HP 3000 Computer Systems

HP 30055A Synchronous Single-Line Controller (SSLC)

Installation and Service Manual

HEWLETT hp PACKARD

19447 PRUNERIDGE AVE, CUPERTINO, CALIFORNIA 95014

Manual Part No. 30055-90001
Microfiche Part No. 30055-90006
Product No. 30055A
Printed Circuit Assembly 30055-60001

Printed in U.S.A. 12/77

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the most recent date on which the technical material on any given page was altered. If a page is simply re-arranged due to a technical change on a previous page, it is not listed as a changed page.

Second Edition . . . . Dec 1977

Changed Pages	Effective Date
Title	Aug 1978
ii	Apr 1979
iii	Apr 1979
iv	Apr 1979
v	Aug 1978
viii	Apr 1979
1-0	Apr 1979
1-1	Apr 1979
1-2	Aug 1978
1-3	Apr 1979
1-4	Apr 1979
1-5	Aug 1978
2-1	Aug 1978
2-9	Aug 1978
2-12	Aug 1978
2-13	Aug 1978
2-15	Aug 1978
2-17	Aug 1978
2-18	Aug 1978
2-19	Aug 1978
2-27	Apr 1979
3-1	Aug 1978
3-6	Aug 1978
3-8	Aug 1978
3-11	Apr 1979
4-6	Aug 1978
4-7	Apr 1979
4-8	Apr 1979
5-1	Apr 1979
5-3/5-4	Apr 1979
Index	Aug 1978

PRINTING HISTORY

New editions incorporate all update material since the previous edition. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The date on the title page and back cover changes only when a new edition is published. If minor corrections and updates are incorporated, the manual is reprinted but neither the date on the title page and back cover nor the edition change.

First Edition	Nov 1974
Second Edition	Dec 1977
Update No. 1	Aug 1978
Update No. 1 Incorporated	Nov 1978
Update No. 2	Apr 1979
Reprint, Update No. 2 Incorporated	Oct 1979

PREFACE

This manual contains installation and servicing information for the HP 30055A Synchronous Single-Line Controller. The controller provides I/O capability between the HP 3000 Computer System and Bell System 201, 208, and 209 modulator/demodulators (modems) or equivalent. Data transfer rates of up to 2,400 bits per second can be realized with Bell 201 modems , up to 4,800 bits per second with Bell 208 modems, and up to 9,600 bits per second with Bell 209 modems.

This manual was written with the assumption that the reader has a thorough knowledge of the HP 3000 I/O system hardware and is familiar with binary synchronous communication techniques.

This manual is organized as follows:

a. Section I, General Information, describes the main features of the controller and explains briefly how it operates, describes how the controller is configured in the computer system, lists the equipment supplied with a controller, and tabulates controller specifications.

b. Section II, Programming Information, provides reference information regarding the software operation of the HP 3000 Computer System. Included is information regarding I/O programming, controller programming, and sample receive and transmit programs.

c. Section III, Theory of Operation, contains information on the hardware operation of the controller PCA and associated modem. An overview of the I/O system is given as well as a block-level description of the controller PCA.

d. Section IV, Maintenance, contains general servicing information. Included in this section is an alphabetical list of controller PCA signals, a wiring diagram of the controller to modem connecting cable, and a description of applicable maintenance aids.

e. Section V, Installation, describes the installation of the controller PCA and interconnecting cable into an HP 3000 Computer System.

This manual can be used with the related documentation listed below.

HP 3000 Series II and Series III Computer Systems:

- HP 3000 Series II/III Computer Systems Service Manual, part no. 30000-90018.
- HP 3000 Series II Computer System Installation Manual, Models 5, 7, and 9, part no. 30000-90019.
- HP 3000 Series II Computer System Installation Manual, Models 6 and 8, part no. 30000-90081.
- HP 3000 Series III Computer System Installation Manual, part no. 30000-90081.
- Synchronous Single-Line Controller Stand-Alone Diagnostics Manual (D434A and D434B), part no. 30055-90008.
- HP 3000 Computer Systems Engineering Diagrams Set, part no. 30000-90076.
- Data Communications Handbook, part no. 30000-90105.

HP 3000 pre-Series II Computer Systems:

- HP 3000 Detailed Diagrams Manual, detailed diagram DD-503, part no. 30055-90003.
- Synchronous Single-Line Controller Stand-Alone Diagnostic Manual (D334), part no. 30055-90004.

