR0 and RR0) accomplished as follows. Programming the write registers requires two write operations. Reading the read registers requires both a write and a read operation.

The first write is to WR0 which contains three bits that point to the selected register (note the point high command). The second write is the actual control word for the selected register. If the second operation is a read, the selected register is accessed. When in the non-multiplexed mode, all registers in the SCC section of the ISCC, including the data registers, access this way.

The pointer register automatically clears after the second read or write operation so WR0 (or RR0) addresses again. There is no direct access to the data registers. They are addressed through the pointer (this is in contrast to the Z8530 which allows direct addressing of the data registers through the C/D pin).

When the ISCC starts for non-multiplexed operation, register addressing for the DMA section is (except for CSAR) accomplished as follows. It is completely independent of the SCC section register addressing. Programming the write registers requires two write operations and reading the read registers requires both a write and a read operation. The first write is to the Command Status Address Register (CSAR) which contains five bits that point to the selected register (CSAR bits 4 - 0). The second write is the actual control word for the selected register. If the second operation is a read, the selected register is accessed. The pointer bits automatically clear after the second read or write operation so CSAR addresses again. When in the non-multiplexed mode, all registers in the DMA section of the ISCC are accessed.

Multiplexed Bus Operation

When the ISCC initializes for multiplexed bus operation, all registers in the SCC section are directly addressable with the register address occupying AD5 through AD1 or AD4 through AD0 (Shift Left/Shift Right modes).

The Shift Left/Shift Right modes for the address decoding of the internal registers (multiplexed bus) are separately programmable for the SCC and DMA sections. For the SCC section, the programming and operation is the same as the SCC; programming occurs through Write Register 0 (WR0), bits 1 and 0 , and Write Register Bit Functions (Figure A-2). The programming of the Shift Left/Shift Right modes for the DMA section occurs in the BCR, bit 0. In this case, the shift function is similar to the SCC section; with Left Shift, the internal register addresses decode from bits AD5 through AD1. In Right Shift, the internal register addresses decode from bits AD4 through AD0.

During multiplexed bus mode selection, Write Register 0 (WR0) becomes WR0 in the Z8030, Write Register Bit Functions (Figure A-2).

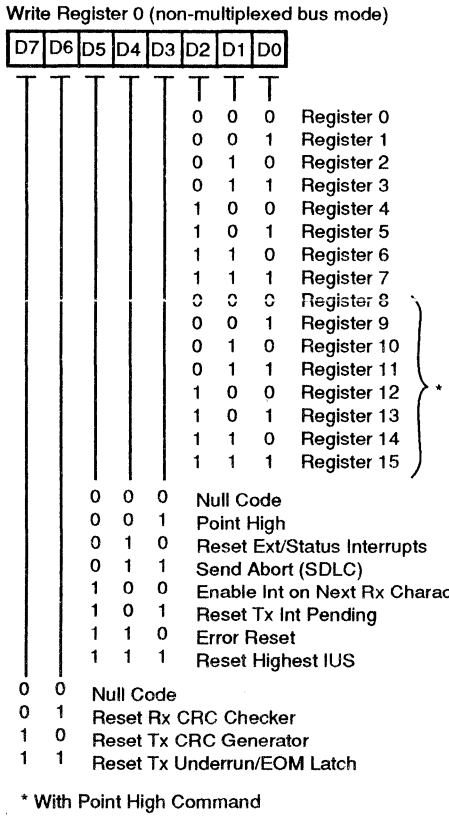


Figure A-1. Write Register 0 Bit Functions (Non-Multiplexed Bus Mode)

