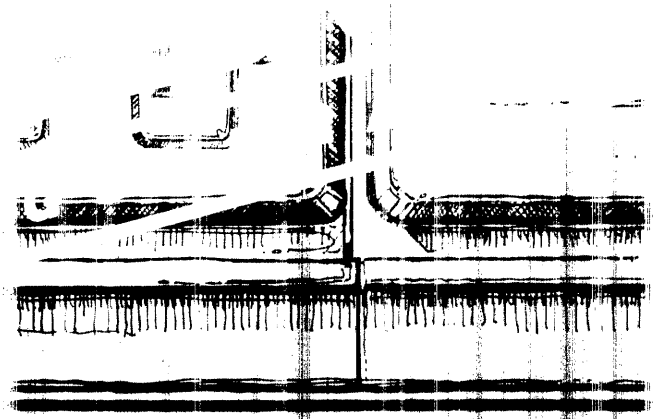


Data General

Technical Refere

DESKTOP GENERATION™

Communications Interfaces

Notice

Data General Corporation (DGC) has prepared this document for use by DGC personnel, customers, and prospective customers. The information contained herein shall not be reproduced in whole or in part without DGC's prior written approval.

DGC reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult DGC to determine whether any such changes have been made.

The terms and conditions governing the sale of DGC hardware products and the licensing of DGC software consist solely of those set forth in the written contracts between DGC and its customers. no representation or other affirmation of fact contained in this document including but not limited to statements regarding capacity, response-time performance, suitability for use or performance of products described herein shall be deemed to be a warranty by DGC for any purpose, or give rise to any liability of DGC whatsoever.

In no event shall DGC be liable for any incidental, indirect, special or consequential damages whatsoever (including but not limited to lost profits) arising out of or related to this document or the information contained in it, even if DGC has been advised, knew or should have known of the possibility of such damages.

Ordering No. 014-000769

Rev. 00, October 1983

Preface

The technical reference manuals for Desktop Generation™ computers and their peripherals are written for assembly language programmers, systems analysts, and engineers. This set of manuals, together with two companion programmer's references, contains the information you need to: 1) write assembly language software, including I/O subroutines; 2) knowledgeably expand your system; 3) learn how your system operates at the card level; and 4) design custom interfaces.

This manual discusses the functional and physical organization of the asynchronous/synchronous communications interfaces available for Desktop Generation systems.

Organization

This book includes two chapters, and an index. It is organized so that portions of it can be read selectively.

- Chapter 1 discusses the functional and physical organization of the universal synchronous/asynchronous multiplexors (USAM), defines their I/O instruction sets, offers guidelines for writing assembly language I/O subroutines, and contains theory of operation.
- Chapter 2 discusses the functional and physical organization of the Model 4207 asynchronous controller card, defines its I/O instruction set, offers guidelines for writing assembly language I/O subroutines, and contains theory of operation.

A documentation comment form follows the index. It invites you to help Data General improve its publications by commenting on this book.

Related Manuals

A comprehensive documentation set supports all the hardware and software products available for Desktop Generation computers. The hardware-related books listed below fall into three categories: the technical reference series; the user guides for operating, installing, and testing; and the introductory guide for Desktop Generation computers.

The following technical and programmer's references address the needs of assembly language programmers and engineers.

16-bit Real Time ECLIPSE Assembly Language Programming

Global in nature, this book explains the processor-independent concepts, functions, and instruction sets of 16-bit ECLIPSE computers. DGC ordering no. 014-000688.

Model 10 and 10/SP Computer Systems
Technical Reference

In addition to the functional and physical organization of Model 10 and 10/SP computers and their technical specifications, this manual explains their processor-unique concepts, functions, and instruction set features. Also included are guidelines for programming the I/O devices, including the diskette subsystem, and explains the theory of operation for the basic components of Models 10 and 10/SP. DGC odering no. 014-000766.

Model 10 and 10/SP System Console
Programmer's Reference

Describes the organization and alphanumeric and graphic features of the system console. Defines the command sets and includes guidelines for programming the monochrome and optional color monitors at assembly and high-level language levels. DGC ordering no. 014-000770.

Model 20 and 30 Computer Systems
Technical Reference

In addition to the functional and physical organization of Model 20 and 30 computers and their technical specifications, this manual explains their processor-unique concepts, functions, and instruction set features. Also included are guidelines for programming the I/O devices, including the diskette subsystem, and explains the theory of operation for the basic components of Models 20 and 30. DGC ordering no. 014-000767.

I/O and Interfacing
Technical Reference

Introduces the microI/O bus and describes the I/O interface required to communicate with this bus and its host Desktop Generation computer. Discusses the I/O instruction set and the I/O program interrupt and data channel facilities. Includes a chapter about the 4210 general-purpose interface, useful to those designing a custom I/O interface for their system. DGC ordering no. 014-000774.

For more detailed information about the microI/O bus and Data General integrated circuits used in the I/O interface, refer to *microNOVA Integrated Circuits Data Manual* DGC ordering no. 014-000074.

Disk Subsystem
Technical Reference

Describes the functional and physical organization of the Model 6271 disk subsystem. Defines the I/O instruction set and provides guidelines for programming the subsystem. DGC ordering no. 014-000768.

Sensor I/O
Technical Reference

Defines instruction sets, offers guidelines for writing assembly language I/O subroutines, describes theory of operation at an overview level, and explains how to connect field wiring for the 4222 digital I/O interface, 4223 analog-to-digital interface, 4224 digital-to-analog interface, and 4335 analog subsystem. DGC ordering no. 014-000775.

IEEE-488 Bus Interface
Technical Reference

Provides the information needed to interface, program in assembly language, and troubleshoot this card in a Desktop Generation system. Reviews the contents of the IEEE-488 bus standard, summarizing its commands, messages, and states, and includes a theory of operation. DGC ordering no. 014-000773.

The following books are how-to manuals written for anyone who needs to know how to install, operate, and test a Desktop Generation system.

Installing Model 10 and 10/SP Systems

The first book that a Model 10 or 10/SP owner should read, explains how to unpack and install either system and its optional peripherals. Simple instructions and ample illustrations make the book accessible to any reader. DGC ordering no. 014-000901.

Operating Model 10 and 10/SP Systems

A logical follow-on to Model 10 and 10/SP installation, this guide takes you from powering up the system and its optional peripherals through performing such routine operations as loading paper in a printer and inserting or removing diskettes. Brings you to the point of loading the system software. Amply illustrated and written for users at any level of experience. DGC ordering no. 014-000900.

Testing Model 10 and 10/SP Systems

Follows the installation and operating manuals with instructions for verifying the operation of Model 10 or 10/SP systems and their optional peripherals. Steps you through the power-up test and Customer Diagnostics and explains how to troubleshoot customer-replaceable components. Simple instructions and diagrams make the book accessible to any user. Includes phone numbers for Data General assistance. DGC ordering no. 014-000902.

Installing Model 20 and 30 Systems

The first book a Model 20 or 30 owner should read, explains how to unpack and install either system and its optional peripherals. Accessibly written and illustrated, for users at any level of experience. DGC ordering no. 014-000904.

Operating Model 20 and 30 Systems

Follows Model 20 and 30 installation, leading you from powering up the system and its optional peripherals through performing such routine operations as loading paper in a printer and inserting or removing diskettes. Brings you to the point of loading the system software. The simple instructions and generous illustrations are suitable for any reader. DGC ordering no. 014-000903.

Testing Model 20 and 30 Systems

A follow-on to the installation and operating manuals, explains how to verify the operation of Model 20 or 30 systems and their optional peripherals. Simple instructions and diagrams lead you through the power-up test, Customer Diagnostics, and trouble-shooting of customer-replaceable components. Includes phone numbers for Data General assistance. DGC ordering no. 014-000905.

This last book is a product overview, addressed to all Desktop Generation users.

The Desktop Generation

Introduces the Desktop Generation, summarizing each model of the family, and describes its many hardware and software products, features, and capabilities. Includes a brief history of Data General, a sampling of applications, and an overview of the customer service and support programs available to you as a Desktop Generation user. DGC ordering no. 014-000751.

Conventions

The following conventions are used throughout this manual.

MNEMONIC	Uppercase sans serif letters indicate a signal name or instruction mnemonic. When a signal is active low, it is barred—for example, $\overline{\text{FDCHE}}$.
argument	Italicized lowercase letters mean that a particular instruction takes an argument. In your program, you must replace this symbol with the exact code for the argument you need.
[optional]	Brackets signify an optional argument. If you decide to use this argument, do not include the brackets in your code; they only set off the choice.

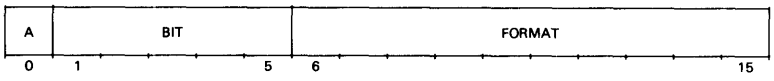
In dialogs between system and user, we use this typeface to show your input:

USER INPUT

and this typeface to show the system's response:

SYSTEM RESPONSE.

In addition, we use the following diagram to show the arrangement of the 16 bits in an instruction. The diagram is always divided into 16 boxes, numbered 0 through 15.



Contents

2 Asynchronous Controller

1 Universal Synchronous/ Asynchronous Multiplexor

The Universal Synchronous/Asynchronous Multiplexors (USAM) are a series of communications multiplexors for use in Desktop Generation™ or Microproducts systems. The USAM cards operate in any I/O slot of the system's computer chassis. They are commonly used as multiterminal controllers or for network interfacing in applications where flexible communications multiplexors are required.

This chapter tells how to program the USAM cards and provides an overview of their theory of operation.

Multiplexor Models

There are two basic designs available: a four-line USAM and a one-line USAM. Their model numbers are:

- 4463-Z four-line USAM (USAM-4) used in Desktop Generation systems
- 4463-ZC four-line USAM (USAM-4) used in Microproducts systems
- 4463-W one-line USAM (USAM-1) used in Desktop Generation systems
- 4463-WC one-line USAM (USAM-1) used in Microproducts systems

The 4463-Z and 4463-ZC models have two asynchronous communications lines and two lines that can be used for either synchronous or asynchronous communications. The asynchronous lines have modem control (RS-232-C) and support asynchronous byte control protocols in full-duplex operation. The two synchronous/asynchronous communications lines are program-selectable, have modem control (RS-232-C), and support both asynchronous and synchronous byte control protocols in full-duplex operation. They also include CRC error checking hardware that supports four program-selectable block check calculation modes when operating in synchronous mode.

The 4463-W and 4463-WC models are single-line, program-selectable, synchronous/asynchronous controllers. The asynchronous features are the same as those described for the USAM-4. The synchronous features are also the same except there are no hardware CRC capabilities. If CRC is required, it must be incorporated into the software.

All line interfaces are individually switch selectable for EIA RS-232-C, EIA RS-422, or 20-mA current loop characteristics for local connection. The RS-232-C interface can also feature remote connection through a modem. Other line characteristics are fully programmable, and operating speeds range from 50 to 19,200 bits per second. The asynchronous data structure may consist of 5 to 8 data bits per character; 1, 1 1/2, or 2 stop bits; and even, odd, or no parity. The synchronous data structure may consist of 5 to 8 data bits per character and even, odd, or no parity. Modem control and associated status signals are supported only on the RS-232-C interface lines.

Technical Specifications

Technical specifications for the USAM series are listed in Table 1-1.

Table 1-1 USAM technical specifications

General Features	
Physical Dimensions	7 x 9-inch PCB (approx.)
Power Requirements	Volts dc / Amperes +5 / 1.62 −5 / 0.10 (0.05 for USAM-1) +12 / 0.20 (0.05 for USAM-1) −12 / 0.20 (0.05 for USAM-1)
Host Interface	Microl/O bus
Number of Lines	USAM-4 Models: 2 asynchronous; 2 synchronous/ asynchronous (lines 0 and 1) USAM-1 Models: 1 synchronous/ asynchronous
Line Interfacing	Switch selectable; RS-232-C, RS-422, and 20-mA current loop
Asynchronous Features	
Lines	
USAM-4	0 thru 3
USAM-1	0
Data Structure	Program selectable; 1 start bit; 5, 6, 7, or 8 data bits
Stop Bits	Program selectable; 1, 1 1/2, or 2
Data Buffering	Two characters for both transmit and receive
Character Checking	Parity, overrun, and framing errors
Parity Type	Program selectable; Even, odd, or none
Synchronous Features	
Lines	
USAM-4	0 and 1
USAM-1	0
Data Structure	Program selectable; 5, 6, 7, or 8 data bits
Data Buffering	Two characters for both transmit and receive
Synchronization	Single or double SYN operation; transparent or non- transparent mode; automatic SYN or DLE-SYN insertion; automatic SYN or DLE stripping;
Block Checking (USAM-4 models only)	Program selectable; CRC-16, CCITT-16, CRC-12, or LRC-8 error checking
Character Checking	Parity and overrun errors
Parity Type	Program selectable; Even, odd, or none

Programming Summary

Functionally, each line of the USAM operates as an independent, programmed I/O interface. Each interface is based on a programmable communications interface (PCI) integrated circuit. The PCI contains two independent devices, a transmitter and a receiver, which provide the capability for full-duplex communications between the CPU and any serial-based data terminal equipment with the following interfaces: 20-mA current loop, EIA RS-232-C, and EIA RS-422. In addition to the PCIs, the USAM-4 contains block checking circuitry that can be programmed to accumulate a separate block check sum, calculated with a separate BCP polynomial, for each synchronous interface.

The USAM-4 can generate the following types of block checksums:

1. CRC-16 (cyclic redundancy checking with 8-bit character transmission codes).
2. CRC-12 (cyclic redundancy checking with 6-bit character transmission codes).
3. VRC/LRC-8 (vertical redundancy checking/longitudinal redundancy checking with 8-bit transmission codes).
4. CCITT-16 (cyclic redundancy checking with 8-bit transmission codes).