CONTENTS

Section Page

I. GENERAL INFORMATION

II. PROGRAMMING INFORMATION

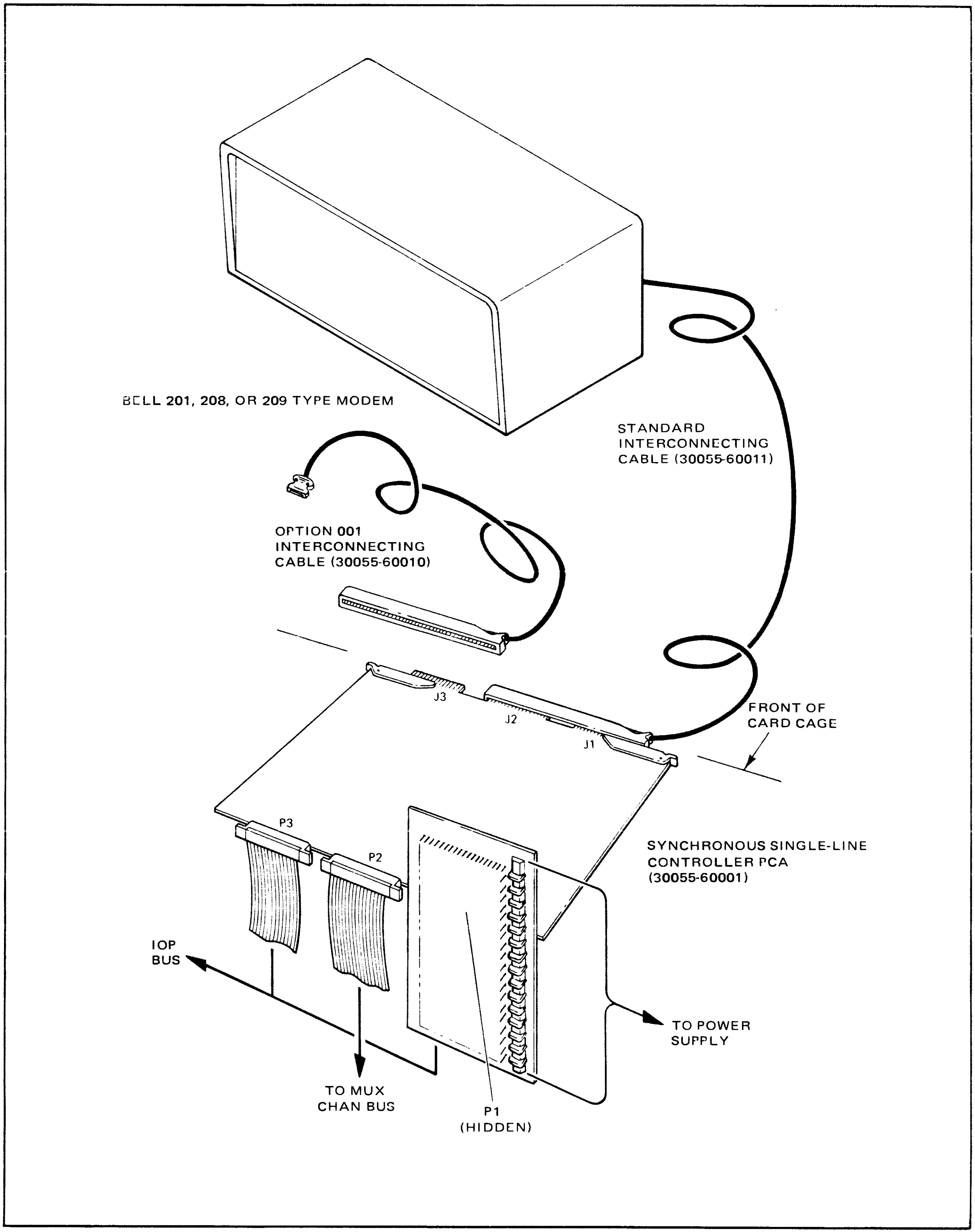
CONTENTS (continued)

ILLUSTRATIONS

Title Page

TABLES

Title Page


BELL 201, 208, OR 209 TYPE MODEM
STANDARD INTERCONNECTING CABLE (30055-60011)
OPTION 001 INTERCONNECTING CABLE (30055-60010)
J3
J2
J1
FRONT OF CARD CAGE
P3
P2
SYNCHRONOUS SINGLE-LINE CONTROLLER PCA (30055-60001)
IOP BUS
TO POWER SUPPLY
TO MUX CHAN BUS
P1 (HIDDEN)

7041-101

Figure 1-1. Synchronous Single-Line Controller.

GENERAL INFORMATION

SECTION I

1-1. INTRODUCTION

This section describes the functional and physical characteristics of the HP 30055A Synchronous Single-Line Controller as illustrated in figure 1-1. Related publications that may be required to service the controller are listed in the preface to this manual.

1-2. GENERAL DESCRIPTION

The HP 30055A Synchronous Single-Line Controller (hereafter referred to as "controller") provides the HP 3000 Computer System with binary synchronous data communication capabilities.

The controller is capable of operating in two modes, synchronous and asynchronous. With synchronous communications, transmissions are IBM Bisync compatible and occur via modems. With asynchronous communications, transmissions are patterned after IBM Bisync (with stop and start bits added) and they occur over hardwired lines. The mode of operation is selectable by the use of different cables; synchronous operation uses the standard 30055-60011 interfacing cable while asynchronous hardwired operation uses the special 30055-60010 interfacing cable (Option 001).

1-3. Features

Important features of the controller are:

* EIA RS-232-C compatible.

* Programmable baud rate.

* Programmable character length (from five to eight bits including parity).

* Associate character recognition capability (i.e., the capability of recognizing a particular character received from an external device and taking some specific action associated with that character). The associate character and the specific action taken are programmable: up to four characters and their associated actions can be specified. A typical use of associate character recognition is to detect that a SYNC character has been received from the external device.)

* The capability of controlling one synchronous device having up to four control and six sense lines.

* Data packing and unpacking capability (i.e., the capability of transferring two characters in one data word between the controller and the computer system memory.

* Programmable parity type (i.e., odd or even).

* Up to four masked interrupts permitted from the modem.

* Synchronous or asynchronous (hardwired) operation selectable by changing the interconnecting cable.

1-4. Operation

The controller provides an interface to EIA RS-232-C compatible devices. Control information and status requests are transmitted to the controller either directly from the CPU or via the multiplexer channel. Data transfer is typically initiated and controlled through the multiplexer channel.

A typical synchronous data input operation of the controller is as follows:

a. The controller is tested via a direct TIO instruction to determine its availability.

b. A direct SIO instruction is issued to the controller initiating execution of an I/O program.

c. The I/O program initializes the controller with such information as baud rate, packing/unpacking, character length, and associate character (the first associate character is typically "SYNC").

d. A read order in the I/O program enables the controller to begin receiving data from the external device.

e. When data is received, each character is compared with the associated character (i.e., SYNC). If characters other than SYNC are received, they are ignored.

f. When the first SYNC character is detected, "character sync" is established. The I/O program, through execution of a control order and another read order, then instructs the controller to begin searching for an STX (start text) or an SOH (start heading) associate character.

g. When an SOH or STX character is detected, another read order causes subsequent characters to be transferred as data until an ETB (end block), ETX (end text), or ENQ associate character is detected.

h. The next character received from the external device is the BCC (i.e., block check character. This character provides validity checking for the data just transferred.

i. The controller issues an interrupt request to the CPU, informing the software that the input data is ready to be processed.

A typical data output operation of the controller is as follows:

a. The controller is tested via a direct TIO instruction to determine its availability.

b. A direct SIO instruction is issued to the controller initiating execution of an I/O program.

c. The I/O program initializes the controller with such information as baud rate, packing/unpacking, and character length.

d. A write order in the I/O program transmits four SYNC characters to the external device.

e. Another write order in the I/O program then transmits the data block (including control and check characters - supplied by the operating system) to the external device.

f. Finally, the controller issues an interrupt request to the CPU, informing the system software that the data transmit operation is completed.

1-5. INTERFACE TO SYSTEM

The controller is installed in the same PCA module as the multiplexer channel. The controller communicates with the multiplexer channel via the MUX CHAN BUS at connector P23, with the IOP via the IOP BUS at connectors Pl and P3, and with the modem or asynchronous repeater via a cable connection at connectors Jl and J2 (see figure 1-2).