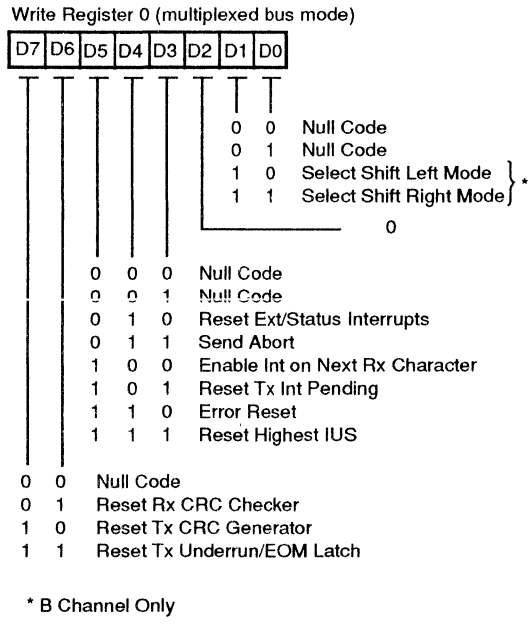


Figure A-2. Write Register 0 Bit Functions (Multiplexed Bus Mode)

BUS DATA TRANSFERS

All data transfers to and from the ISCC are done in bytes regardless of whether data occupies the lower or upper byte of the 16 bit bus. Bus transfers as a slave peripheral are done differently from bus transfers when the ISCC is the bus master during DMA transactions. The ISCC is fundamentally an 8-bit peripheral but supports 16-bit buses in the DMA mode. Slave peripheral and DMA transactions appear in the next sections.

Data Bus Transfers as a Slave Peripheral

When accessed as a peripheral device (when the ISCC is not a bus master performing DMA transfers), only 8 bits transfer. During ISCC register read, the byte data present on the lower 8 bits of the bus is replicated on the upper 8 bits of the bus. Data is accepted by the ISCC only on the lower 8 bits of the bus.

ISCC DMA Bus Transfers

During DMA transfers, when the ISCC is bus master, only byte data transfers occur. However, data transfers to or from the ISCC on the upper 8 bits of the bus or on the lower 8 bits of the bus. Moreover, odd or even byte transfers activate on the lower or upper 8 bits of the bus. This is programmable and explained next.

During DMA transfers to memory from the ISCC, only byte data transfers occur. Data appears on the lower 8 bits and replicates on the upper 8 bits of the bus. Thus, the data is written to an odd or even byte of the system memory by address decoding and strobe generation.

During DMA transfers to the ISCC from memory, byte data only transfers. Normally, data appears only on the lower 8

bits of the bus. However, the byte swapping feature determines which byte of the bus data is accepted. The byte swapping feature activates by programming the Byte Swap Enable bit to a 1 in the BCR. The odd/even byte transfer selection occurs by programming the Byte Swap Select bit in the BCR. If Byte Swap Select is a 1, then even address bytes (transfers where the DMA address has A0 = 0) are accepted on the lower 8 bits of the bus. Odd address bytes (transfers where the DMA address has A0 = 1) are accepted on the upper 8 bits of the bus. If Byte Swap Select is a 0, then even address bytes (transfers where the DMA address has A0 = 0) are accepted on the upper 8 bits of the bus. Odd address bytes (transfers where the DMA address has A0 =1) are accepted on the lower 8 bits of the bus.

Bus Interface Handshaking

The ISCC supports data transfers by either a data strobe (DS) combined with a read/write (R/W) status line, or separate read (RD) and write (WR) strobes. These transactions activate via chip enable (CE).

ISCC programming generates interrupts upon the occurrence of certain internal events. The ISCC internally prioritizes its own interrupts, therefore, the ISCC presents one interrupt to the processor even though lower priority internal interrupts may be pending. Interrupts are individually enabled or disabled. Refer to the sections on the SCC core.

Interrupt Acknowledge (INTACK) is an input to the ISCC showing that an interrupt acknowledge cycle is progressing. INTACK is programmed to accept a status acknowledge, a single pulse acknowledge, or a double pulse acknowledge. This programming activates in the BCR. The double pulse acknowledge is compatible with 8X86 family microprocessors and the status acknowledge is compatible with 68000 family microprocessors.

During an interrupt acknowledge cycle, the SCC and DMA interrupt priority daisy chain internally resolves. Thus, the highest priority internal interrupt is presented to the CPU.