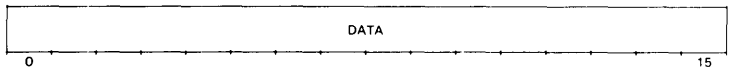
With the USAM-1, if a block check function is required, it must be incorporated into the software.

Table 1-2 Programming summary: USAM specifications

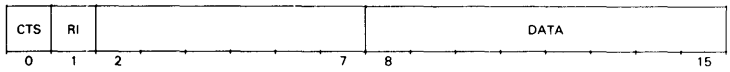
Device Codes		
Primary	34_8	
Secondary	74_8	
Mnemonics		
Primary	ASLM	
Secondary	ASLM1	
Priority Mask Bit	11_8	
Transmission Baud Rates (bits/second)	program selectable; 50, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600, 19200	
Modem Control Functions (RS-232-C interface only)	Carrier Detect	
	Data Set Ready	
	Ring Indicator	
	Clear to Send	
	Request to Send	
	Data Terminal Ready	
	Transmit Timing Out	Synchronous lines only
	Transmit Timing In	Synchronous lines only
	Receiver Timing	Synchronous lines only
Maximum Allowable I/O Latency	Data structure and transmission rate dependent (Table 1-5)	

Read Data (DIA)

Context: Second instruction in 2-instruction sequence (DOB/DIA).
BCC Accumulation Register



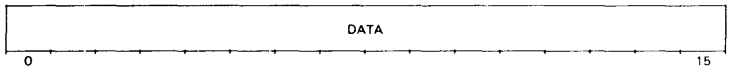
PCI Register and Modem Status Register*



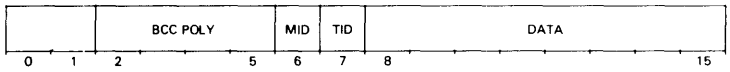
**The modem status register contains the status of the CTS (Clear to Send) and RI (Ring Indicator) modem control lines and is addressed only when the PCI status register is addressed.*

Write Data (DOA)

Context: Second instruction in 2-instruction sequence (DOB/DOA).
BCC Accumulation Register

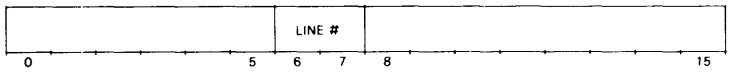


PCI and Other Registers*



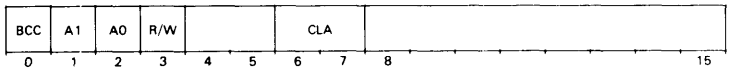
**Other registers refer to the BCC polynomial select register, addressed only when the PCI mode registers are addressed, and the interrupt disable registers, addressed only when the PCI command register is addressed.*

Read Interrupt Line (DIB)



Specify Comm Line and Operation (DOB)

Context: First instruction in 2-instruction sequence (DOB/DIA or DOB/DOA).



Bits					Bits		
0	**1**	**2**	**3**		**6**	**7**	
0	0	0	0	Read receiver holding register.	0	0	Line 0
0	0	0	1	Write transmitter holding register.	0	1	Line 1
0	0	1	0	Read status register.	1	0	Line 2
0	0	1	1	Write SYN1/SYN2/DLE registers.	1	1	Line 3
0	1	0	0	Read mode register 1, 2.			
0	1	0	1	Write mode register 1, 2.			
0	1	1	0	Read command register.			
0	1	1	1	Write command register.			
1	X	X	0	Receiver BCC accumulation (read/write).			
1	X	X	1	Transmitter BCC accumulation (read/write).			

DG-25838

Figure 1-1 Programming summary: accumulator formats

Table 1-3 Programming summary: START, CLEAR, PULSE, and IORST functions

f= S	Sets the Busy flag to 1 and the Done flag to 0. Enables interrupts when the Interrupt Disable flag is set to 0.
f= C	Sets both the Busy and Done flags to 0. Disables interrupts.
f= P	Sets the Done flag to 0 and causes the last data transferred to or from the USAM to be used in an error checking calculation (USAM-4 models only).
IORST	Sets the Busy and Done flags to 0 and resets the USAM by clearing all internal registers except the non-PCI modem status and interrupt line registers and the BCC accumulation register. Disables interrupts and sets the Interrupt Disable flag to 0.

Registers

Each USAM card has two sets of program-accessible registers. One set consists of eight registers that are shared by the communications lines. The other set consists of the nine internal registers of each PCI that governs an interface.

Shared Registers

The names and uses of the eight shared registers (external to the PCI) are:

- Current line address register - used to select the address of the line to be affected by the DIA, DOA, and IOPLS instructions. This is a 2-bit write-only register.
- Operation address register - used to determine which type of information - data, control, status, error checking, or parameter - is to be passed to or from the line specified in the current line address register. This is a 4-bit write-only register.
- Interrupt line register - used to hold the address of the highest-priority line requesting service. This is a 2-bit read-only register.
- Modem status register - used to hold the status of the modem signals Ring Indicator and Clear to Send for each of the communications lines. This is a 4 x 2-bit read-only register.
- Transmitter interrupt disable register - used to hold one interrupt disable bit for each of the four communications lines. This is a 4-bit write-only register.
- Modem change/transmitter empty interrupt disable register - used to hold one interrupt disable bit for each of the four communications lines. This is a 4-bit write-only register.
- BCC polynomial select register - used to determine which error checking polynomial - LRC-8, CRC-12, CRC-16, or CCITT-16 - is to be used to calculate a block check sum for the line. This is a 4 x 4-bit register and is write-only. (Not present on the USAM-1.)
- BCC accumulation register - used to hold the 16-bit accumulated block check sum for the receiver and transmitter sections of each of the four communications lines. This is an 8 x 16-bit, read/write register. (Not present on the USAM-1.)

PCI Registers

The names and primary uses of the nine 8-bit PCI registers are as follows:

- Mode register 1 - used to set up the PCI for asynchronous or (lines 0 and 1) synchronous operations and to define the following line characteristics: parity, character length, number of SYN characters and transparency mode (for synchronous operation), and baud rate factor. This is a read/write register.
- Mode register 2 - used to set up the PCI timing by choosing internal or external clocks for the receiver and transmitter and by choosing the baud rate. This is a read/write register.
- Command register - used to enable and disable transmitter and receiver sections and to activate certain modem control functions. This is a read/write register.
- Status register - used to monitor receiver and transmitter conditions and modem status. This is a read-only register.
- Transmitter (data) holding register - used to hold the character supplied by the CPU in parallel form until the transmitter shift register is ready to convert the data to the specified serial bit stream and send it over the communications line. This is a write-only register.
- Receiver (data) holding register - used to hold the character supplied by the receiver shift register in parallel form. The holding register latches the character and makes it available to the CPU until the next received character overwrites it. This is a read-only register.
- SYN1, SYN2, and DLE registers - used to hold the synchronization and transparency mode characters defined by the program for synchronous operation. These are write-only registers (lines 0 and 1 only).

Interrupts

The USAMs use a combination of Busy and Done logic. If the Interrupt Disable flag is set to 0, interrupts are requested when Done is set to 1. The Done flag is set to 1 if one of the following interrupt conditions occurs while the USAM Busy flag is 1:

Receiver Ready — One of the receiver data holding registers has a character ready for the CPU.

Transmitter Ready — One of the transmitter holding registers is ready to accept a character from the CPU.

Modem Status Change/Transmitter Empty — The transmitter holding register for one of the transmitters is empty, modem signal Data Set Ready or Carrier Detect has changed state, or the modem signal Ring Indicator (if enabled by switch) is asserted.

The transmitter ready and modem change/transmitter empty interrupts can be disabled by setting the appropriate bits in their respective interrupt disable registers.

Priorities

The USAM uses the following priority scheme to resolve conflicts among sections (transmitter/receiver/modem) requesting service at the same time.

- Each line is assigned a relative priority based on its address. Line 0 has the highest priority and line 3 the lowest.
- Receiver sections have priority over any other sections.
- Modem changes and transmitters have equal priority.

The priorities are implemented by a priority encoder, which sets the system Done flag to 1 when it receives an interrupt from any section. The priority encoder will then respond to a Read Interrupt Line instruction DIB by supplying the address of the highest priority line requesting service.

Instruction Set

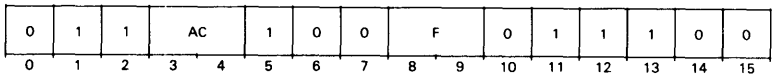
The communications interfaces respond to four input/output (I/O) instructions. One selects the communications line and the operation to be performed for that line. After this instruction is executed, two instructions are available for the actual transfer of data between the selected register (or registers) and a specified CPU accumulator. The last instruction of the four transfers the contents of the interrupt line register to a specified CPU accumulator.

The mnemonics and device codes as well as the effects of the device flag commands and I/O Reset instruction (IORST) are defined in the programming summary.

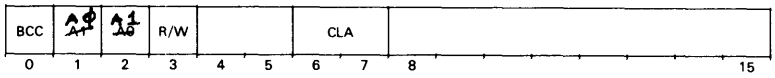
Specify Comm Line and Operation

DOB*[f]* *ac*,ASLM

Context: First instruction in 2-instruction sequence (DOB/DIA or DOB/DOA).



Loads the contents of the specified accumulator into the current line address register and the operation address register. The contents of the specified accumulator remain unchanged. After the data transfer, performs the function specified by f. The format of the specified accumulator is:



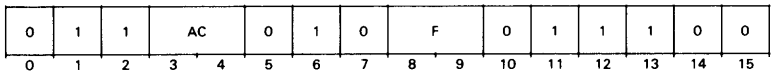
Bits	Name	Contents or Function
0	Block Check	
1	Address 0	
2	Address 1	
3	Read/Write	Bits 0 1 2 3 0 0 0 0 = Read receiver holding register 0 0 0 1 = Write transmitter holding register 0 0 1 0 = Read PCI status register and modem status register 0 0 1 1 = Write SYN1/SYN2/DLE registers 0 1 0 0 = Read mode 1/2 registers 0 1 0 1 = Write mode 1/2 registers and BCC polynomial register 0 1 1 0 = Read command register 0 1 1 1 = Write command register, transmitter interrupt disable register, and modem change/transmitter empty interrupt disable register 1 X X 0 = Read or write BCC accumulation * register - receiver section 1 X X 1 = Read or write BCC accumulation * register - transmitter section
4-5	—	Reserved for future use.
6-7	Current Line Address	Bits 6 7 0 0 = Line 0 0 1 = Line 1 1 0 = Line 2 1 1 = Line 3
8-15	—	Reserved for future use.

*An X indicates that the bit can be either 0 or 1 - a "don't care" bit.

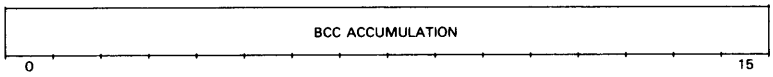
Write Data

DOA*[f]* *ac,*ASLM

Context: Second instruction in 2-instruction sequence (DOB/DOA).

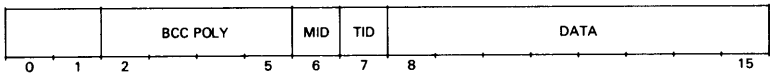


Loads the contents of the specified accumulator into the register(s) specified by the preceding Specify Line and Operation instruction (DOB). The contents of the accumulator remain unchanged. After the data transfer, performs the function specified by f. The format of the specified accumulator for the BCC accumulation register is:



Bits	Name	Contents or Function
0-15	BCC	The block check accumulation to be loaded into the BCC accumulation register for the specified line

The format of the specified accumulator for PCI registers and shared registers is:



Bits	Name	Contents or Function
0-1		Reserved for future use.
2-5[1]	BCC polynomial select	If either PCI mode register is addressed, bits 2-5 contain the contents of the BCC polynomial register for the current line: Bits 2 3 4 5 0 0 0 1 = LRC-8 0 0 1 0 = CRC-12 0 1 0 0 = CRC-16 1 0 0 0 = CCITT-16
6[2]	Modem Change/ Transmitter Empty Interrupt Disable	If the PCI command register is addressed, bit 6 contains the contents of the transmitter empty interrupt disable register for the current line: 0 = interrupt enabled 1 = interrupt disabled
7[2]	Transmitter Interrupt Disable	If the PCI command register is addressed, bit 7 contains the contents of the transmitter interrupt disable register for the current line: 0 = interrupt enabled 1 = interrupt disabled
8-15	Data	The contents of the specified PCI register; if the specified register is the transmitter holding register, the character should be right justified and any unused bits are set to 0 (see contents of PCI Registers)

[1]*Loaded at the same time that either PCI mode register is loaded; otherwise, these bits are reserved for future use.*

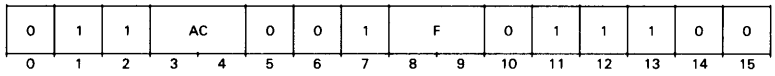
[2]*Loaded at the same time that the PCI command register is loaded; otherwise, these bits are reserved for future use.*

NOTE *If the BCC polynomial register is loaded with a combination of bits other than those listed above, the results of any BCC calculations will be indeterminate.*

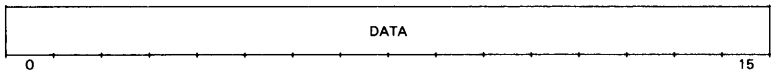
Read Data

DIA*[f]* *ac*,ASLM

Context: Second instruction in 2-instruction sequence (DOB/DIA).