1-6. EQUIPMENT SUPPLIED

The HP 30055A synchronous single-line controller consists of the following:

a. One Synchronous Single-Line Controller PCA, part no. 30055-60001.

b. One controller-to-modem interconnecting cable assembly, part no. 30055-60011.

c. One test connector assembly, part no. 30055-60009.

d. One installation and service manual, part no. 30055-90001.

e. One Stand-Alone HP 30055A Diagnostic Operating Manual, part no. 30055-90004. This manual should be used if the controller is installed in a pre-Series II Computer System. If the controller is installed in a Series II or Series III Computer System, discard the manual with part no. 30055-90004 and use the Synchronous Single-Line Controller Stand-Alone Diagnostic Manual, part no. 30055-90008 (which is supplied with the computer system).

f. If Option 001 is specified, one interconnecting cable assembly, part no. 30055-60010 is supplied in place of interconnecting cable assembly, part no. 30055-60011.

1-7. SPECIFICATIONS

The specifications for the HP 30055A synchronous single-line controller are presented as follows:

a. Programming specifications are listed in table 1-1.

b. Hardware specifications are listed in table 1-2.

Table 1-1. Programming Specifications

Programming Methods	I/O program under control of MUX CHAN and IOP or under direct control of CPU/IOP.
I/O Program Initiation	CPU SIO instruction.
I/O Orders Executed	Control Write Read End Jump Sense Interrupt
Interrupts Generated	Programmed Word Transfer Required Associate Character Recognition Status from modem

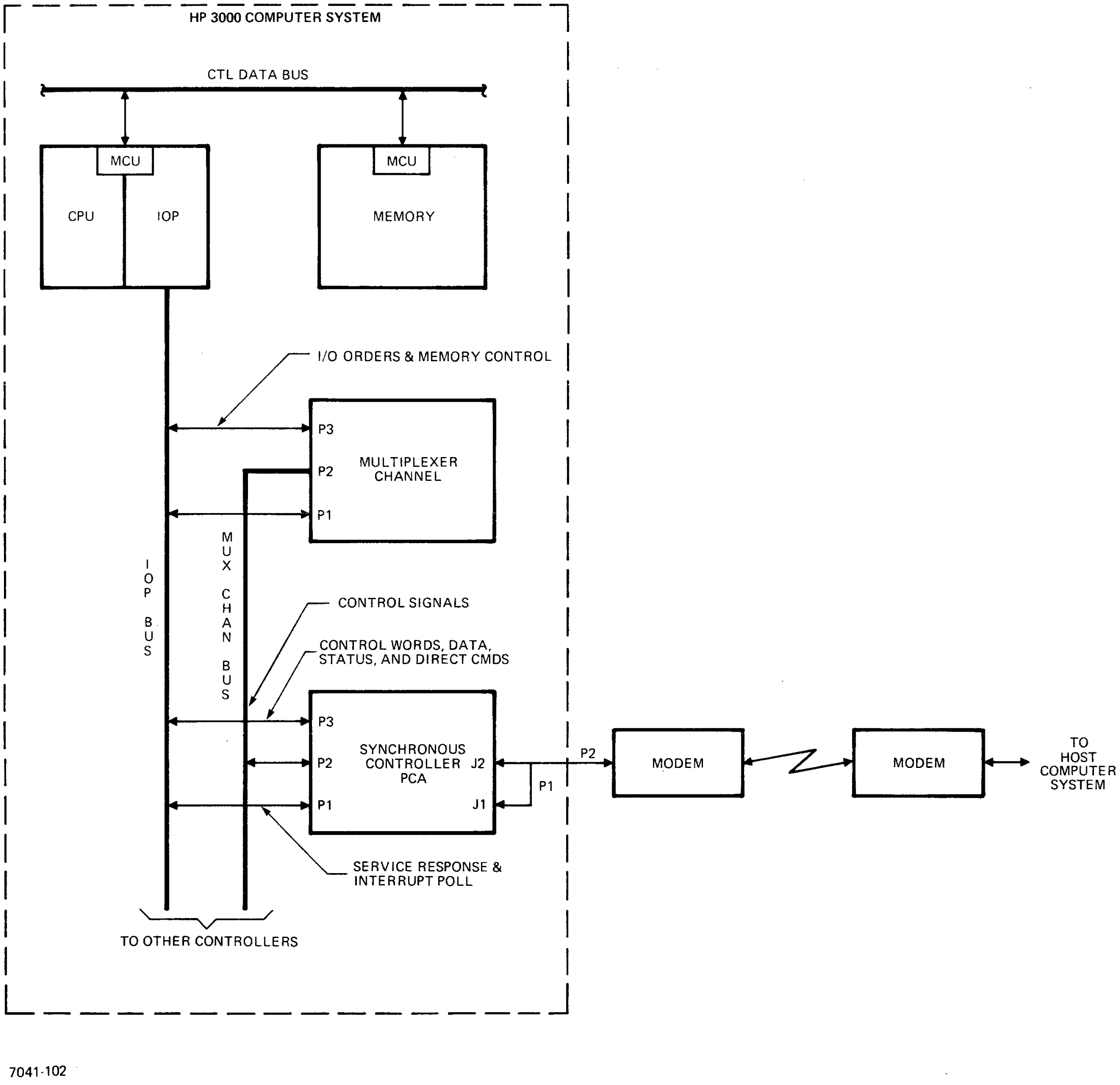


Figure 1-2. Typical Interface To System For Synchronous Operation.

Table 1-2. Hardware Specifications

Part Number	30055-60001
Dimensions	11.5 in. x 13.7 in. x 0.7 in.
+5V Supply Current Required	3.5 A
Logic Levels:	
Logic 0 (low)	+0.4V or less
Logic 1 (high)	+2.4V or greater
Environmental Requirements:	Same as CPU
Interfaces:	
IOP BUS	Data, command, control, status, and interrupt between controller and CPU/IOP
MUX CHAN BUS	Control between controller and multiplexer channel
Interconnecting Cable	EIA RS-232-C interface to modem
Data Paths	
From Remote Device to controller	Bit Serial
From Controller to CPU/IOP	16-bit Parallel

PROGRAMMING INFORMATION

SECTION II

2-1. INTRODUCTION

This section contains programming information pertaining to the operation of the controller in the HP 3000 Computer System. Descriptions of the pertinent I/O instructions and programmed I/O orders are included as well as control word formats, status formats, and data formats. Also included is information regarding I/O programming for data transfer operations.