The ISCC can return an interrupt vector that encodes with the type of interrupt pending enabled during this acknowledge cycle. The ISCC may request an interrupt but not return an interrupt vector [note that the no vector bit(s) in the SCC section (WR9 bit 1) and in the DMA section (ICR bit 5) individually control whether or not an interrupt vector returns by these cores]. The interrupt vector can program to include a status field showing the internal ISCC source of the interrupt. During the interrupt acknowledge cycle, the ISCC returns the interrupt vector when INTACK, RD or DS go active and IEI is high (if the ISCC is not programmed for the no vector option).

During the programmed pulsed acknowledge type (whether single or double), INTACK is the strobe for the interrupt vector. Thus when INTACK goes active, the ISCC drives the bus and presents the interrupt vector to the CPU. When the status acknowledge type programs, the ISCC drives the bus with the interrupt vector when RD or DS are active.

WAITRDY programs to function either as a WAIT signal or a READY signal using the BCR write. When programmed as a wait signal, it supports the READY function of 8X86 family microprocessors. When programmed as a ready signal, it supports the DTACK function of 680x0 family microprocessors.

The WAIT/RDY signal functions as an output when the ISCC is not a bus master. In this case, this signal serves to indicate when the data is available during a read cycle, when the device is ready to receive data during a write cycle, and when a valid vector is available during an interrupt acknowledge cycle.

When the ISCC is the bus master (DMA section has taken control of the bus), the WAIT/RDY signal functions as a WAIT or RDY input. Slow memories and peripheral devices use WAIT to extend the data strobe (/DS) during bus transfers. Similarly, memories and peripheral devices use RDY to indicate valid output or that it is ready to latch input data.

CONFIGURING THE BUS

The bus configuration programming is done in two separate steps (actually it is one operation), to enable the write to the Bus Configuration Register (BCR). The first operation that accesses the ISCC after a device reset must be a write to the BCR since this is the only time that the BCR is accessible. Before and during the write, various external signals are sampled to program bus configuration parameters. During this write, the AØ/SCC//DMA pin must be Low.

Address strobe programs multiplexed/non-multiplexed selection. In a non-multiplexed bus environment, address strobe (as an input) is not used but tied high through a suitable pull-up resistor. Thus, no address strobe is present before the BCR write. Then, when write to the BCR takes place, the non-multiplexed mode is programmed because there is no address strobe before this first write to the device. Note that address strobe becomes an output during DMA operations so it is not tied directly to Vcc.

During the write operation to the BCR, the A1/A/B input is sampled to select the function of the WAIT/RDY pin (Table A-2). When the BCR Write is to the SCC Channel A (A1/A//B High during the BCR write), the WAIT/RDY signal functions as a wait. When the BCR Write is to Channel B (A1/A//B Low during the BCR write), the WAIT/RDY signal functions as a ready.

Table A-2. Signals Sampled During the BCR Write

A1/A//B	WAIT/RDY Function
1	WAIT (8086 RDY compatible)
0	READY (68000 DTACK compatible)

This programming affects the function of the WAIT/RDY signal both as an input, when the ISCC is bus master during DMA operations, and as an output when the ISCC is a bus slave.

With this programming, the ISCC is immediately configured to function successfully on this first and subsequent bus transactions. The remaining bus configuration options are programmed by the value written to the BCR.

Bit 0 of the BCR controls the Shift Left/Shift Right address decoding modes for the DMA section. In this case, the shift function is similar to the SCC section. During Left Shift, the internal register addresses decode from bits AD5 through AD1. During Right Shift, the internal register addresses are decode from bits AD4 through AD0. This function is only applicable in the multiplexed bus mode.

Bits 1 and 2 of the BCR control the interrupt acknowledge type as shown in the Table A-3.

Table A-3. BCR Control of Interrupt Acknowledge

BCR bit 2	BCR bit 1	Interrupt Acknowledge
0	0	Status Acknowledge
0	1	Pulsed Acknowledge (single)
0	1	Reserved (action not defined)
1	1	Double Pulsed Acknowledge

The Status Acknowledge remains active throughout the interrupt cycle and is directly compatible with the 680x0 family interrupt handshaking. The Status Acknowledge signal latches with the rising edge of AS for multiplexed bus operation. It latches by the falling edge of the strobe (RD or DS) for non-multiplexed bus operation. The Pulsed Acknowledges are timed to be active during a specified period in the interrupt cycle. The Double Pulsed Acknowledge is directly compatible with the 8x86 family interrupt handshaking. Refer to the timing diagrams in the ISCC Product Specification for details on the Acknowledge signal operation.