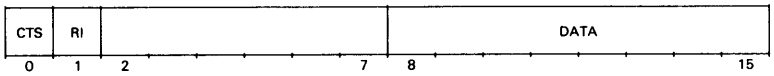


Loads the contents of the register specified by the preceding Specify Comm Line and Operation instruction (DOB) into the specified accumulator. The previous contents of the accumulator are lost. After the data transfer, performs the function specified by f. The format of the specified accumulator for the BCC accumulation register is:



Bits	Name	Contents or Function
0-15	BCC Accumulation	The contents of the specified BCC accumulation register

The format of the specified accumulator for the PCI registers is:

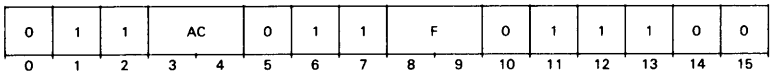


Bits	Name	Contents or Function
0-1*	—	The contents of the modem status register, if specified, for the current line:
0*	Clear to Send	0 = Clear to Send not asserted 1 = Clear to Send asserted
1*	Ring Indicator	0 = Ring Indicator not asserted 1 = Ring Indicator asserted
2-7	—	Reserved for future use All bits are set to 1
8-15	Data	The 8-bit control word or character; if reading the receiver holding register, the character is right justified and unused bits are set to 0 (see contents of PCI registers)

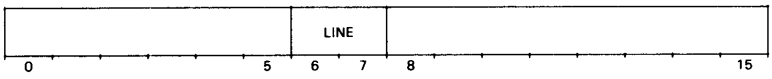
* *These bits are set to 1 if the modem status register is not specified.*

Read Interrupt Line

DIB*[f]* *ac*,ASLM



Loads the 2-bit contents of the interrupt line register into the specified accumulator. The previous contents of the accumulator are lost. After the data transfer, performs the function specified by f. The format of the specified accumulator is:

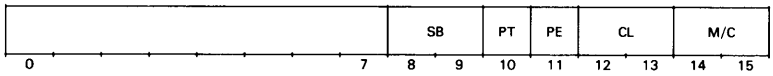


Bits	Name	Contents or Function
0-5	—	Reserved for future use; all bits are set to 1
6-7	Line number	Address of highest-priority line requesting service: Bits 6 7 0 0 = Line 0 0 1 = Line 1 1 0 = Line 2 1 1 = Line 3
8-15	—	Reserved for future use All bits are set to 1

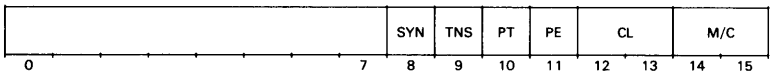
Contents of the PCI Registers

A description of the contents and accumulator formats of the nine PCI registers referred to in the instruction set follows:

Mode Register 1 (Asynchronous Mode)



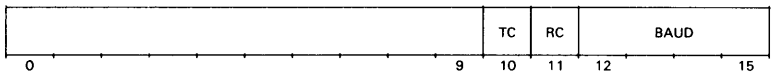
Mode Register 1 (Synchronous Mode)



Bits	Name	Contents or Function
0-7	—	Reserved for shared registers external to PCI
8-9	Stop Bits (Async)	Bits 8 9 0 0 = Invalid 0 1 = 1 stop bit 1 0 = 1 1/2 stop bits 1 1 = 2 stop bits
8	Number of SYNs (Sync)	0 = 2 SYN characters 1 = 1 SYN character (lines 0 and 1)
9	Transparency (Sync)	0 = Nontransparent mode 1 = Transparent mode
10	Parity Type	0 = Odd 1 = Even
11	Parity Enable	0 = Parity disabled 1 = Parity enabled
12-13	Character Length	Bits 12 13 0 0 = 5 bits 0 1 = 6 bits 1 0 = 7 bits 1 1 = 8 bits
14-15	Mode and Baud Rate Factor	Bits 14 15 0 0 = Synchronous, 1x clock rate (lines 0, 1) 0 1 = Asynchronous, 1x clock rate* 1 0 = Asynchronous, 16x clock rate 1 1 = Asynchronous, 64x clock rate

** Baud rate factor in asynchronous mode applies only when external clocks are selected. The factor is 16x clock rate when internal clocks are selected.*

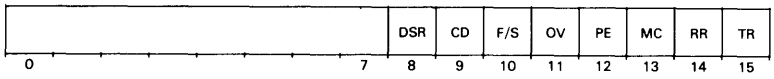
Mode Register 2



Bits	Name	Contents or Function
0-7	—	Reserved for shared registers external to PCI
8-9	—	Reserved for future use
10	Transmitter Clock	0 = External (lines 0 and 1 only) 1 = Internal (Programmable)
11	Receiver Clock	0 = External (lines 0 and 1 only) 1 = Internal (Programmable)
12-15	Baud Rate	Programmable baud rates when bit 10 and/or bit 11 are set to 1:

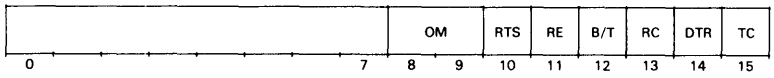
Bits 12	13	14	15	Bits/second (baud)
0	0	0	0	50
0	0	0	1	75
0	0	1	0	110
0	0	1	1	134.5
0	1	0	0	150
0	1	0	1	300
0	1	1	0	600
0	1	1	1	1200
1	0	0	0	1800
1	0	0	1	2000
1	0	1	0	2400
1	0	1	1	3600
1	1	0	0	4800
1	1	0	1	7200
1	1	1	0	9600
1	1	1	1	19200

Status Register



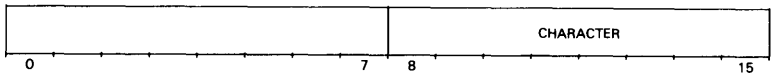
Bits	Name	Meaning when 1
0-7	—	Reserved for shared registers external to PCI
8	Data Set Ready	Data Set Ready asserted
9	Carrier Detect	Carrier Detect asserted
10	Framing Error (Async)	No valid stop bit detected in last character received
10	SYN Detect (Sync)	SYN character was received. (Lines 0 and 1 only)
11	Overrun	Lost data; character in receiver holding register was overwritten before it was retrieved by CPU
12	Parity Error (Async)	Parity error detected in last character received
12	Parity Error (Sync)	Parity error detected in last character received (with parity enabled); or, in transparent mode (with parity disabled), a DLE character was received. (Lines 0 and 1 only)
13	Modem Change/Transmitter Empty	Data Set Ready or Carrier Detect changed state, and/or transmitter shift register empty
14	Receiver Ready	Receiver holding register has a character available to the program
15	Transmitter Ready	Transmitter holding register is ready to receive a character from the program

Command Register



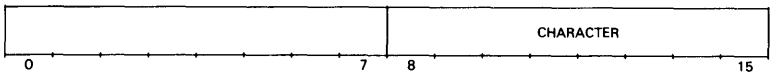
Bits	Name	Contents or Function
0-7	—	Reserved for shared registers external to PCI
8-9	Operating Mode	Bits 8 9 0 0 = Normal operation 0 1 = Asynchronous: echo mode Synchronous: SYN and/or DLE stripping mode (lines 0 and 1 only) 1 0 = Local loop back 1 1 = Remote loop back
10	Request to Send	1 = Assert Request to Send
11	Reset	1 = Reset error bits in status register (framing, overrun, and parity) and DLE character detect bit
12	Force Break (Async)	1 = At the end of the current character transmission, the transmit line is placed in the spacing state until bit 12 is set to 0
12	Transmit DLE (Sync)	1 = Transmit contents of DLE register prior to transmitting character in transmitter holding register. (Lines 0 and 1 only)
13	Receive Control	0 = Disable receiver 1 = Enable receiver
14	Data Terminal Ready	1 = Assert Data Terminal Ready
15	Transmit Control	0 = Disable transmitter 1 = Enable transmitter

Transmitter and Receiver Holding Registers



Bits	Name	Contents or Function
0-7	—	Reserved for future use
8-15	Character	5- to 8-bit character, right justified

SYN1, SYN2, DLE Registers (lines 0 and 1 only)



Bits	Name	Contents or Function
0-7	—	Reserved for future use
8-15	Character	SYN1, SYN2, or DLE character, as applicable to appropriate register

Programming Considerations

Programming each communications interface proceeds in six steps:

- Set up the interface for asynchronous or synchronous operations;
- Manipulate and monitor the modem control lines, when applicable;
- Set up the error checking routine for synchronous operation, when applicable;
- Enable the transmitter and/or receiver and transfer data;
- Service interrupt requests;
- Monitor status conditions.

To implement these steps, the program must address the PCI registers and the shared registers external to the PCI and must transfer information between these shared registers and one or more of the CPU accumulators. Each information transfer, except reading the interrupting line number, requires two I/O instructions that are executed in sequence — Specify Comm Line and Operation (DOB) followed either by a Write Data instruction (DOA) or by a Read Data instruction (DIA).

NOTE *Any number of nonrelated instructions, except for I/O reset (IORST), may be executed between the two instructions of the sequence.*

The first instruction of the sequence, Specify Comm Line and Operation (DOB), loads the operation address register. This register, which is shared by the four communications interfaces, performs the following functions:

- It selects the interface — 0, 1, 2, or 3;
- It addresses either the BCC accumulation register (USAM-4 only) or a particular register within the PCI of the selected interface;
- In some cases, it also addresses one or more shared registers whose contents are loaded along with the chosen PCI register;
- It prepares the registers for a read or write operation when the appropriate second instruction in the sequence is executed.

Table 1-4 lists the USAM registers and specifies their read/write capabilities. Notice in particular the following features of the registers:

1. Some registers are read only or write only; an instruction sequence that addresses a read-only register and specifies a write operation will result in incorrect register addressing. The second instruction in a 2-instruction sequence must move data in the direction specified by the first instruction. Otherwise, the data read or written is indeterminate.
2. Some registers share the same address. There are two categories of such registers; PCI registers that share the same address, and registers external to the PCI that have the same address as a PCI register.

PCI Registers with Shared Addresses

Mode registers 1 and 2 share address 10 binary, and the SYN1, SYN2, and DLE registers share address 01 binary. These registers are addressed sequentially - that is, successive transfers to the given address will result in transfers to the registers sharing that address one after another. For example, if the first instruction of a sequence specifies a mode register, then the mode register addressed is mode register 1; the next transfer to that address will reference mode register 2, the next will reference mode register 1, and so on. Similarly, the first instruction that specifies the SYN1/SYN2/DLE registers addresses the SYN1 register; the next, the SYN2 register; the next, the DLE register; the next, the SYN1 register; and so on. An I/O Reset instruction (IORST) or a read PCI command register instruction sequence resets the pointers for the mode and SYN registers - that is, it causes the first register to be addressed thereafter as the mode 1 register or the SYN1 register.

Non-PCI Registers Sharing PCI Register Addresses

The following registers external to the PCIs share addresses with PCI registers.

- The BCC polynomial select register has the same address as the PCI mode registers.
- The modem status register has the same address as the PCI status register.
- The transmitter interrupt disable register and the modem change/transmitter empty interrupt disable registers have the same address as the PCI command register.

The nonPCI registers are loaded at the same time as the PCI registers that have the same addresses; the PCI registers are loaded by bits 8-15 of the specified accumulator and the nonPCI registers are loaded by a subset of bits 0-7.

Note in particular that the BCC polynomial select register has the same address as both mode registers. This means that the BCC polynomial select bits must be loaded with the last set of mode register bits; that is, loading the mode registers without specifying the contents of the BCC polynomial register will result in indeterminate data being passed to the BCC polynomial register.

Table 1-4 USAM registers

Register Names	Register Address BCC	A1	A0	Read/Write	Read/Write Capability
Receiver holding register (PCI)	0	0	0	0	Read only
Transmitter holding register (PCI)	0	0	0	1	Write only
Status register (PCI) and modem status register (nonPCI) [1]	0	0	1	0	Read only
SYN1 register (PCI)	0	0	1	1	Write only
SYN2 register (PCI)	0	0	1	1	Write only
DLE register (PCI)	0	0	1	1	Write only
Mode register 1 (PCI)	0	1	0	0/1	Read and write
Mode register 2 (PCI)	0	1	0	0/1	Read and write
BCC polynomial select register (nonPCI)[2]	0	1	0	0	Write only
Command register	0	1	1	0/1	Read and write
Transmitter interrupt disable register (nonPCI) [3]	0	1	1	1	Write only
Modem change/transmitter empty interrupt disable register (nonPCI) [3]	0	1	1	1	Write only
BCC accumulation register - receiver section [4]	1	X	X	0	Read/write
BCC accumulation register - transmitter section [4]	1	X	X	1	Read/write

[1] *The modem status register is automatically read when the PCI status register is read.*

[2] *The BCC polynomial select register is automatically written into when either PCI mode register is loaded.*

[3] *The transmitter interrupt disable register is automatically written into when the PCI command register is loaded.*

[4] *The read/write bit does not determine for these registers the direction of transfer; instead, this bit selects the section (transmitter or receiver). The direction of transfer is determined by the instruction issued (DOA or DIA).*

Setting Up the Interface

Before data communications can begin, the program must define the PCI's operational mode by setting the appropriate bits in both the mode and command registers. When synchronous mode is selected, the SYN1 register, and optionally the SYN2 and DLE registers, must be loaded; the BCC polynomial and the initial BCC accumulation must be loaded if error checking is to be performed (USAM-4).

Because the interpretation of some PCI command fields depends upon the mode of operation selected, asynchronous and synchronous setup information is explained separately below.