2-2. GENERAL INFORMATION

The controller permits bi-directional data transfer between an external device and the HP 3000 Computer System main memory. Controller operations are programmed through direct I/O instructions and through programmed I/O orders. (Execution of I/O orders is initiated by the direct SIO instruction.)

All control, data, and status information transferred between the computer system memory and the controller is via the IOP and MUX CHAN buses. Data is transferred between the controller and memory in 16-bit words. Each word contains either one or two characters with each character containing up to eight valid bits. The numbers of characters per word and bits per character are programmable. Data between the controller and the external device is transferred synchronous or asynchronous, bit serial, with up to eight valid bits per character.

2-3. I/O PROGRAMMING

The HP 3000 Computer System controls the operation of an I/O device and the associated data transfer by executing I/O instructions to that device. These instructions cause the IOP to issue direct commands to the specified device controller. One of these commands, SIO, is used to initiate the execution of an I/O program associated with that device controller. The I/O program, which is stored in memory, is generated prior to the execution of the SIO instruction. When the SIO instruction is received and acknowledged the multiplexer channel begins fetching and executing the I/O program associated with the device. The I/O program, after initiation, runs independently of CPU operation.

The I/O program for a particular device controller, consists of a number of I/O program doublewords. A doubleword contains such information as the order (specifying the operation to be performed), device controller dependent control words, and the count field and buffer address for data transfers. As part of the I/O operation associated with a device controller, the device controller returns a 16-bit status word when directed.

Figure 2-2 illustrates all pertinent programming word formats.

2-4. I/O Instructions

The HP 3000 Computer System has an instruction set of 170 machine instructions. Within this set is a subset of 11 instructions pertaining to I/O and interrupt operations. Seven of these instructions can be used to control the operation of the synchronous single-line controller and the associated modem. They are: SMSK (Set Mask), SIO (Start I/O), RIO (Read I/O), WIO (Write I/O), TIO (Test I/O), CIO (Control I/O), and SIN (Set Interrupt).

See figure 2-2 for the I/O instruction format. The following is a list of items common to all I/O instructions:

a. Bits 0 through 7 specify the I/O group of instructions (i.e., subop code 03).

b. Bits 8 through 11 specify the particular I/O instruction within the I/O group of instructions.

c. Bits 12 through 15 are defined as the "K" field. (The "K" field is used to locate a device number within the stack.) The "K" field can contain any value from 0 to 15. The number in the "K" field, when subtracted from the Top of the Stack (TOS) address (S reg. minus "K" value) yields the address in the stack of the word containing the device number. For example, if the device number is stored in the second word from the TOS, the "K" value should be 0010 binary.

d. If the desired I/O instruction is executed successfully, the condition code indicator in the CPU is set to CCE.

e. If the specified controller is not ready to perform the desired I/O command, the condition code indicator is set to CCG.

f. If the specified controller does not respond to the desired I/O command, the condition code indicator is set to CCL.

2-5. SMSK (SET MASK) INSTRUCTION. This instruction issues a mask word, located on the TOS, along with a Set Mask command to all device controllers and stores the new mask word in the CPU mask register. Each bit in the mask word corresponds to a group of device controllers (i.e., subsystems). A "1" bit sets the Mask FF(s), enabling the interrupt circuitry in the group of device controllers wired corresponding to that bit. A "0" bit clears the Mask FF(s), disabling the interrupt circuitry in the corresponding group. If there is no response to the Set Mask command, the condition code indicator in the CPU is set to CCL. Otherwise, the condition code indicator is set to CCE.

NOTE

> The Mask FFs in all device controllers are set when an I/O Reset signal is issued (i.e., by operating the I/O Reset switch on the auxiliary control panel.

2-6. SIO (START I/O) INSTRUCTION. This instruction expects the absolute starting address of an I/O program (i.e., the I/O program pointer) to be on the TOS and a device number to be in the stack at S-K. The SIO instruction first issues a Test I/O command to the device controller specified in the stack. The status from that device controller is then checked. If the device controller and associated device are ready then bit zero of the status word is a "1". If bit zero (i.e., SIO OK) is a "1", the TOS, containing the I/O program pointers, is stored in the first word location of the DRT associated with the device controller. A Start I/O command is then issued to the specified device controller. The device controller, in turn, signals the multiplexer channel which begins fetching and executing the I/O program on behalf of the specified device controller. When the CPU is informed by the multiplexer channel that the Start I/O command has been accepted, the CPU deletes the I/O program pointer from the TOS and sets the condition code indicator to CCE.

If the device controller status word bit zero indicates that either the device or device controller is not ready (i.e., bit zero = "0"), the device controller status word is pushed onto the TOS and the condition code indicator is set to CCG. If the controller fails to respond to the Start I/O command, the TOS is unchanged (i.e., I/O program pointer still on TOS) and the condition code indicator is set to CCL.

2-7. RIO (READ I/O) INSTRUCTION. This instruction expects a device number to be in the stack at S-K. The RIO instruction first checks the status of the device controller by issuing a Test I/O command to the device controller. If the device controller is ready (i.e., RIO/WIO OK = "1"), then a Read I/O command is issued to the appropriate device controller. When the device controller receives the command, it returns a 16-bit data word to the CPU. This 16-bit data word is then pushed onto the TOS and the CPU condition code indicator is set to CCE. (Refer to paragraph 2-40 for a description of data word formats.)

If the device controller is not ready (i.e., RIO/WIO OK = "0"), then the status word from the device controller is pushed onto the TOS and the condition code indicator is set to CCG. If the device controller fails to respond to the Test I/O command, the condition code indicator is set to CCL and the RIO instruction is aborted.

2-8. WIO (WRITE I/O) INSTRUCTION. This instruction expects a data word to be on the TOS and a device number to be in the stack at S-K. (Refer to paragraph 2-40 for a description of data word formats.) The WIO instruction first checks the status of the device controller by issuing a Test I/O command to the device controller. If the device controller is ready (i.e., RIO/WIO OK = "1"), then a Write I/O command and the data word from the TOS are issued to the specified device controller. When the device controller receives the command and data word, the CPU deletes the TOS and sets the condition code indicator to CCE.