Reserve bits 3, 4, and 5 of the BCR program as zeros. Bits 6 and 7 of the BCR control the byte swap feature (Table A-4). Byte swap is applicable only in DMA transfers when the ISCC is the bus master and only affects ISCC data acceptance (transfers from memory to the ISCC).

Table A-4. Byte Swap Control

Enable (BCR bit 7)	DMA Data Read by the ISCC
0	lower 8 bits of bus only
1	upper or lower 8 bits of bus

Swap Select*	A0	DMA Data read by the ISCC
0	0	upper 8 bits of bus
0	1	lower 8 bits of bus
1	0	lower 8 bits of bus
1	1	upper 8 bits of bus

* BCR bit 6

The following application examples explain and illustrate the methods of interfacing the ISCC to a Motorola 68000 and an Intel 8086.

68000 Interface to the ISCC

Figure A-3 shows a connection of the ISCC to a 68000 microprocessor. The 68000 data bus connects directly, or through bus transceivers, to the ISCC address/data bus. R/W and RESET also directly connect. In this example, the ISCC is on the lower half of the bus; DS of the ISCC connects to LDS of the 68000. The processor address lines decode to produce a chip enable for the ISCC. In addition, processor addresses A1 and A2 connect to A0/SCC/DMA and A1/A/B, respectively, through a tri-state driver.

The driver is normally ON (enabled) but turns OFF by BGACK to grant the bus to ISCC for DMA transfers. This is done since the A0/SCC/DMA and A1/A/B pins become outputs during DMA transfers and should not drive the system address bus. RD and WR tie high through independent pull-ups. They are not used in this application but become active outputs during DMA transfers and are not tied directly to Vcc.

Although not shown in Table A-5, the A0/SCC/DMA and A1/A/B pins may be decoded during DMA transfers to identify the active DMA channel.

Table A-5. DMA A/B Channel Decode

A1/A/B	A0/SCC/DMA	DMA Channel
1	1	Receiver Channel A
1	0	Transmitter Channel A
0	1	Receiver Channel B
0	0	Transmitter Channel B

External logic can use this information to abort a DMA in progress.

For normal slave device bus interaction, a DTACK is generated. WAIT/RDY is programed for ready operation and INTACK programs for the status type. WAIT/RDY generates a DTACK for normal data transfers and interrupt responses. Additional logic may be required when other interrupt sources are present.

During DMA transfers, the ISCC becomes bus master. Becoming bus master is done through the BUSREQ output and BUSACK input signals of the ISCC. They connect to an external bus arbitration circuit. This circuit performs bus arbitration for multiple bus master requests and generates bus grant acknowledge (BGACK) which controls certain bus drive signal sources.

When the ISCC becomes the bus master, a 32-bit address generation by the DMA section is output on the ISCC address/data bus. The lower 16 bits of this address store in an external latch by AS (Address Strobe). Also, the upper 16 bits of this address store in an external latch by UAS (Upper Address Strobe). With BGACK low (active) and with the processor address lines tri-stated, the latch outputs drive the system address bus.

AS is pulled high by an external resistor. This pull-up insures an inactive AS (at a logic high level) when the ISCC is not driving this signal. Therefore, on power up or after a RESET, AS is inactive and programs the non-multiplexed bus mode on BCR write.

In this application, the outputs of the address latches are connected to the address bus so that A1 through A23 of the ISCC drives the system address bus (the ISCC provides a total of 32 address lines). A0 from the address latch is diverted to logic which generates UDS and LDS bus signals from the ISCC data strobe (DS). UDS is generated when A0 is low and LDS is generated when A0 is high. The lower and upper data strobes are applied to the system bus through tri-state drivers which are enabled only when BGACK is active. Bus direction is now controlled by the ISCC R/W signal which is now an output.