NOTE *To interpret the fields of the PCI registers, refer to the contents of PCI Registers section.*

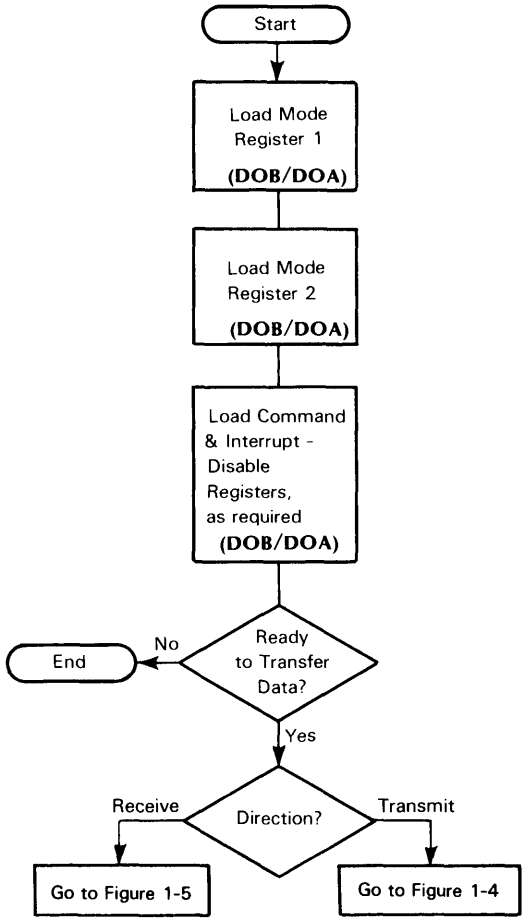


Figure 1-2 Asynchronous mode setup

Asynchronous Mode

Figure 1-2 is a flowchart of events required to set up a PCI for asynchronous operations. It involves three steps:

- Load mode register 1 to select the data structure (5 to 8 data bits; 1, 1 1/2, or 2 stop bits; optional parity bit), asynchronous mode, and clock rate factor. If programmable baud rates will be used, select 16x the asynchronous clock rate.
- Load mode register 2 to select internal or external clocks. If internal clocks are selected, choose the baud rate (50 to 19,200 bits per second).
- Load the command register to select type of operation (normal, echo, local loop back, or remote loop back) and to activate the Data Terminal Ready signal, as required; and, in the same step, load the interrupt disable registers to determine whether transmitter interrupts are to be disabled and whether modem change/transmitter empty interrupts are to be disabled.

This manual assumes the use of the normal mode; that is, the transmitter and receiver operate independently while sharing the line characteristics defined by the contents of the mode registers. For information about other operating modes, refer to the Signetics data sheet for the 2661 programmable communications interface (PCI) circuit. The Signetics data sheet should also be referenced when an external clock source (e.g., modem clocks) will be used.

Notice that in the asynchronous mode the PCI automatically inserts 1 start bit at the beginning of each character bit stream.

Synchronous Mode

These steps apply only to lines 0 and 1. Figure 1-3 is a flowchart of events required to set up the PCI and the error checking logic for synchronous operations on one line. It involves four steps (five, if error checking is performed):

- Load mode register 1 to select the data structure (5 to 8 data bits and optional parity bit), synchronous mode, number of synchronization characters (1 or 2), and transparent or normal mode.
- Load mode register 2 and the BCC polynomial select register to select the internal clocks and to choose the baud rate (50 to 19,200 bits per second), and to select the BCC polynomial to be used for error checking, when used. If external clocks will be used, mode register 2 selects external clocks by default and this step can be eliminated. However, when this step is eliminated, the BCC polynomial selection must be made when mode register 1 is addressed.
- Load synchronization character register(s). Load the SYN1 register to select the first, or only, synchronization character; load the SYN2 register to select the second synchronization character, when used; load the DLE register to select the transparency mode character, when used.
- Load the command register to select type of operation (normal, automatic SYN/DLE stripping, local loop back, or remote loop back) and to activate the Data Terminal Ready signal, as required; and, in the same step, load the interrupt disable registers to determine whether transmitter interrupts are disabled and whether modem change/transmitter empty interrupts are disabled.
- Load the BCC accumulation register with the initial value of the BCC accumulation for the receiver and for the transmitter, when used.

The number of synchronization characters selected in mode register 1 has the following effect: it determines the number of synchronization characters that will be used to establish synchronization and provide character fill when transparent mode is selected. When transparent mode is selected, the same number of synchronization characters will be used to establish synchronization, but DLE-SYN1 characters will be used to provide character fill.

This manual assumes the use of either normal or automatic SYN/DLE stripping modes. For information about the other operating modes, refer to the Signetics data sheet for the 2661 programmable communications interface (PCI) circuit. The Signetics data sheet should also be referenced when an external clock source (e.g., modem clocks) will be used.

In normal mode, the transmitter and receiver operate independently while sharing the line characteristics defined by the contents of the mode registers.

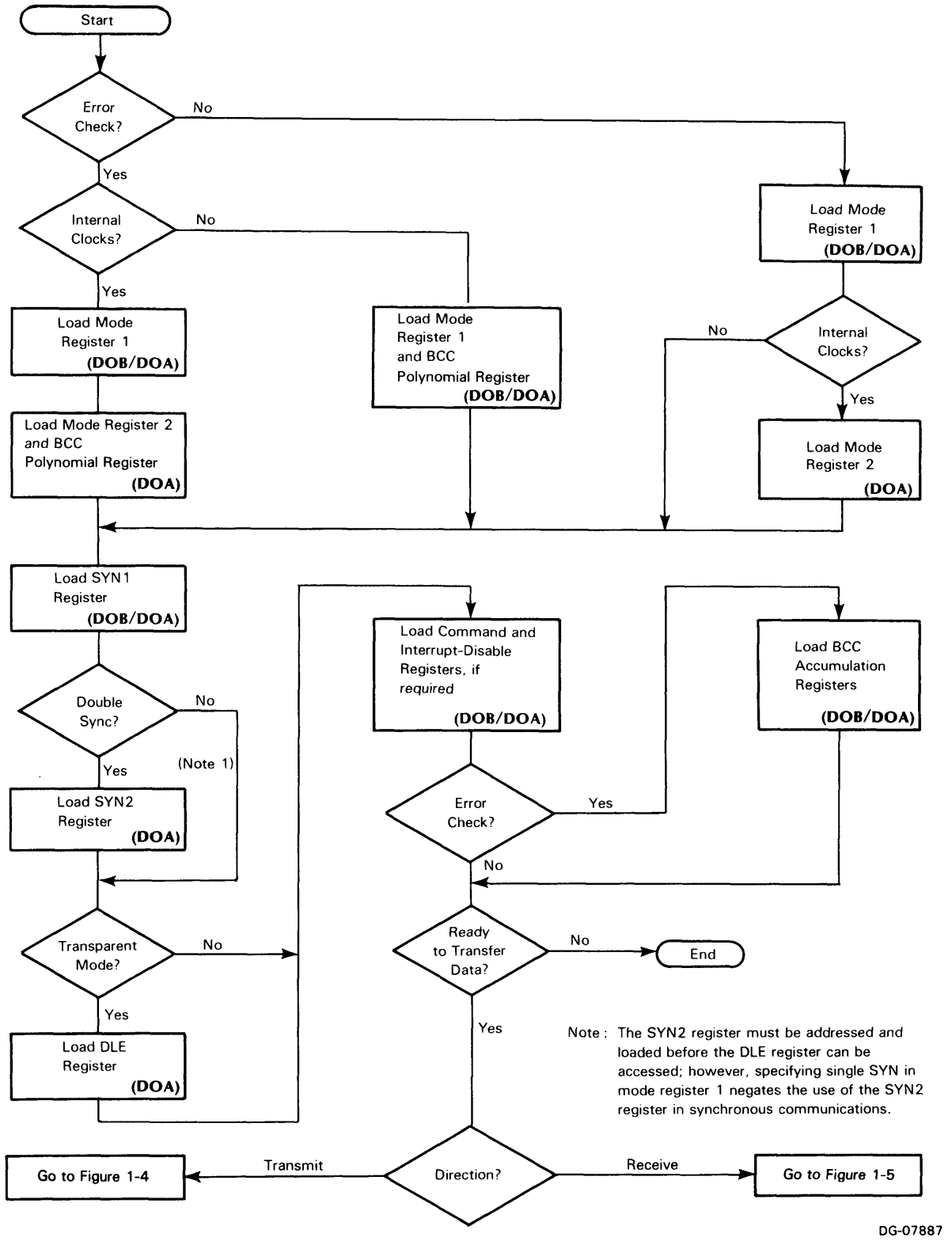


Figure 1-3 Synchronous mode setup

In the automatic SYN/DLE stripping mode, the transmitter and receiver operate independently, and a character stripping function is performed by the PCI on incoming data. The character(s) stripped depends upon the selection of the number of synchronization characters and normal/transparent mode as follows:

- If single SYN and nontransparent mode are specified in mode register 1, characters received from the transmission line that match the contents of the SYN1 register are not transferred to the receiver holding register. Thus, these characters are stripped from the data stream.
- If double SYN and nontransparent mode are specified in mode register 1, characters received from the transmission line that match the contents of the SYN1 and SYN2 registers and that appear in the appropriate sequence (that is, SYN1-SYN2) are not transferred to the receiver holding register. However, only the first SYN1 character of a SYN1-SYN1 sequence is stripped. Also, SYN2 characters that are not preceded by a SYN1 character are not stripped.
- If transparent mode is specified in mode register 1, characters received from the transmission line that match the contents of the DLE register (and the SYN1 register, if immediately preceded by a DLE) are not transferred to the receiver holding register. However, only the first DLE character of a DLE-DLE sequence is stripped.

Notice that the automatic stripping mode does not affect the setting of the SYN Detect and DLE Detect bits in the PCI status register.

Setup Change

The operating mode of the 0 and 1 lines in the USAM can be changed at any time by the program. Before implementing a setup change, disable the transmitter and receiver of the appropriate line by setting its Transmit Control and Receiver Control bits in the command register to 0.

Controlling the Modem

Each communications interface supports six standard EIA modem control functions: Clear to Send, Request to Send, Data Terminal Ready, Data Set Ready, Carrier Detect, and Ring Indicator. All but the Ring Indicator line are PCI related. Circuitry external to the PCI tracks the state of modem signals Clear to Send and Ring Indicator, and this circuitry can be configured (switch selected) to allow Ring Indicator to initiate interrupt requests.

The PCI uses two modem control functions — Clear to Send and Carrier Detect — as conditions for transmitting and receiving. When modem is not present, the hardware asserts them automatically.

The state of two modem control lines — Request to Send and Data Terminal Ready — are under direct program control. These lines can be activated/ deactivated by manipulating the Request to Send and Data Terminal Ready bits in the PCI command register. The use of these control lines, as well as the Data Set Ready line, is modem dependent. Consult your modem manual for specific information.

Transmitting Data

Before transmission can begin, the Clear to Send line must be asserted and the program must enable the transmitter by setting the Transmit Control bit in the command register to 1. When these conditions are present, the Transmitter Ready status bit is set to 1, which in turn sets the Done flag to 1 (if the Busy flag has been set to 1 initially). The Done condition initiates an interrupt request when interrupts are enabled.

The program responds to the interrupt by reading the interrupt line register and clearing the Done condition with a Read Interrupt Line instruction (DIBC) accompanied by a Clear command. After reading the PCI status register and determining that it was the transmitter section that caused the interrupt, it sends a character to the transmitter holding register. If a USAM-4 is being used in the synchronous mode with error checking, the program indicates at the same time (with an I/O Pulse command) whether or not the character is to be included in the error checking calculation. Sending a character to the transmitter holding register sets the Transmitter Ready bit to 0 and, if error checking was specified and a P command issued, causes the hardware to calculate a new BCC accumulation at the same time that the character is being transmitted.

As the PCI moves the character from the holding register to the transmitter shift register, the Transmitter Ready bit is set to 1 again and initiates an interrupt request (if Busy has been set to 1). If the program fails to respond to the interrupt by sending another character to the holding register before the shift register completes the previous character transmission, the Transmitter (Shift Register) Empty status bit is set to 1.

Figure 1-4 shows the sequence of events required to transmit data in both asynchronous and synchronous modes with interrupts enabled. When the interrupt facility is used, the program determines which communications line requested an interrupt by issuing a Read Line instruction. Once the line address is established, the program must issue a Specify Comm Line and Operation/Read Data instruction sequence (DOB/DIA) specifying the PCI status register to determine whether the interrupt was caused by the receiver, transmitter, or modem.

When the interrupt facility is not used, the program can check the status of a transmitter in any communications interface by issuing a DOB/DIA instruction sequence specifying the PCI status register of that interface.

When operating in synchronous mode, the PCI does not automatically transmit synchronization (or idle fill) characters until the program loads the first character into the holding register. Hence, the first character sent by the program to the transmit holding register is normally a synchronization character. After the first character is transmitted, the PCI automatically inserts the appropriate idle fill character(s) each time the transmitter is idle, that is, when the program fails to load the holding register before the current character transmission is completed by the shift register. None of these idle fill characters are included in the BCC accumulation.

After the last character in a synchronous transmission block is transmitted, the block check accumulation must be read into a CPU accumulator with a DOB/DIA instruction sequence specifying the BCC accumulation register if error checking is being performed. Then the program can transmit the 16-bit block check character one byte at a time.