If the device controller is not ready (i.e., RIO/WIO OK = "0"), then the status word from the device controller is pushed onto the TOS and the condition code indicator is set to CCG. If the device controller fails to respond to the Test I/O command, the inqtruction condition code indicator is set to CCL and the WIO instruction is aborted.

2-9. <u>TIO (TEST I/O) INSTRUCTION.</u> This instruction causes a Test I/O command to be issued to the device controller specified by the device number at S-K. The status of the device is obtained and pushed into the TOS and the condition code indicator in the CPU is set to CCE. If the device controller fails to respond to the Test I/O command, the TOS remains unchanged and the condition code indicator is set to CCL.

2-10. <u>CIO (CONTROL I/O) INSTRUCTION.</u> This instruction causes a control word located on the TOS and a Control I/O command to be issued to the device controller specified by the device number at S-K. If the device controller acknowledges receiving the control word, the TOS is deleted and the condition code indicator in the CPU is set to CCE. If the device controller fails to respond, the condition code indicator is set to CCL.

2-11. <u>SIN (SET INTERRUPT) INSTRUCTION.</u> This instruction causes a Set Interrupt command to be issued to the device controller specified by the device number at S-K. Provided that the interrupt circuitry in the device controller is enabled, an interrupt occurs. If the device controller fails to respond to the Set Interrupt command, the condition code indicator is set to CCL. Otherwise, the condition code indicator is set to CCE.

2-12. Direct Commands

There are eight direct commands which can be transmitted, via the IOP bus, to a device controller. They are: Set Mask, Test I/O, Start I/O, Read I/O, Write I/O, Control I/O, Set Interrupt, and Reset Interrupt. Refer to table 2-1 for IOP bus bit configurations of direct commands.

Table 2-1. Direct Commands

DIRECT COMMAND	BITS ON IOP BUS		
	IOCMD 00	IOCMD 01	IOCMD 02
Set Mask	1	1	0
Test I/O	1	0	1
Start I/O	0	1	0
Read I/O	1	1	1
Write I/O	0	1	1
Control I/O	0	0	1
Set Interrupt	0	0	0
Reset Interrupt	1	0	0

2-13. SET MASK COMMAND. This command occurs when the CPU executes a SMSK instruction. The Set Mask command causes device controllers to set or clear their Mask FFs depending on the mask word configuration and the particular device controller group interrupt mask jumper configuration. If the bit selected by the device controller group interrupt mask jumper is a "1", the device controller Mask FF sets. If the bit is a "0", the device controller Mask FF Clears. Setting the device controller Mask FF permits that device controller to issue interrupt requests to the CPU. Clearing the Mask FF prevents a device controller from issuing interrupt requests.

2-14. TEST I/O COMMAND. This command occurs when the CPU executes either an SIO instruction, an RIO instruction, a WIO instruction, or a TIO instruction. The Test I/O command causes the selected device controller to gate its status word onto the IOP bus. In the case of a rejected SIO instruction, a rejected RIO instruction, a rejected WIO instruction, or a TIO instruction, the status word is stored on the TOS in the CPU.

2-15. START I/O COMMAND. This command occurs when the CPU executes an SIO instruction to a device controller found to be ready to perform an I/O program (i.e., Test I/O command returned SIO OK status = "1"). The Start I/O command causes the selected device controller to issue an SIO signal to the multiplexer channel. This, in turn, initiates the execution of an I/O program associated with the device controller. As a result of the Start I/O command, bit zero of the device controller status word (i.e., SIO OK) is set to "0". This indicates that the device controller and associated multiplexer channels are busy executing an I/O program.

2-16. READ I/O COMMAND. The Read I/O command occurs when the CPU executes a RIO instruction to a device controller found to be "ready" to perform a direct I/O operation (i.e., Test I/O command returned RIO/WIO OK status = "1"). The Read I/O command causes the selected device controller to gate a data word onto the IOP bus. The data word is stored on the TOS in the CPU.

2-17. WRITE I/O COMMAND. The Write I/O command occurs when the CPU executes a WIO instruction to a device controller found to be "ready" to perform a direct I/O operation (i.e., Test I/O command returned RIO/WIO OK status = "1"). The Write I/O command causes the selected device controller accept the data word on the IOP bus. The data word is from the TOS in the CPU.

2-18. CONTROL I/O COMMAND. This command occurs when the CPU executes a CIO instruction. The Control I/O command causes the specified device controller to perform the action(s) specified by the associated 16-bit control word. The control word is sent from the TOS in the CPU via the IOP bus. Refer to paragraph 2-30 for the control word formats associated with the single-line controller.

2-19. SET INTERRUPT COMMAND. This command occurs when the CPU executes a SIN instruction. The Set Interrupt command sets the Interrupt Request FF in the specified device controller. If the device controller Mask FF is set, the device controller sends an interrupt request to the CPU. The Set Interrupt command also sets bit two of the device controller status word to a "1" (i.e., interrupt pending).

2-20. RESET INTERRUPT COMMAND. This command occurs when the CPU executes an EXIT instruction associated with a particular device controller interrupt. The Reset Interrupt command clears the interrupt circuitry, except for the Mask FF, on that device controller.

2-21. I/O Program Orders

Execution of an I/O program is initiated when the CPU/IOP issues a direct Start I/O command to a device controller. An I/O program consists of a number of I/O program doublewords. The doublewords consist of an I/O command word (IOCW) and an I/O address word (IOAW). Contained in the IOCW is an I/O program order. The order specifies the type of operation the device controller and/or multiplexer channel is to perform. There are seven orders which can be executed to a device controller. They are: Control, Write, Read, End, Jump, Sense, and Interrupt. See figure 2-2 for the I/O program doubleword format for the seven orders.

2-22. CONTROL ORDER. This order causes the mutiplexer channel to send bits 4:15 of the Control order IOCW and bits 0:15 of the IOAW to the device controller as control information. The control information may specify an address, set up some type of transfer mode, or perform some other function on the device controller. Only the control word contained in the IOAW is used by this controller. Refer to paragraph 2-30 for the control word formats for the controller.

2-23. WRITE ORDER. This order causes the multiplexer channel to transfer a block of data from memory to the device controller. The block length in words is specified in twos complement form by the word count (IOCW bits 4:15). The 12 bits of word count information provide for transfers of from one word (i.e., 4:15 = %7777) to 4096 words (i.e., 4:15 = %0000). The absolute starting address of the data block in memory to be written is specified by the IOAW. (Refer to paragraph 2-40 for a description of data word formats.)