For initialization, the BCR write (the first write to the ISCC after RESET) is done with A2 = 0 (A1/A/B ISCC input at logic low). This selects the ready option of the WAIT/RDY signal to conform to the 68000 bus style. The AS signal programming of the non-multiplexed bus has already been discussed. The BCR is written with C0h to enable byte swapping. It also selects the sense of byte swapping with respect to A0 appropriate to this bus style and selects the STATUS type of interrupt acknowledge.

8086 Interface with the ISCC

Figure A-4 shows the connection of the ISCC to an 8086 microprocessor and companion clock state generator. In this application, the ISCC connects for multiplexed address access to the internal ISCC registers. AD15 through AD0 of the 8086 connect directly, or through a bus transceiver, to the corresponding AD15 through AD0 address/data ISCC bus pins. RD and WR are directly compatible and tie together to form the read and write bus signals.

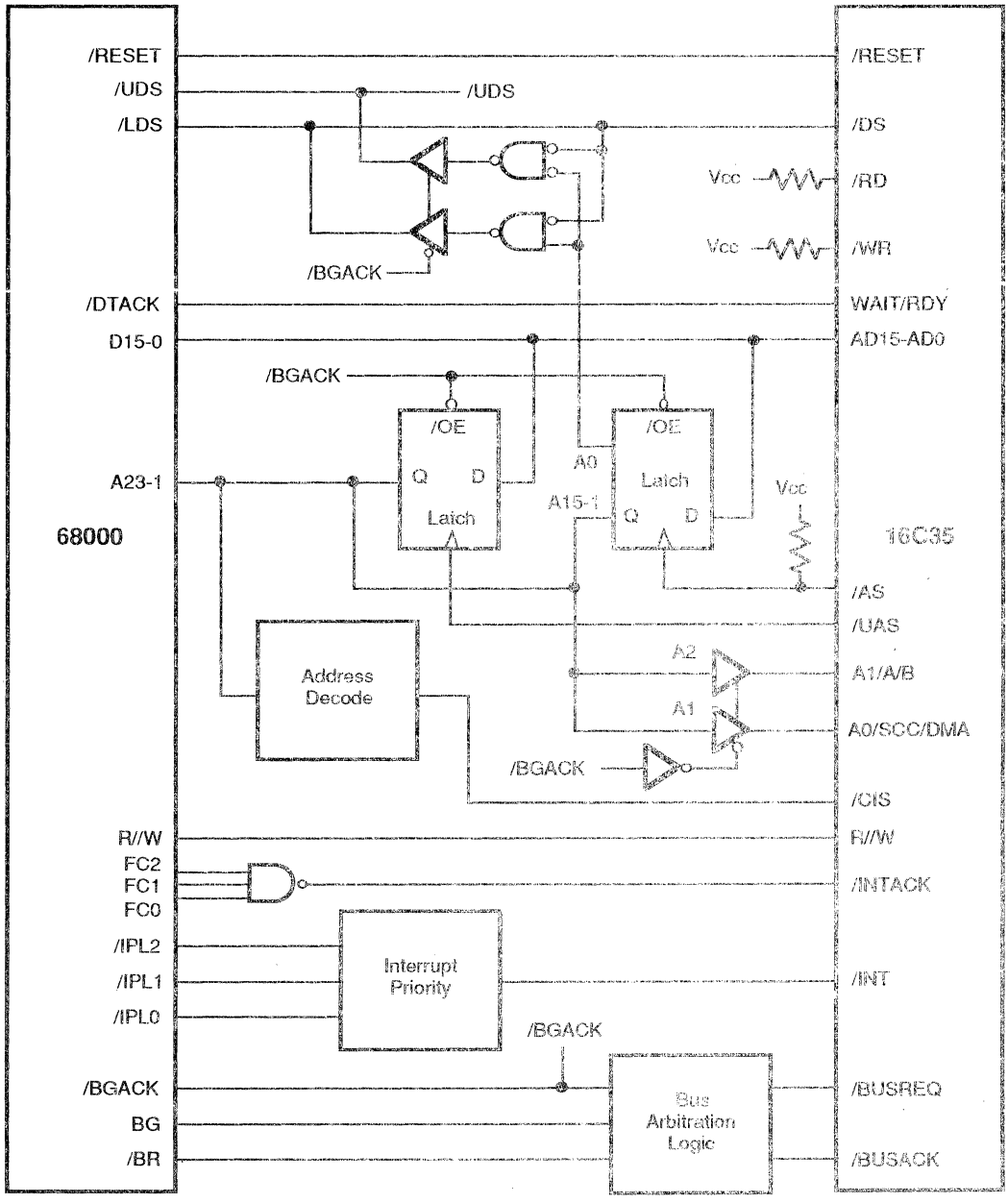
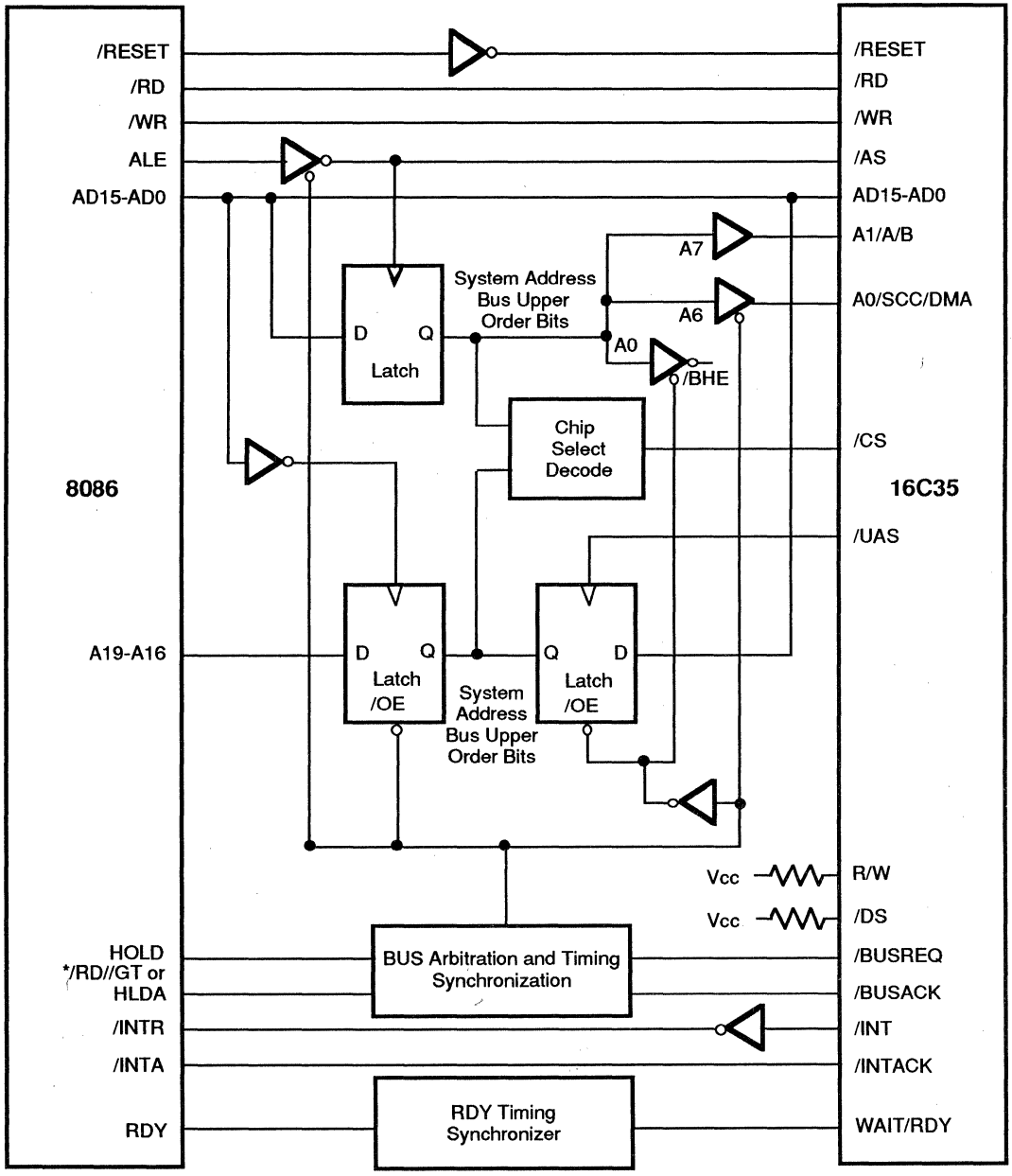