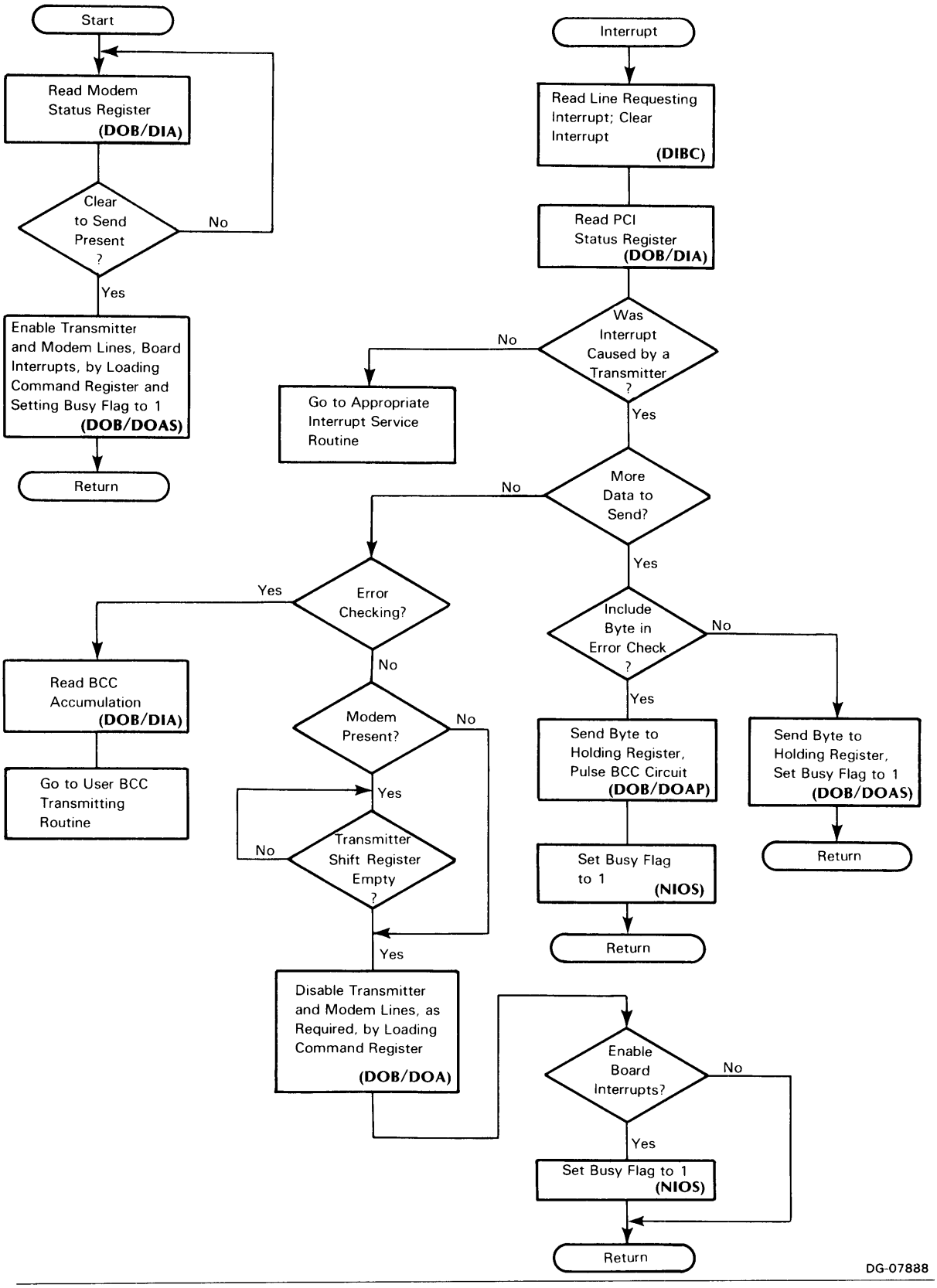


Figure 1-4 Transmit data with interrupts enabled

When terminating transmission and operating without a modem, the program should insure that the Transmitter Ready status bit is set to 1 before it disables the transmitter by setting the Transmit Control bit to 0.

When terminating transmission and operating with a modem, the program should insure that the Transmitter (Shift Register) Empty status bit is set to 1 before it disables the transmitter and deactivates the modem control lines. While disabling the transmitter before the shift register is empty will not affect current character transmission, deactivating the control lines may affect character reception by the modem.

Receiving Data

Before data reception can begin, the Carrier Detect line must be asserted and the program must enable the receiver by setting the Receive Control bit in the command register to 1. When these conditions are present and the receiver shift register moves a byte into the receiver holding register, the Receiver Ready status bit is set to 1, thus setting the Done flag to 1 (if the Busy flag was 1 to begin). The Done condition initiates an interrupt request when interrupts are enabled.

NOTE *In synchronous mode, reception of the synchronization character(s) sets the SYN Detect bit in the PCI status register, but the SYN character(s) used to establish synchronization is never placed in the receiver holding register.*

The program responds to the interrupt by reading the interrupt line register. After determining that the interrupt was caused by the receiver section, the program reads the byte in the receiver holding register. At the same time, if error checking is being performed, the program can issue a Pulse command to include the byte in the BCC accumulation. Reading the byte in the receiver register sets the Receiver Ready bit to 0.

Figure 1-5 shows the sequence of events required to receive data in both asynchronous and synchronous modes with interrupts enabled. If the interrupt facility is not used, the program can monitor the state of the receiver by issuing a DOB/DIA instruction sequence specifying the PCI status register of that interface.

To terminate character reception, disable the receiver by setting the Receive Control bit in the command register to 0.

Servicing Interrupts

In general, the program services interrupts in accordance with the following priority scheme: receiver interrupts first, then transmitter interrupts or modem change/transmitter empty interrupts. Failure to service a receiver interrupt promptly can result in data loss, while failure to service a transmitter interrupt results in inefficient use of the transmission line.

To clear an interrupt:

1. Issue instructions that remove the interrupt condition.
2. Reset the Done flag to 0.

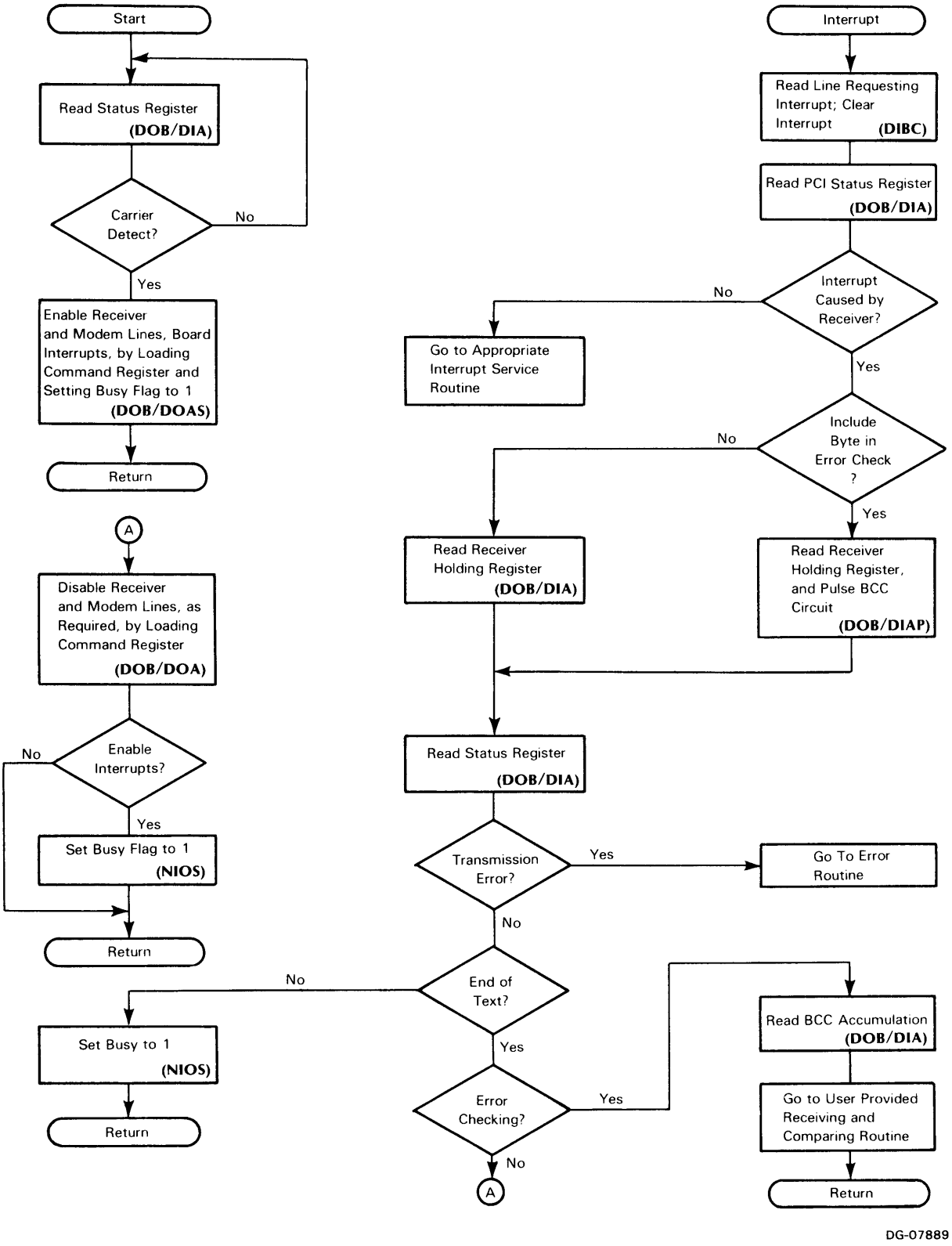


Figure 1-5 Receive data (with interrupts enabled)

Interrupt conditions are cleared as follows:

Receiver Ready — Cleared by reading the receiver holding register or by disabling the receiver by setting the Receive Control bit in the command register to 0.

Transmitter Ready — Cleared by sending a byte to the transmitter holding register or by disabling the transmitter by setting the Transmit Control bit in the command register to 0.

Modem Change/Transmitter Empty — If caused by a modem change (Data Set Ready or Carrier Detect),* cleared by reading the contents of the PCI status register. If caused by a transmitter (shift register) empty, cleared by sending a character to the transmitter holding register or by disabling the transmitter as above.

NOTE *In the synchronous mode, the automatic transmission of idle fill characters has no effect on the initiation of transmitter interrupts.*

CAUTION *If the Ring Indicator (RI) signal is not being used, disable the RI signal with switch SW1.*

Reading Status Information

Reading the PCI status register and the nonPCI modem status register provides the program with the following information:

- Transmitter and receiver ready conditions;
- The states of Carrier Detect, Data Set Ready, Clear to Send, and Ring Indicator modem control lines;
- Synchronization and transparency mode character detection;
- The presence of transmission errors in received data.

Transmitter/receiver ready conditions and a change in the state of a modem or a transmitter holding register empty conditions set the Done flag to 1 and initiate interrupt requests when interrupts are enabled. The status of these conditions can be determined by issuing a Specify Comm Line and Operation/Read Data (DOB/DOA) instruction sequence specifying the PCI status register.

In synchronous mode, the states of the SYN Detect bit and the DLE Detect bit in the status register provide the program with the following information.

In the synchronous transparent mode, the SYN Detect bit is set to 1 upon detection of the initial synchronization characters (SYN1 or SYN1-SYN2), and after synchronization is achieved, on reception of DLE-SYN1. In the synchronous nontransparent mode, the SYN Detect bit is set to 1 on reception of the SYN1-SYN2 pair in double-SYN mode. This bit is automatically set to 0 after the program reads the PCI status register or when the program sets the Receive Control bit in the command register to 0.

The DLE Detect bit is set to 1 when a character has been received in synchronous transparent mode that matches that in the DLE Register, if parity is disabled. (If parity is enabled, this bit indicates a received parity error when set.)

In the asynchronous mode, the 1 state of either the Parity Error (when parity has been enabled) or Framing Error bits in the status register indicates a transmission error in the last character received.

In either mode, the 1 state of the Overrun bit indicates data loss. The program should check the state of the error detecting bits after reading a character from the receiver holding register. After checking these bits, the program can force all of them to the 0 state by setting the Reset Error bit in the command register to 1.

Timing

After the Receiver Ready status bit is set to 1, the character in the receiver holding register is available to the program for a time interval determined by two factors: the transmission (baud) rate and the number of bits per character. To avoid possible data loss, the program must respond to a receiver interrupt request by reading the character within a time interval that is computed as:
(1/baud rate) x (bits/character)
The time intervals tabulated in Table 1-5 are based on the assumption that characters transmitted at 50 to 110 baud contain 11 bits (including 2 stop bits), and that characters transmitted at 134.5 to 19,200 baud contain 10 bits (including 1 stop bit).

Table 1-5 Timing

Transmission Rate (baud)	Maximum Allowable Programmed I/O Latency (ms)
50	219.00
75	146.00
110	100.00
134.5	74.35
150	66.66
300	33.33
600	16.66
1200	8.33
1800	5.55
2400	4.16
4800	2.08
9600	1.04
19,200	0.52

After the Transmitter Ready status bit is set to 1, the program should provide another character within the time period indicated above to maintain the maximum transmission rate.

Power-Up Response

After power up or when a system reset is performed, the Busy and Done flags are set to 0, the Interrupt Disable flag is set to 0, and all PCI and nonPCI registers are reset, except for the BCC accumulation register and the nonPCI modem status and interrupt line registers.

The PCIs remain in a reset state (idle) until a Specify Comm Line and Operation instruction (DOB) is issued.

Electrical Interfacing

The USAM card receives power and communicates with its host CPU via the card's "B" connector which plugs into the backpanel. Communication with its peripheral devices is provided by cable which connects to the card's "A" connector. Tables 1-6 and 1-7 show the pin assignments for the "A" and "B" connectors. Power requirements are listed in Table 1-1, Technical Specifications.

Installation/Configuration

For installation and jumpering information, refer to the appropriate installation documentation for the associated systems.