If bit zero of the Write order IOCW is a "1", then data chaining is specified. Data chaining means that the transfer caused by the data chained Write order and the next order in sequence (which must also be a Write and may be chained) are linked together as a single block transfer. Therefore, the counts of two or more orders are combined to provide for transfers of greater than 4096 words (i.e., the limit imposed by the 12 bits of the Write order IOCW). As many Write orders as necessary can be chained together, but the last Write order in the series must not specify data chaining. The data addresses in memory specified by the IOAWs in a series of chained writes can specify noncontiguous location in memory (i.e., gather write).

The device controller can prematurely terminate the block transfer by asserting the Device End signal to the multiplexer channel. Device End causes the multiplexer channel to stop transferring data associated with the current Write order, then fetch the next I/O program doubleword in sequence. A Return Residue order can then be used to return the word count remaining when the Device End was asserted.

2-24. READ ORDER. This order causes the multiplexer channel to transfer a block of data from the controller to memory. The block length in words is specified in twos complement form by the word count (IOCW bits 4:15). The 12 bits of word count information provide for transfers of from one word (i.e., 4:15 = %7777) to 4096 words (i.e., 4:15 = %0000). The absolute starting address of the data storage location (i.e., data buffer) in memory is specified by the IOAW. (Refer to paragraph 2-40 for a description of data word formats.)

If bit zero of the Read order IOCW is a "1", then data chaining is specified. Data chaining means that the transfer caused by the data chained Read order and the next order in sequence (which must also be a Read and may be chained) are linked together as a single block transfer. Therefore, the count of both (or all) orders is combined to provide for transfers of greater than 4096 words (i.e., the limit imposed by the 12 bits of the Read order IOCW). As many Read orders as necessary can be chained together, but the last Read order in the series must not specify data chaining. The data addresses in memory specified by the IOAWs in a series of chained Reads can specify non-continguous locations in memory (i.e., scatter read).

The device controller can prematurely terminate the block transfer by asserting the Device End signal to the multiplexer channel. Device End is normally used by the controller as a function of associate character recognition. If programmed by the appropriate control word, the controller can be made to issue a device end signal when a pre-specified character is received from the external device. Device End causes the multiplexer channel to stop transferring data associated with the current Read order, then fetch the next I/O program doubleword in sequence. A subsequent Return Residue order can be used to return the word count remaining when the Device End was asserted.

2-25. <u>END ORDER.</u> This order causes the multiplexer channel to stop the fetching and the execution of I/O program orders. Three actions occur as a result of executing the End order: 1) The device controller status word is obtained and stored in the End order IOAW location in memory; 2) If bit four of the IOCW is a "1", the device controller is ordered to interrupt the CPU; and 3) Bit zero of the device controller status word is set to "1" (i.e., SIO OK).

2-26. <u>JUMP ORDER.</u> The execution mode of this order depends on the state of bit four of the IOCW. If bit four is a "0", then an unconditional jump is specified and the address in the Jump order IOAW is used as the address of the next doubleword to be executed.

If bit four is a "1", then a conditional jump is specified. This mode is used by the controller in conjunction with the associate character recognition feature. When programmed by the appropriate control word configuration, the controller will issue a jump signal (when interrogated by the multiplexer channel during the DRT fetch sequence) if the specified associate character has been received from the external device. If the associate character was detected then the address of the next order to be executed is taken from the Jump order IOAW location. If the jump condition (i.e., associate character detected) was not met then the next order to be executed is the one following the Jump order.

2-27. SENSE ORDER. This order causes the multiplexer channel to obtain the device controller status word, then store the status word in the Sense order IOAW location in memory.

2-28. INTERRUPT ORDER. This order causes the multiplexer channel to order the device controller to interrupt the CPU. If the device controller Mask FF is set, an interrupt request is sent to the CPU/IOP. I/O program execution proceeeds with the next order in sequence.

2-29. RETURN RESIDUE ORDER. This order causes the multiplexer channel to return the residue count to the Return Residue order IOAW location in memory. The residue count is the count from the just-completed Read or Write order IOCW bits 4:15 minus the number of words transferred by that order (i.e., no. of words expected but not received). The residue count is returned in twos complement form. The value can be from 4096 words (i.e., %170000) to 0 words (i.e., %00000.)

2-30. Control Word

Twelve separate control words are used to control the operation of the controller and associated modem. The control words each consist of 16 bits, the first four bits (bits 0:3) determine which of the 12 possible control words is being received, the remaining 12 bits (bits 4:15) determine the operation to be performed.

The 12 possible control words are: 1) I/O Reset, 2) Clear Interrupt and Receive Logic Logic, 3) Interrupt Control, 4) Receive/Transmit Channel Select, 5) Receive Baud Rate, 6) Transmit Baud Rate, 7) Data Format and Control, 8) Receive/Transmit Control, and 9-12) four Associate Character and Function Control control words. The Data Format and Control, Receive/ Transmit Control, and four Associate Character and Function Control control words can be transmitted to the controller between data characters. Care must be taken to ensure that a control word which acts on a data character (such as an associate character) arrives at least 1.5 microseconds before the data character is fully clocked into the controller receiver section. The remaining six control words should never be transmitted to the controller while data is being received or transmitted.

Paragraphs 2-31 through 2-38 describe each of the 12 control word formats. Thse descriptions apply to the control word transmitted from the TOS during CPU execution of a CIO instruction as well as the control word transmitted as the IOAW portion of a control order. Refer to figure 2-2 for the various control word formats.

2-31. I/O RESET. This control word initializes the controller logic circuitry. After this control word is issued, all control status bits are cleared (i.e., logic "0"). (The same function can be performed by operating the I/O RESET switch on the CPU auxilliary control panel.) Additionally, control line C3 to the modem is asserted (i.e., Data Set Ready).

2-32. CLEAR INTERRUPT AND RECEIVE LOGIC. The operation of this control word is similar to I/O RESET except that only the interrupt and receive logic circuits on the controller are cleared. Clearing the interrupt logic clears all of the circuits that can initiate an interrupt request and therefore clears the interrupt pending status bit (bit 2).

The receive logic clearing initializes the controller receive section circuits and as a result clears status bits 1, 3, 5, 6, 8, and 10. (Status word format is described in paragraph 2-39.)