Figure A-3. ISCC Interface to a 68000 Microprocessor



* maximum mode

Figure A-4. ISCC Interface to an Intel 8086 Microprocessor

When the ISCC becomes a bus master during DMA operations, RD and WR of the 8086 are tri-stated which allows the corresponding ISCC signals to control the bus transactions. The sense of RESET reverses, so the ISCC RESET signal inverts from the reset applied to the 8086 from the clock state generator.

RD/WR and DS of the ISCC are inactive in this application and tie high. They tie high through independent pull-ups since these signals become active when the ISCC is bus master during DMA transactions.

Assuming other devices in the system, the ISCC chip enable input (CE) activates from a decode of the address. In this example, the ISCC internally decodes addresses A1 through A5 and uses A6 and A7, externally. Thus, the address decode circuitry decodes address lines A0 and A8 and above. The decode of A0 for chip enable places the ISCC as an 8-bit peripheral on the lower byte of the bus. A0 and the upper level address lines (including A6 and A7) demultiplex from the 8086 address/data bus through a latch strobed by ALE.

The demultiplexed addresses A6 and A7 connect to A0/SCC/DMA and A1/A/B, respectively, of the ISCC to control selection of the DMA and SCC channels A and B. This connects through the tri-state drivers. They enable when the 8086 is the bus master and disable when the ISCC is bus master. This prevents the ISCC from improperly driving the system address bus since A0/SCC/DMA and A1/A/B become active outputs when the ISCC is the bus master.

The address map for the ISCC appears in Table A-6 for this application.

Table A-6. ISCC Address Map

A0	A1-A5	A6	A7	Registers Addressed
1	x	x	x	ISCC not enabled
0	-	0	x	DMA Registers per A1 - A5
0	-	1	1	SCC Core Channel A Registers
0	-	1	0	SCC Core Channel B Registers

Since A0 specifies the lower byte of the bus and includes the chip enable decode, the internal ISCC register addresses decode without A0. Thus, Table 6 implies that the Left Shift address decode selection is made for both the SCC and DMA sections of the ISCC. The left shift selection is the default selection after reset. Left/Right Shift selection programming is discussed later.

The ALE signal of the 8086 applies to AS of the ISCC through an inverting tri-state buffer. The buffer disables when the ISCC becomes a bus master during DMA transactions. This prevents conflicts since ALE remains active even when the 8086 is in the HOLD mode during DMA transfers. Now, the ISCC AS is an active output. The address strobe for the demultiplexing latch of addresses A0 through A15 connects on the ISCC side of the ALE tri-state buffer. This allows the latch to serve two functions; to hold either the 8086 or the ISCC address when it is bus master.

After reset, ALE is active and the tri-state buffer enabled. This supplies address strobes to the ISCC. The presence of one of these address strobes, before writing to the BCR, programs the ISCC to the multiplexed bus mode of operation. The ISCC chip enable (CE) can be inactive and still recognize an address strobe (AS) before the BCR write (Figure 4 shows open latches when the input strobe is low).

When the ISCC is bus master during DMA transactions, BHE generates from A0. This is done from the output of the lower order address latch through an inverting tri-state driver. This driver enables only when the ISCC is the bus master. Whole word transfers are not done by the ISCC DMA, thus, BHE generated for the ISCC is always the inverse of A0.

The upper bus system address lines demultiplex from the 8086 and the ISCC in separate latches. Like the 68000 example, high order address lines from the ISCC latch via UAS (upper address strobe). The separate latches drive the same upper order address lines. A16 from the ISCC connects to the corresponding A16 address bus line as derived from the 8086. The output of the two latches alternately enable depending upon bus mastership.