Table 1-6 A connector pin assignments

Even	Functional Signal Names	Odd
2	TDORTN	TDATA0	1
4	RDORTN	RDATA0	3
6	CTS0	GND	5
8	DCD0	RTS0	7
10	DSR0	DTR0	9
12	TETOUT0	RI0	11
14	RET0	TETIN0	13
16	TD1RTN	TDATA1	15
18	RD1RTN	RDATA1	17
20	CTS1	GND	19
22	DCD1	RTS1	21
24	DSR1	DTR1	23
26	TETOUT1	RI1	25
28	RET1	TETIN1	27
30	TD2RTN	TDATA2	29
32	RD2RTN	RDATA2	31
34	CTS2	GND	33
36	DCD2	RTS2	35
38	DSR2	DTR2	37
40	TDATA3	RI2	39
42	RDATA3	TD3RTN	41
44	GND	RD3RTN	43
46	RTS3	CTS3	45
48	DTR3	DCD3	47
50	RI3	DSR3	49

Table 1-7 B connector pin assignments

Even	Functional Signal Names	Odd
2	MCLOCK*	MCLOCK	1
4	BI/ODATA1	GND	3
6	CLEAR*	BI/ODATA1*	5
8	BEXTINT*		7
10			9
12	BI/ODATA2*	GND	11
14	GND	BI/ODATA2	13
16	BI/OCLOCK	BI/OCLOCK*	15
18			17
20	INTP IN	INTP OUT	19
22	DCHP IN	DCHP OUT	21
24			23
26			25
28			27
30			29
32			31
34			33
36	GND		35
38			37
40		−12V	39
42			41
44			43
46			45
48			47
50			49
52			51
54	GND	GND	53
56	+12V	+12V	55
58	−5V	+5V	57
60	+5V	+5V	59

*Asterisks indicate negated signals

Theory of Operation

The following theory of operation describes the USAM-4 models. Operation of the USAM-1 is identical to that of the USAM-4 except for the reduced number of lines and the absence of hardware block check (CRC) functions. The USAM-4 supports four independent communications lines in full-duplex mode. The card interprets serial data sent from the CPU via the microI/O bus as I/O instructions, formats the data to be transmitted to conform with one of several program-selectable protocols, and transmits the data in serial form via the communications lines. The card can simultaneously receive serial data from the communications lines, reformat it under program control and place it on the microI/O bus for transfer to the CPU, and monitor and report the status of the received data to the CPU. In addition, the USAM-4 can perform error checking on data sent and received in synchronous mode.

The USAM has five major functional blocks, shown in Figure 1-6. (The USAM-1 has four, as it has no block check circuitry.) The blocks, which are discussed in detail below, are:

- The I/O bus interface and associated logic,
- The programmable communications interfaces (PCIs) and their associated circuitry,
- The control circuitry,
- The interrupt control circuitry,
- The block check circuitry.

I/O Bus Interface

The microI/O bus, sometimes called Microproducts I/O bus or microNOVA I/O bus interface contains a single I/O controller integrated circuit (IOC) and its support elements. The IOC converts serial data on the microI/O bus to parallel data, supplies a card timing signal ($\overline{\text{FSTROBE}}$), and contains interrupt control and request logic.

The interrupt logic in the interface IOC will be discussed as part of the interrupt control functional block. A description of the interface IOC appears in the Input/Output and Interfacing manual. For more detailed information, see also the microNOVA Integrated Circuits Data Manual.

Associated with the interface IOC are the instruction decoder and control logic, and the data bus driver. The decoder and control logic issues data bus control signals; card reset signals; and signals associated with the four I/O instructions, DOA, DIA, DOB, and DIB, and the three commands, Start, Clear, and Pulse.

The data bus driver drives data either from the USAM data bus into the IOC or from the IOC onto the USAM data bus; the decoder and control logic determines the direction of transfer. The USAM data bus is a 16-bit parallel bidirectional bus that carries all the data that are loaded into and read from the USAM's program accessible registers, including the PCIs.

The PCIs

The USAM-4 contains four programmable communications interface (PCI) chips. These large-scale integration (LSI) chips support several data communications disciplines, synchronous and asynchronous, in the full-duplex mode. The PCIs serialize parallel data characters received from the I/O interface via the USAM data bus for transmission. Simultaneously, they can receive serial data from the communications line and convert it into parallel characters for input to the CPU via the USAM data bus and I/O interface.

Each PCI consists of six major sections: transmitter, receiver, operation control, modem control, SYN/DLE control, and timing. These sections contain the nine program-accessible registers.

The operation control section contains the two mode registers, the command register, and the status register. The contents of the mode registers control the type of communication (synchronous or asynchronous), data structure, choice of parity, and baud rate. The contents of the command register control the operating mode, the Request to Send line, the Data Terminal Ready line, the receiver, and the transmitter. The program also uses bits in the command register to reset error flags in the PCI status register and to force an asynchronous break condition or a synchronous DLE stream. The contents of the status register indicate receiver, transmitter, and modem conditions.

Containing the transmitter holding and shift registers, the transmitter section accepts parallel data from the USAM bus, converts it to a serial data stream, inserts the appropriate characters or bits, and transmits a serial stream of data via the communications line. The transmitter section asserts the Transmitter Ready interrupt control signal and sets the Transmitter Ready status bit in the PCI status register when the transmitter is ready to receive a character.

The receiver section, which contains the receiver holding and shift registers, accepts data in serial form from the communications line, converts this data to parallel format, checks for bits or characters that are unique to the communication protocol, as well as for transmission errors, and sends an assembled character to the USAM data bus under program control. The receiver section asserts the Receiver Ready interrupt control signal and sets the Receiver Ready status bit in the PCI status register when the receiver has an assembled character ready to be read by the CPU.

The modem control section is the interface for three input signals - Data Set Ready, Carrier Detect, and Clear to Send - and three output signals - Request to Send, Data Terminal Ready, and Modem Change/Transmitter Empty - used for "handshaking" and status indication between the CPU (via the interface) and the modem (used on RS-232-C interface only). The modem control section asserts the Modem Change/Transmitter Empty interrupt control signal when either the Data Set Ready or the Carrier Detect modem input signal has changed state or when the transmitter shift register is empty. Status of the modem signals is reported in the status register contained in the operation control section of the PCI. A modem status register external to the PCIs monitors and reports the status of two modem signals, Ring Indicator and Clear to Send, that are not reported by the PCIs.

The SYN/DLE control section contains the SYN1 register, the SYN2 register, and the DLE register. These 8-bit registers are addressed by bits 1 and 2 of the operation address, and whether they are to be read or loaded is determined by

operation address bit 3. Their contents are the characters required for synchronization, idle fill, and data transparency in the synchronous mode.

Finally, each PCI contains a baud rate generator which can be programmed either to accept an external clock (lines 0 and 1) or to generate internal transmit or receive clocks. A 10-MHz crystal oscillator external to the PCIs supplies a common timing reference for the four PCI chips.

The PCI for a given line is selected when it receives a PCI select signal from the PCI line select decoder, described below. The contents of all PCI internal registers are reset to 0 by a system reset signal, which results from an I/O Reset instruction.

Control Circuitry and Registers

The control circuitry and registers include two shared (nonPCI) registers and their associated logic plus the PCI line select decoder. The control registers are the current line address register and the operation address register.

The current line address register holds two bits that determine which of the four communications lines is to be affected by DOA, DIA, and IOPLS instructions. It is loaded with data bits D6 and D7 when a Specify Comm Line and Operation instruction (DOB) is issued, and it is reset to 00 by an I/O Reset instruction (IORST).

The PCI line select decoder receives the contents of the current line address register and produces a PCI-select signal for the line addressed. Table 1-8 shows how data bits D6 and D7 determine the current line address.

The operation address register holds four bits that determine which type of information - data, control, status, error checking, or parameter - is to be passed to or from the line specified in the current line address register. These bits also determine, except in the case of the BCC accumulation register, the direction of transfer of that information. The register is loaded with data bits D<0-3> when a Specify Comm Line and Operation instruction (DOB) is issued. Data bits D<0-3> become operation address bits <0-3>. An I/O Reset instruction (IORST) resets the operation address to 0000.

The contents of the operation address register are routed to the different program-accessible registers, including those in the PCIs, as shown in Figure 1-6. Table 1-9 shows how operation address bits <0-3> control the operation of these registers.

Table 1-8 Current line address

Bit 6	Bit 7	Current Line Address	Line Number Addressed
0	0	00	0
0	1	01	1
1	0	10	2
1	1	11	3

Table 1-9 Operation address

Bit 0 (BCC)	Bit 1 (A0)	Bit 2 (A1)	Bit 3 (R/W)	Specified Operation
0	0	0	0	Read receiver holding register
0	0	0	1	Write transmitter holding register
0	0	1	0	Read PCI status register and modem status register
0	0	1	1	Write SYN1, SYN2, DLE registers *
0	1	0	0	Read Mode 1/2 registers *
0	1	0	1	Write Mode 1/2 registers* and BCC polynomial register
0	1	1	0	Read command register
0	1	1	1	Write command register, transmitter interrupt disable register, data set change interrupt disable register
1	X	X	0	Read or write BCC accumulation register - receiver section
1	X	X	1	Read or write BCC accumulation register - transmitter section

** These registers are internal to the PCIs and are addressed sequentially; that is, successive accesses to a particular address will reference the different registers sharing that address one after another. Mode register 1 and register SYN1 are the first to be accessed after an I/O Reset (IORST) or Read Command Register (DOB/DIA) instruction sequence. An X means the bit can be either 0 or 1 — a "don't care" bit.*

Interrupt Control Circuitry

The interrupt control circuitry includes the Busy/Done logic and interrupt control logic of the I/O interface IOC, the two interrupt disable registers and their associated logic, and the interrupt line register.

The Busy/Done logic of the I/O interface IOC responds to CPU-issued Start and Clear commands as described in the programming summary. This IC also contains circuitry that responds to a CPU MSKO instruction to turn the USAM card's Interrupt Disable flag off or on. For a description of the MSKO instruction refer to the assembly language programmer's reference manual for your CPU.

When the USAM's Interrupt Disable flag is set to 0 and its Busy flag is set to 1, the signal "DEVCOM" from the logic associated with the interrupt line register will set the Busy flag to 0 and the Done flag to 1, thus causing the I/O interface to initiate an interrupt request.

The interrupt line register and logic will assert the signal "DEVCOM" whenever it receives an interrupt control signal from one of the PCIs. Whether or not a particular signal asserted by a PCI reaches the interrupt line register depends upon whether interrupts for that type of signal have been enabled: receiver ready interrupts are always enabled; transmitter ready interrupts are controlled by the contents of the transmitter interrupt disable register; and modem change/transmitter empty interrupts are controlled by the contents of their interrupt disable register.

The interrupt line logic includes a priority encoder that assigns priorities to the lines in numerical order - line 0 having highest priority and line 3 lowest - and to receivers first, then transmitters and modems; that is, receivers 0 through 3 have priority over any other section, and transmitter 1 has priority over transmitters 2 and 3 and modems 2 and 3, and so on. The address of the highest priority line requesting service is stored in the line interrupt register, from which it can be read via the USAM data bus (bits D6 and D7) with a Read Interrupt Line instruction (DIB). This register always contains the current highest-priority address and cannot be reset.

Block Check Circuitry

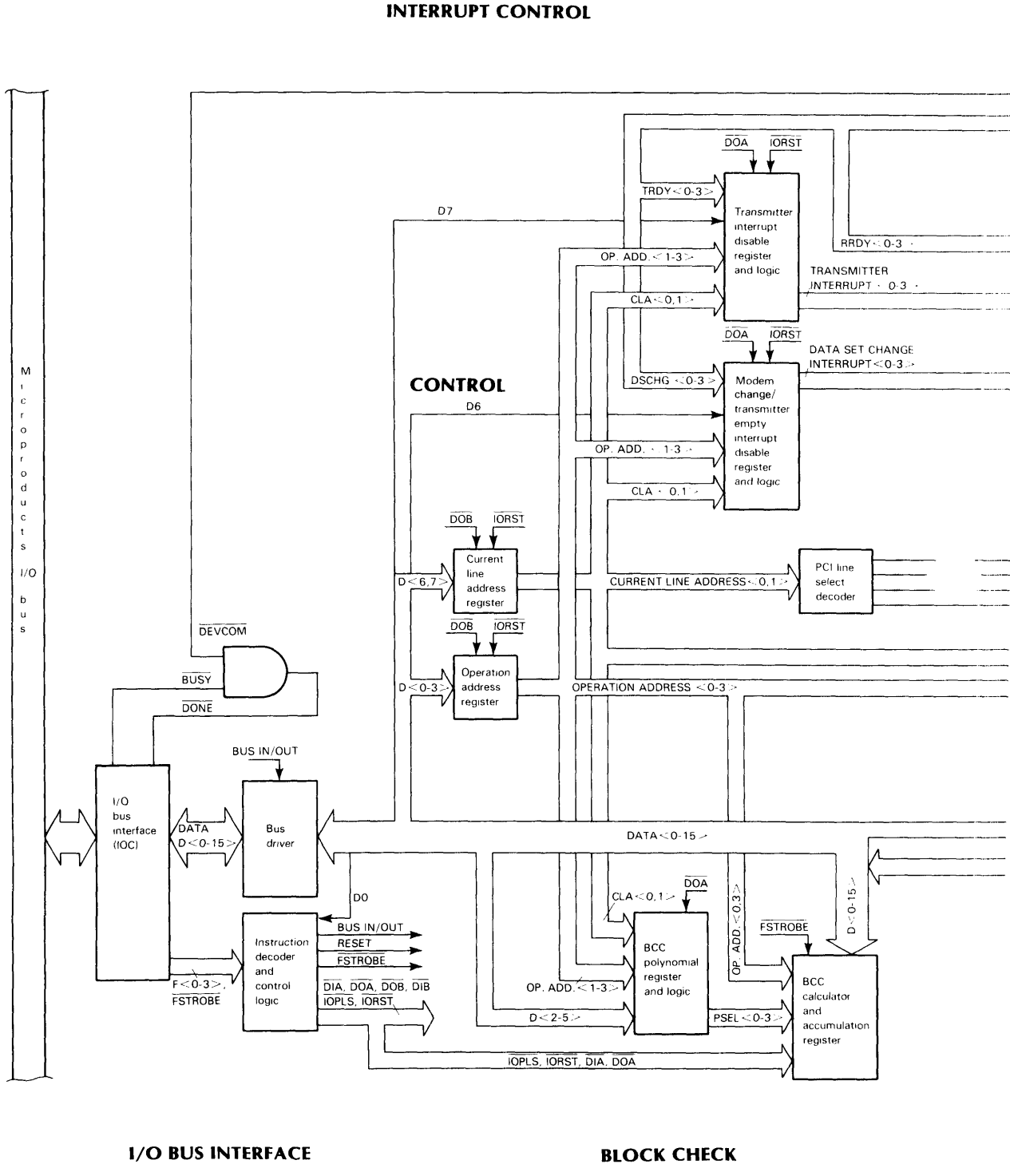
The block check circuitry, which performs error checking for synchronous operation (USAM-4 only), consists of the BCC polynomial register with its associated logic and the BCC calculator and accumulation register.