2-33. INTERRUPT CONTROL. This control word enables or disables the controllers ability to issue interrupts to the CPU and, also, can clear any pending interrupts.

When an interrupt condition occurs, that condition is stored in the controller and consequently, the interrupt pending status bit is set (bit 2). If bit 4 of this control word is set, the interrupt condition is permitted to issue an interrupt request to the CPU (provided that the Mask FF is set). If interrupts are disabled (i.e., bit four = "0") then no interrupt can occur until the interrupt circuitry is enabled. If bit 5 of the control word is set, then any pending interrupts are cleared (clearing the interrupt pending status bit).

2-34. RECEIVE/TRANSMIT CHANNEL SELECT. Presently, only one channel is used. Therefore channel "0" is used and bits four through seven are always set to logic "0s".

In addition to being controlled by this control word, the channel numbers are cleared to logic "0s" when an I/O RESET control word or auxiliary control panel signal is received.

2-35. RECEIVE AND TRANSMIT BAUD RATES. These control words set the receive and transmit baud rates. The receive and transmit baud rates are set, and function, independently. The range of baud rates is from 75 bits per second to 96,000 bits per second and is determined by configurating bits 4:15 of the control word.

A table included in figure 2-2 lists the most commonly used baud rates and the corresponding control word configuration. The following is a method for determining baud rate configurations for baud rates not included in the table. With certain baud rates, a precise baud rate configuration cannot always be calculated. The configuration may contain an error. If the percent error is less than 1.4%, then the baud rate configuration is acceptable to most systems.

To determine the error in calculating a baud rate and to calculate the control word configuration do the following.

1. The percentage error is calculated using the following method:

   $$I(\text{intermediate calculation} = \frac{1.92 \times 10^5}{\text{desired baud rate}}$$

   then

   $$I(\text{error}) = \frac{(I(\text{rounded off})-I) \times 100}{I}$$

   For example, to determine the error in calculating the configuration for 1050 bits per second proceed as follows:

   $$I = \frac{1.92 \times 10^5}{1.05 \times 10^3} = 182.76$$

   $$I(\text{rounded off}) = 183$$

   $$I(\text{error}) = \frac{(183.00-182.76) \times 100}{182.76} = 0.13\%$$

   0.13% is within the allowable 1.4% error margin.

2. When the calculated baud rate error is determined to be less than the allowable 1.4%, the intermediate calculated baud rate I from step 1 is used to determine the configuration of bits 4:15 as follows:

   F(final calculation) = a. Binary equivalent of I

   b. One´s complement of binary equivalent of I.

   c. Most significant bit on right (i.e., bit 15 of control word).

   For example, to determine the control word configuration for 1050 bits per second proceed as follows:

   a. Binary equivlent of I for 1050 = 00010110111 (control word requires 12 bits).

   b. Ones complement of binary equivalent of I = 000100101111.

c. Most significant bit on right

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

= %457.

For synchronous data transfer (such as when operating with BELL 201, 208, or 209 type modems) the receive baud rate control word is needed. Also, the transmit baud rate control word is used only for devices that require a transmit clock (the BELL 201, 208, or 209 modem supplies its own transmit clock).

2-36. <u>DATA FORMAT AND CONTROL.</u> This control word sets up to eight data format options for the data transmitted between the controller and the external device:

a. Echo (bit 4). If this bit is a "1", the incoming data from the external device is immediately sent back to the external device. Echo is controlled automatically by the controller, therefore there are no special programming considerations.

b. Parity Sense (bit 5). This bit determines the type of parity used by both the receive and transmit sections. If a "1", odd parity is used, if a "0", even parity is used.

c. Transmit Unpack (bit 6). If this bit is a "1", then the data word received from the IOP is assumed to contain two characters and the two characters are transmitted to the external device (data word bits 0:7 transmitted followed by data word bits 8:15). If bit six is a "0" then only data word bits 8:15 are considered data. The actual number of bits transmitted is dependent upon the specified character length (see figure 2-2).

d. Transmit Parity (bit 7). If this bit is a "1", a parity bit is inserted into the transmitted data code after the last data bit (regardless of the length of the character). Parity type depends on parity sense (bit 5).

e. Receive Pack (bit 8). If this bit is a "1" then two incoming data characters are packed into one data word for transfer to the IOP. The first character received is placed into bits 0:7, the second character received is placed into bits 8:15. The actual number of data bits in a character is determined by the specified character length.

f. Parity Interrupt Receive (bit 9). If this bit is a "1", then any parity error on received data will generate an internal interrupt request.

g. Mask Bit 8 (bit 10). If bit 10 is a "1", then the received data character bit 8 is transmitted to the IOP as a logic "0".

h. Character Length (bits 11:15, including parity bit if present). These bits select the length of the transmitted character. They do not include the start and stop bits for asynchronous mode. If two stop bits must be sent, then two character length bits are set to one. The second bit is set according to the length of the character plus one.

Table 2-2 illustrates typical data format and control word configurations.

Table 2-2. Typical Data Format and Control Word

```
 _______________________________________________________________
|                                                               |
| Typical control word:                                         |
|                                         1   1   1   1   1   1 |
|                                                               |
| 0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5 |
|---------------------------------------------------------------|
|   |           |           |           |           |           |
| 0 | 0   1   0 | 0   1   1 |     1     |     0   0 |           |
|___|___________|___________|___________|___________|___________|
|                 |   |   |   |   |   |   |   |   |   |   |   | |
| Echo-----------/    |   |   |   |   |   |   |   |   |   |   | |
|    Parity Sense---/     |   |   |   |   |   |   |   |   |   | |
|        Transmit Unpack/     |   |   |   |   |   |   |   |   | |
|                             |   |   |   |   |   |   |   |   | |
|   Error Checking Type:    ----- | --------- |   |   |   |   | |
|        CRC-16             | 0 | | | 0 | 0 | |   |   |   |   | |
|                           ----- | --------- |   |   |   |   | |
|        VRC/LRC            | 1 | | | 1 | 1 | |   |   |   |   | |
|                           ----- | --------- |   |   |   |   | |
|        VRC/CRC-16         | 0 | | | 0 | 0 | |   |   |   |   | |
|                           ----- | --------- |   |   |   |   | |
|        CRC-12             | 0 | | | 0 | 0 | |   |   |   |   | |
|                           ----- | --------- |   |   |   |   | |
|                             |   |   |   |   |   |   |   |   | |
|          Transmit Parity--/     |   |   |   |   |   |   |   | |
|              Receive Pack-----/     |   |   |   |   |   |   | |
|                  Parity Int Rec.--/     |   |   |   |   |   | |
|                      Mask Bit 8-------/     |   |   |   |   | |
|                                             |   |   |   |   | |
|                                             |---------------| |
|                                                 Character     |
|  Bit No. | 11 | 12 | 13 | 14 | 15 |               Length      |
|  --------|----|----|----|----|----|                 |         |
|  Async   | 9  | 8  | 7  | 6  | 5  |-----------------|         |
|  --------|----|----|----|----|----|                           |
|  Sync    |    |    | 8  | 7  | 6  |                           |
|_______________________________________________________________|
```