The diagram shows INT fron the ISCC connected to the 8086 INTR input via an inverter since these signals are of opposite sense. In actual practice, the ISCC interrupt request is first processed by an interrupt priority circuit. INTA (Interrupt Acknowledge) of the 8086 connects directly to the INTACK input of the ISCC. Conforming to the 8086 style of interrupt acknowledge, the ISCC is programed to the Double Pulse Interrupt Acknowledge type. When this selection occurs, the ISCC responds to two interrupt acknowledge pulses. The first pulse is recognized but no action follows. The second pulse causes the ISCC to go active on the data bus and return the interrupt vector to the CPU. This action also takes place with the Single Pulse Interrupt Acknowledge type selection, except that the bus goes active with the first and only interrupt acknowledge pulse.

To start, the BCR write (first write to the ISCC after RESET) is done with A7 = 1 (A1/A/B ISCC input at logic high). This selects the wait option of the WAIT/RDY signal to conform to the 8086 bus style. The AS signal programming of the multiplexed bus was covered earlier. The BCR is written

with 86h to enable byte swapping, select the sense of the byte swapping with respect to A0 (appropriate to this bus style), and select the Double Pulse type of interrupt acknowledge.

When the ISCC begins DMA transfers, it communicates requests for the bus through BUSREQ and BUSACK. The 8086 receives and grants bus requests through HOLD and HLDA in the minimum mode and through RQ/GT in the maximum mode. Depending upon the system requirements, there could be more than one potential bus master. Therefore, there is a requirement for a bus arbitration circuit.

The minimum mode connection is relatively straightforward. The maximum mode configuration requires a translation of the ISCC BUSREQ and BUSACK signals into/from the 8086 RQ/GT timed pulse style of handshake. Refer to the information on the 8086 for detailed application information.

The ISCC WAIT/RDY output is compatible with the 8086 clock generator RDY input except that one edge of the signal must be synchronous with the 8086 clock. The synchronization occurs through external circuitry. Refer to the information on the 8086 for detailed application information.

ZILOG DOMESTIC SALES OFFICES AND TECHNICAL CENTERS

CALIFORNIA
Agoura 818-707-2160
Campbell 408-370-8120
Tustin 714-838-7800

COLORADO
Boulder 303-494-2905

FLORIDA
Largo 813-585-2533

GEORGIA
Norcross 404-448-9370

ILLINOIS
Schaumburg 708-517-8080

NEW HAMPSHIRE
Nashua 603-888-8590

NEW JERSEY
Clark 201-382-5700

NORTH CAROLINA
Raleigh 919-790-7706

OHIO
Seven Hills 216-447-1480

PENNSYLVANIA
Ambler 215-653-0230

TEXAS
Dallas 214-987-9987

WASHINGTON
Seattle 206-523-3591

INTERNATIONAL SALES OFFICES

CANADA
Toronto 416-673-0634

GERMANY
Munich 49-89-672-045

JAPAN
Tokyo 81-3-587-0528

HONG KONG
Kowloon 852-7238979

KOREA
Seoul 82-2-552-5401

SINGAPORE
Singapore 65-2357155

TAIWAN
Taipei 886-2-741-3125

UNITED KINGDOM
Maidenhead 44-628-392-00

The information contained herein is subject to change without notice. Zilog will not be responsible for any such changes. Zilog will not be responsible for notifying any user of changes. Zilog assumes no responsibility for the use of any circuitry or other technology embodied in a Zilog product. No patent licenses, intellectual property rights, or other rights are implied.

Zilog will not be responsible for any damage to the user that may result from accidents or any other reasons during operations of the products described herein.

All specifications (parameters) are subject to change without notice. Zilog will not be responsible for any such changes. Zilog will not be responsible for notifying any user of changes. The applicable Zilog test documentation will specify which parameters are tested.

Zilog, Inc. 210 Hacienda Ave., Campbell, CA 95008-6609
Telephone (408) 370-8000 TWX 910-338-7621