The BCC polynomial register holds a 4-bit polynomial select word for each of the four communications lines.

The BCC calculator and accumulation register contains a serial divider that is shared by the communications lines. When the divider is activated by an IOPLS (P) command, it performs a block check calculation using the contents of the BCC accumulation register for the current line and the data on the data bus accompanying a DOA or DIA instruction. (If a P command is appended to any other instruction, it may cause the BCC accumulation to be indeterminate.) The results of this calculation replace the previous contents of the BCC accumulation register for the current line. The calculation is performed independently by the hardware at the same time that the PCI is performing its function, so that the error checking does not affect the data exchange rate. The type of calculation performed depends on the contents of the BCC polynomial register for the current line. These serve to select one of four BCC generator polynomials as follows:

BCC Polynomial (Bits D<2-5>)	Protocol	Generating Polynomial
0001	LRC-8	$X^8 + 1$
0010	CRC-12	$X^{12} + X^{11} + X^3 + X^2 + X^1 + 1$
0100	CRC-16	$X^{16} + X^{15} + X^2 + 1$
1000	CRC-CCITT	$X^{16} + X^{12} + X^5 + 1$

The BCC accumulation register holds a 16-bit BCC accumulation for each section (receiver or transmitter) of each communications line. The program sets the register's contents for each section to some initial value for each line for which error checking is to be performed.

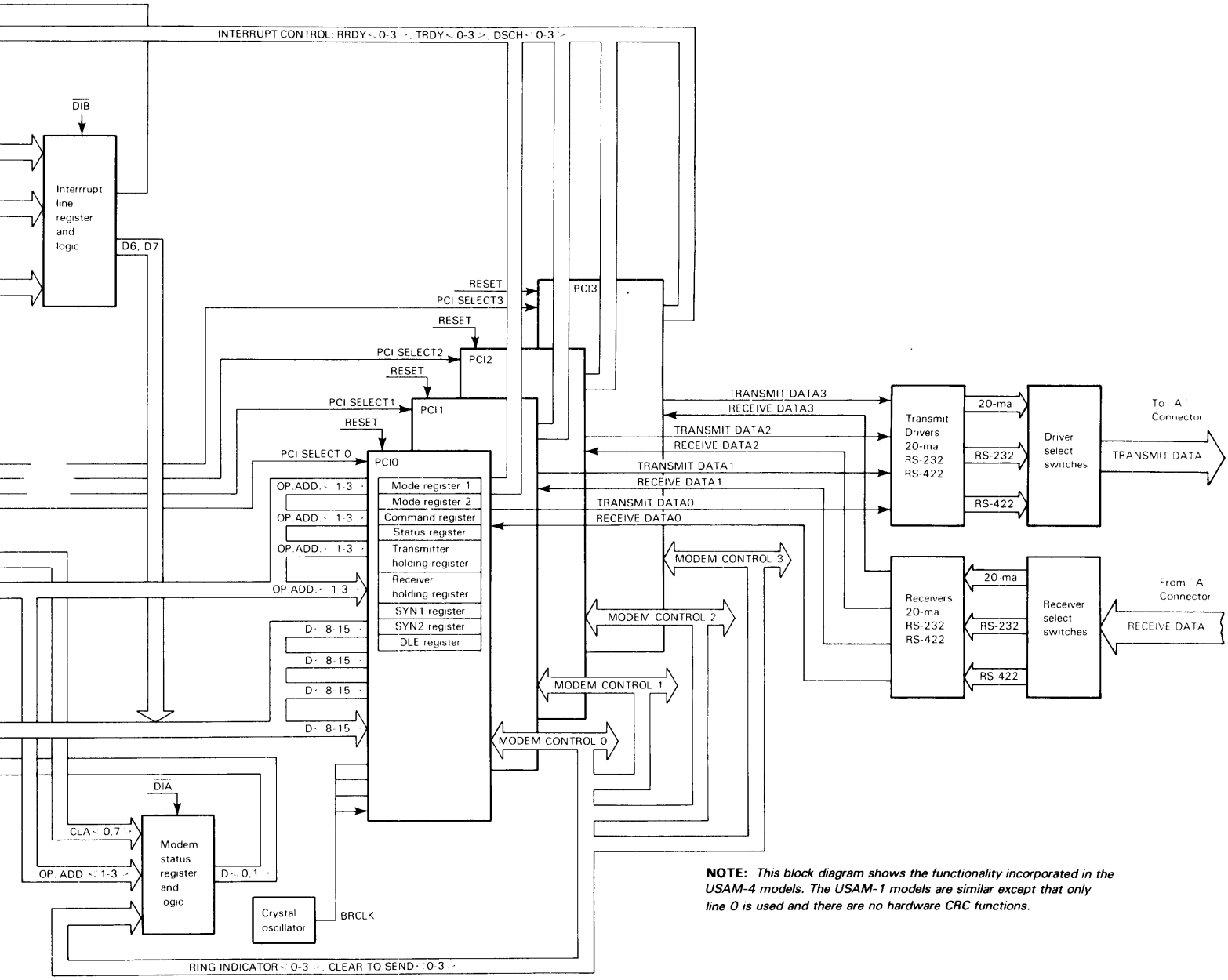


Schematic No. 001-003440

DG-07891

Figure 1-6 USAM functional block diagram

PCIS



NOTE: *This block diagram shows the functionality incorporated in the USAM-4 models. The USAM-1 models are similar except that only line 0 is used and there are no hardware CRC functions.*

Asynchronous Controller 2

The Model 4207-S asynchronous controller provides communication between a Desktop Generation computer system and a full-duplex, asynchronous communication line. This line may be local, with either 20-mA or EIA RS-232-C characteristics, or it may be connected to the system through a modem, either dedicated line or auto-answer. The communications line may operate at 50 to 19,200 baud. The data structure may consist of 5 to 8 data bits/character; 1 or 2 stop bits; and even, odd, or no parity.

The asynchronous controller is commonly used to control a terminal via a local connection, and to communicate with a remote site via modems operating over long lines.

This chapter tells how to program the controller and provides an overview of its theory of operation.

Programming Summary

This section defines the program-accessible registers of the controller as well as its assembly language instruction set. It also presents a programming flowchart and describes programming considerations. Table 2-1 presents a summary of the controller instruction set; Figure 2-1 presents the accumulator formats of the instructions used to program the controller; and Table 2-2 summarizes the response of the controller to device flag commands and the I/O reset instruction.

Table 2-1 Programming summary: asynchronous controller specifications

Mnemonic	
first transmitter	TTO*
first receiver	TTI*
second transmitter	TTO1
second receiver	TTI1
Device code	
first transmitter	11_3*
first receiver	10_8*
second transmitter	51_8
second receiver	50_8
Mask bit	
Transmitter	15_8
Receiver	14_8
Data structure	1 start bit 5, 6, 7, or 8 data bits (jumper selectable) even, odd, or no parity (jumper selectable) 1 or 2 stop bits (jumper selectable)
Line speed	50, 75, 110, 134.5, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, and 19,200
Maximum allowable program I/O latency (ms)	200 at 50 baud to 0.5 at 19,200 baud
Modem control signals	Carrier On Data Set Ready Ring Indicator Clear To Send Request To Send Data Terminal Ready

* *These are the device codes and mnemonics used by DGC software for the system console. However, you can select any two device codes for the asynchronous controller via its jumpers.*

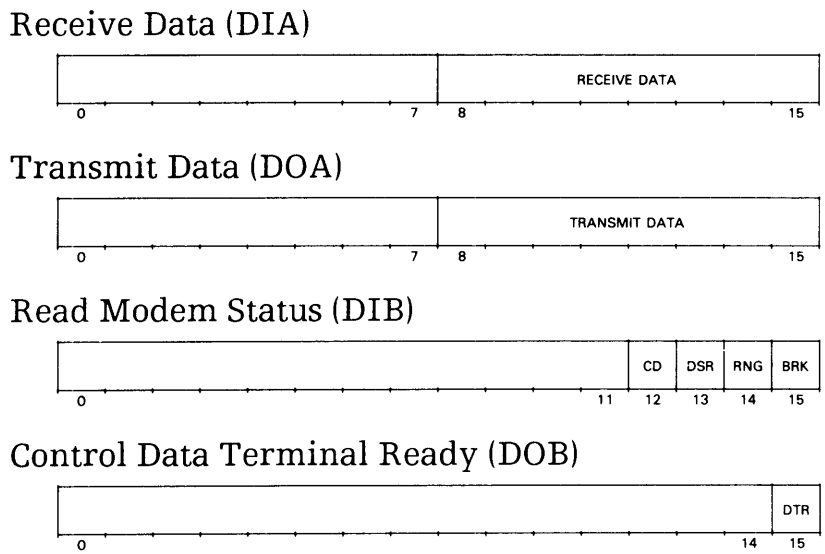


Figure 2-1 Programming summary: accumulator formats

Table 2-2 Programming summary: START,CLEAR, PULSE, and IORST functions

f = (option omitted)	Does not affect Busy and Done flag
f = S	Sets the Busy flag to 1 and the Done flag to 0
f = C	Sets both the Busy and Done flags to 0
f = P	Sets the Done flag to 0*

**A pulse signal directed to the receiver sets the Break Indicator contained in the modem status register to 0.*

Registers

The asynchronous controller has four registers: a transmit data register, a receive data register, a modem status register, and a data terminal ready (DTR) control register. The transmit and receive data registers are both eight bits in length. The transmit data register accepts data from the processor for transmission to a communications line. The receive data register accepts data from the communications line.

The modem status register provides the processor with the status of the communications line and modem (if one is used). This register contains four bits, three of which indicate the modem's operational state and a fourth which indicates whether the receiver has detected a "break" condition.

The DTR control register determines the state of the controller's data terminal ready control line. This line directs the modem to answer or disconnect incoming calls.

Interrupts

The asynchronous controller has two Interrupt Disable flags, one for the transmitter and one for the receiver. These flags determine whether or not a program interrupt request initiated by either the transmitter or receiver is passed on to the processor. The Maskout instruction manipulates these flags. They are also cleared (interrupts enabled) by an I/O Reset instruction.

Instruction Set

The asynchronous controller appears as two separate devices to the program: an input device or receiver, and an output device or transmitter. Each device requires its own Busy and Done flags.

You use four instructions to program the asynchronous interface: three for the receiver and one for the transmitter. One of these loads a character from an accumulator into the transmit data register. The second reads a character from the receive data register into an accumulator. The third allows the program to determine, in detail, the state of the controller and modem (if applicable), and the fourth controls the state of Data Terminal Ready.

You control the operation of the two parts of the controller through manipulation of its two sets of Busy and Done flags as follows:

f=S Sets the Busy flag to 1 and the Done flag to 0. If directed to the receiver and a terminal with reader is attached, starts the reader.

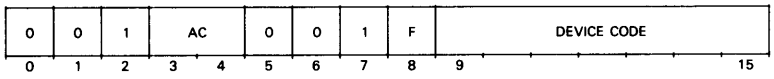
f=C Sets both the Busy and Done flags to 0. If directed to the receiver and a terminal with reader is attached, stops the reader.

f=P Sets the Done flag to 0 without affecting the Busy flag. If directed to the receiver, sets the Break Indicator flag to 0.

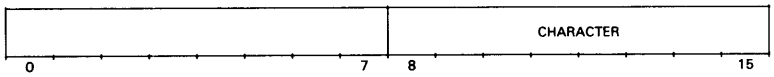
IORST Sets both the Busy and Done flags to 0. Also sets the Break Indicator flag to 0 and removes the assertion of DTR.

Read Receive Data Register

DIA*[f]* *ac,*TTI



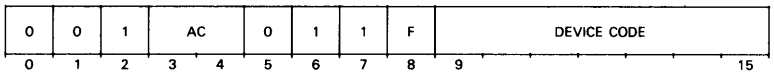
Loads the contents of the receive data register into bits 8-15 of the specified accumulator. The character is right justified. After the data transfer, performs the function specified by f. The previous contents of the specified accumulator are lost. The format of the specified accumulator is



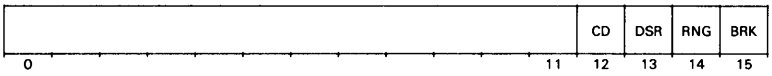
Bits	Name	Contents
0-7	—	Reserved for future use; all bits are set to 0
8-15	Character	The last character received from the communication line, right justified

Read Modem Status

DIB*[f]* *ac,*TTI



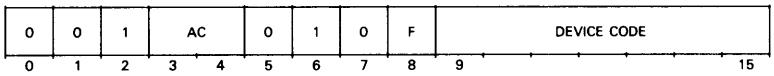
Places the contents of the modem status register in bits 12-15 of the specified accumulator. Bits 0-11 are set to 0. The previous contents of the accumulator are lost. After the transfer, the function specified by f is performed. The specified accumulator has the format:



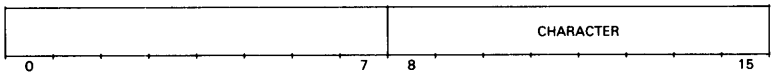
Bits	Name	Contents or Function
0-11	—	Reserved for future use; all bits are set to 0
12	Carrier detect	State of the modem's Carrier Detect signal (1 = on; 0 = off)
13	Data set ready	State of the modem's Data Set Ready signal (1 = on; 0 = off)
14	Ring	State of the modem's Ring signal (1 = on; 0 = off)
15	Break	When 1, a framing error has been detected by the receiver (the character did not have a stop bit)

Transmit Character

DOA*[f]* *ac,*TTO



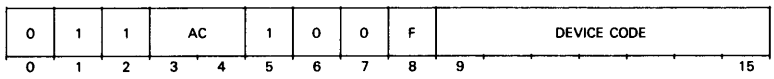
Places the contents of bits 8-15 of the specified accumulator into the transmit data register. The controller appends the appropriate start, stop, and parity bits and transmits the character. The contents of the specified accumulator remain unchanged. After the data transfer, performs the function specified by f. The format of the specified accumulator is



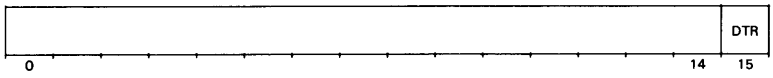
Bits	Name	Contents
0-7	—	Reserved for future use
8-15	Character	The character, right justified, to be transmitted

Control Data Terminal Ready

DOB*[f]* *ac*,TTI



Sets the state of the modem control signal, Data Terminal Ready, according to the state of bit 15 of the specified accumulator. The contents of the specified accumulator remain unchanged. After the data transfer, performs the function specified by f. The specified accumulator has the format:



Bits	Name	Contents or Function
0-14	—	Reserved for future use
15	Data terminal ready	Sets the state of the Data Terminal Ready signal (0 = off; 1 = on)

Programming Considerations

Since the asynchronous controller has both a transmitter and a receiver, each operating as a separate device and each having its own Busy and Done flags, the following discussion treats these as independent entities.

Transmitting

Transmitting data from the controller to the communications line is a straightforward operation. If the transmitter is idle (the Busy flag is 0), execute a Transmit Character instruction (DOA). The character must be right justified in the accumulator (the least significant bit of the character is in bit 15 of the specified accumulator). You may use a Start command to set the Busy flag to 1 and the Done flag to 0. This allows the program to monitor the state of the transmission by examining the Busy and Done flags and allows the controller to initiate a program interrupt request upon the completed transmission. The transmitter automatically adds the requisite start, stop, and parity bits.

Upon completion of the transmission, the controller sets the Busy flag to 0 and the Done flag to 1, thus initiating a program interrupt request. (See Figure 2-2.)

With the exception of the Data Terminal Ready (DTR) signal, all the signals required to operate a modem are automatically asserted by the controller. In general, modems require DTR before they respond to other signals from the controller. (See "Timing," below, for more details.) The controller automatically asserts Request to Send after a character is loaded into its transmit data register, it waits until the modem returns Clear to Send before performing the serial character transmission. A jumper allows continual assertion of Clear to Send for applications that require it. Tables 2-3 and 2-4 list the modem control signals.

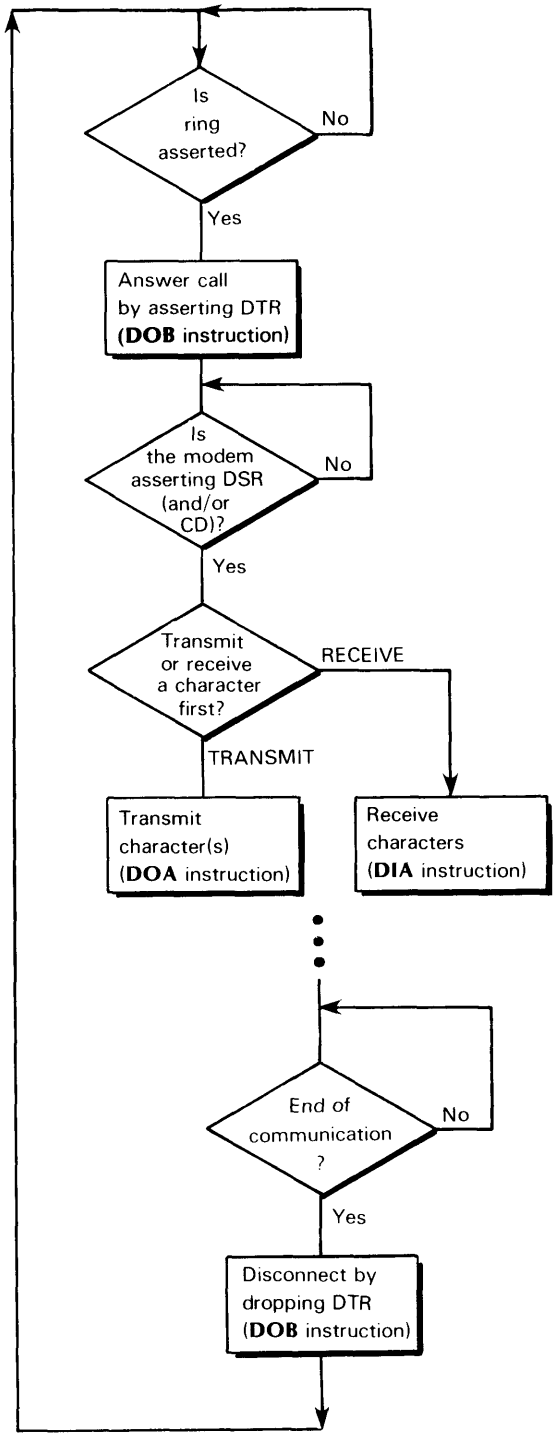


Figure 2-2 Communications under modem control

Receiving

Receiving characters is also a straightforward operation. The receiver portion of the controller strips the start, stop, and parity bits from the serial data stream and places the character in the receive data register. No matter what the state of the receiver's Busy flag at this time, the receiver sets the Busy flag to 0 and the Done flag to 1, thus initiating a program interrupt request. The program can then read the character by executing a Read Receive Data Register instruction (DIA).

A start command sets the Busy flag to 1 and the Done flag to 0, while a clear command sets both the Busy and Done flags to 0. Using the Start command allows the program to determine if the receiver is waiting for a character (Busy is 1), or if it has a character fully assembled in its receive data register (Done is 1).

When operating with a modem, you must consider an additional factor. Modems require the signal Data Terminal Ready in order to operate. The program can control the state of Data Terminal Ready. Data Terminal Ready is not asserted upon power up; the program must explicitly set it before operating with a modem. All other modem control signals are generated automatically by the controller. Note that an IORST sets Data Terminal Ready to 0.

Timing

The only timing constraints encountered with the asynchronous controller are concerned with the maximum allowable programmed I/O latency times.

The maximum allowable programmed I/O latency for a receiver is a time less than one character time. A character time is defined as

$$\text{Charater Time} = \frac{\text{Bits/character}}{\text{Bits/second}}$$

where bits/character =
1(start) + n(data bits) + 1(parity) + #stop bits

If the maximum allowable programmed I/O latency is exceeded for the receiver, a second character may overwrite the first, thus causing data to be lost. If the figure is exceeded for the transmitter, no data is lost but the communication line is operating inefficiently.

Table 2-3 Modem control signals (input to asynchronous controller)

Signal Name	Mnemonic	Function
Ring Indicator	RNG	Indicates to the controller that the modem is receiving an incoming call. The software should test this bit to determine when the modem is receiving an incoming call.
Data Set Ready	DSR	Indicates to the controller that the modem is powered up and ready to transfer data. This signal is usually asserted by the modem after the controller asserts Data Terminal Ready.
Carrier Detect	CD	Indicates to the controller that the modem is detecting a carrier signal. It may be used by the program to determine when to receive data (however, it is not necessary in many protocols).
Clear to Send	CTS	This signal is program transparent. Indicates that the modem is ready to transmit data.

Table 2-4 Modem control signals (output from asynchronous controller)

Signal Name	Mnemonic	Function
Data Terminal Ready	DTR	Directs the modem to form a connection to the communications line. DTR must remain on for the duration of the call. Setting DTR to 0 directs the modem to disconnect the circuit (hang up).
Request to Send	RTS	This signal is program transparent. Directs the modem to prepare for data transmission.

Power-Up Response

The asynchronous controller is initialized upon power up. The following registers and flags are set to 0:

Busy flag,

Done flag,

Interrupt disable flag,

DTR control register,

Break indicator bit,

Transmit register,

Receive register.

An I/O Reset instruction initializes these registers and flags in the same way.

Installation and Jumpering

For installation and jumpering information, refer to the appropriate installation document for your system.

Theory of Operation

The asynchronous controller may be divided into two devices: the transmitter and the receiver. Each device consists of four major elements as shown in Figure 2-3. These elements are:

- An interface to the I/O bus,
- A control section,
- A universal asynchronous transmitter/receiver (UAR/T),
- Interface logic to the terminal or modem.

Of these four elements, only the UAR/T is common to both the transmitter and the receiver.

A description of the interface to the I/O bus appears in *Input/Output and Interfacing*. For more detailed information, see *microNOVA Integrated Circuits Data Manual* (DGC No. 014-000074).

Each control section uses a ROM to decode the control signals issued from the bus interface. A signal from the transmitter control section directs the loading of the transmit register in the UAR/T. Signals from the receiver control section direct the reading of the receive register in the UAR/T, the reading of the modem status register, and the loading of the DTR control register.

The universal asynchronous transmitter/receiver performs the parallel-to-serial conversion for the transmitter and the opposite conversion for the receiver. It also appends the appropriate start bit, stop bit(s), and parity bit to transmit data and strips them from received data. If the UAR/T detects a framing error (referred to in this document as a line break condition), it loads bit 15 of the modem status register with a 1.

There are three interface sections: the transmit interface logic, the receive interface logic, and the modem interface logic. The transmit and receive interface logic are configured via jumpers to provide either 20-mA current loop or EIA RS-232-C connections. The modem interface logic provides EIA RS-232-C connections.

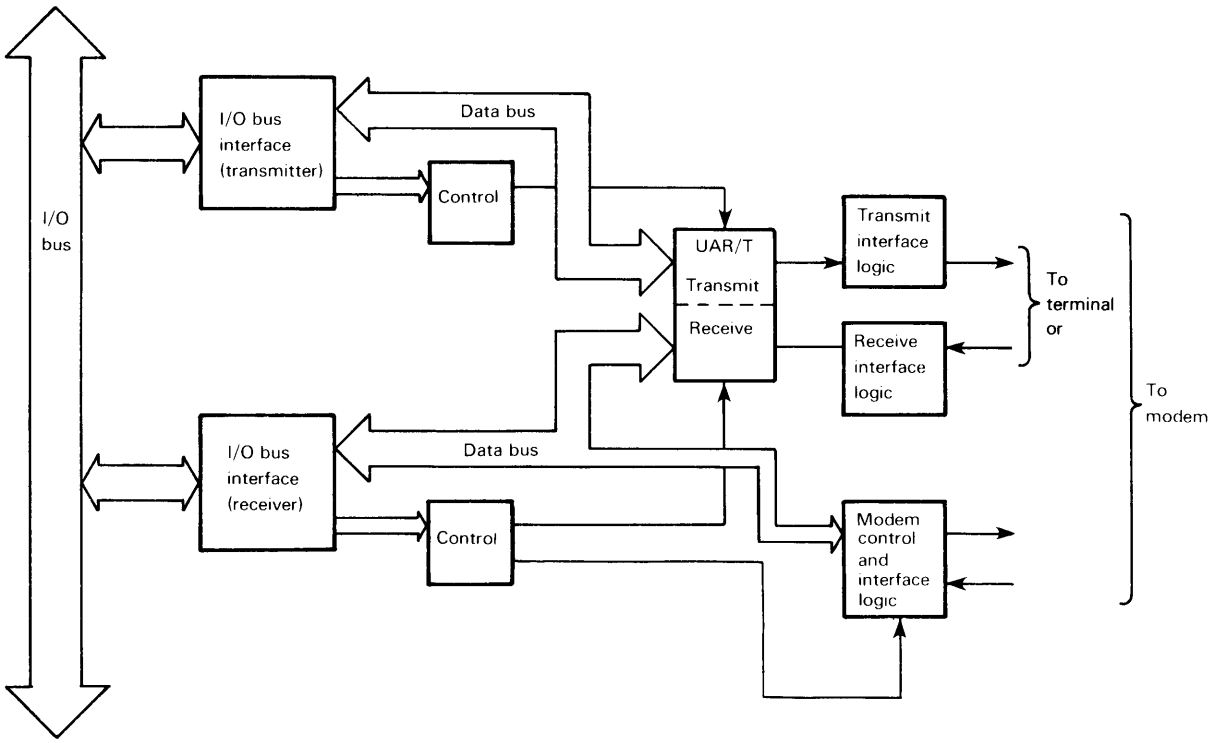


Figure 2-3 Asynchronous controller block diagram

Index

Within this index, the letter *f* following a page entry indicates "and the following page"; the letters *ff* following a page entry indicate "and the follwing pages."

T

U

moisten & seal

Documentation Comment Form

Manual Title _____

Manual No. _____

Your Name _____

Your Title _____

Company _____

Street _____

City _____ State _____ Zip _____

Please help us improve our future publications by answering the questions below. Use the space provided for your comments. Thank you.

	Yes	No
Is this manual easy to read?	☐	☐
Is it easy to understand?	☐	☐
Are the topics logically organized?	☐	☐
Is the technical information accurate?	☐	☐
Can you easily find what you want?	☐	☐
Does it tell you everything you need to know?	☐	☐
Do the illustrations help you?	☐	☐

If you wish to order manuals, contact your sales representative or dealer.

Comments:

Date

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 26 SOUTHBORO, MA. 01772

Postage will be paid by addressee:

DataGeneral

ATTN: Design Services (E219)
4400 Computer Drive
Westboro, MA 01581