2-37. RECEIVE/TRANSMIT CONTROL. This control word is used to control data transmission between the controller and external device, transmit control information to the modem, and enable selectable interrupts from the modem.

a. Blind Receive (bit 4). When this bit is a "1", data from the external device is prevented from entering the controller receive section (and, therefore, no associate character comparison takes place). This can be used when operating with modems during line turnaround time.

b. Disable Transmit (bit 5). When this bit is a "1", the controller transmit section operates normally, but no data is transmitted to the external device. During the time that transmit is disabled, the line to the modem is held in a "mark" condition. By using the disable transmit function, a controlled time delay can be inserted into an SIO program. The time delay can vary between one character time and %10,000 character times.

c. Transmit Interrupt Enable (bit 6). If this bit is a "1", the controller generates an interrupt (if the interrupt circuitry is enabled) to the IOP when the controller needs a data character to transmit to the external device.

d. Receive Interrupt Enable (bit 7). If this bit is a "1", the controller generates an interrupt to the IOP (if the interrupt circuitry is enabled) when a data word is ready to be transmitted to the IOP.

e. Status Interrupt Enable (bits 8:11). Each bit in this portion of this control word corresponds to one of four status lines from the modem. If a bit is a "1" and the modem changes the state of the corresponding status line, the controller generates an interrupt to the IOP (if interrupt is enabled by the interrupt control word and Mask FF).

f. Control to Device (bits 12:15). These bits can be used to transmit control information to the modem.

NOTE

The use of the status and control bits is dependent on the type of modem used. The actual configuration of the lines associated with these bits is a function of the interconnecting cable assembly associated with the external device.

Table 2-3 illustrates typical receive/transmit control word configurations.

Table 2-3. Typical Receive/Transmit Control Word

```
Typical Control Word:

                                        1   1   1   1   1   1
0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
--------------------------------------------------------------
  |           |           |           |           |
0 | 0   1   1 | 0   0     |           |           |
__|___________|___________|___________|___________|___________
                |   |   |   |   |   |   |   |   |   |   |   |
Blind Receiver/     |   |   |   |   |   |   |   |   |   |   |
 Disable Transmit-/     |   |   |   |   |   |   |   |   |   |
                      --------- |   |   |   |   |   |   |   |
        Direct I/O    | 1 | 1 | |   |   |   |   |   |   |   |
                      --------- |   |   |   |   |   |   |   |
        I/O Program   | 0 | 0 | |   |   |   |   |   |   |   |
                      --------- |   |   |   |   |   |   |   |
                        |   |   |   |   |   |   |   |   |   |
        Rec Int Enable-/    |   |   |   |   |   |   |   |   |
        Xmit Int Enable----/    |   |   |   |   |   |   |   |
           Status Interrupt:    |   |   |   |   |   |   |   |
              S0 --------------/    |   |   |   |   |   |   |
              S1 ------------------/    |   |   |   |   |   |
              S2 ----------------------/    |   |   |   |   |
              S3 --------------------------/    |   |   |   |
                  Control to Device:            |   |   |   |
                    C0 ------------------------/    |   |   |
                    C1 ----------------------------/    |   |
                    C2 --------------------------------/    |
                    C3 ------------------------------------/
```

2-38. ASSOCIATE CHARACTER AND FUNCTION CONTROL. The controller has the ability to compare data being received from the external device with up to four pre-selected characters. If a character being received from the external device matches one of those characters, a controller function, also pre-selected, occurs. The controller functions are conditional jump, assert device end, interrupt, and disable service request.

To select a character for comparison and select an associated controller function, the desired character is entered into bits 8:15 (right justified) and the desired function(s) is selected by setting the corresponding bit(s) to a "1" (bits 4:7). Additionally, if character length is less than eight bits, then the unused bits (i.e., most significant) should be logic "0´s" (synchronous) or "1´s" (asynchronous).

For example, suppose it is desired to issue a device end signal to the multiplexer channel, disable further service request signals, and issue an interrupt request to the CPU when an ASCII ETX (end of text) character is received from the external device, then the control word would look as in table 2-4 (with the possible exception of bits 2:3).

Table 2-4. Typical Associate Function Control Word
(Synchronous Mode)

```
 ____________________________________________________________
|                                                            |
|                                      1   1   1   1   1   1 |
| 0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5 |
|                                                            |
|____________________________________________________________|
|   |           |         |                                  |
| 0 | 1   0   0 | 0   1   1 | 1   0   0 | 0   0   0 | 0   1   1 |
|___|___________|_________|___________________|______________|
| |-------|           |   |   |   |   |---------------------------| |
| Associate           |   |   |   |          ASCII               |
| Function            |   |   |   |       ETX Character          |
|                     |   |   |   |                              |
| Jump Cond------/        |   |   |                              |
|                         |   |   |                              |
|    Device End------/        |   |                              |
|        Interrupt-------/        |                              |
|            Disable SR------/                                   |
|____________________________________________________________|
```

The specific actions taken by the controller for each selected function are as follows:

a. Jump Conditionally (bit 4). If this bit is a "1" and the associated character is encountered during a Read order, a controller flip-flop sets, storing the fact that this condition has been encountered. Then, the next time a conditional jump I/O order is executed, an I/O program jump will occur (refer to paragraph 2-26). The flip-flop storing the jump condition is cleared when the multiplexer channel executes a Jump order to the controller or when an I/O RESET signal occurs.

b. Assert Device End (bit 5). If this bit is a "1" and the associated character is encountered during a Read order, then a device end signal is immediately issued to the mutliplexer channel, causing the channel to terminate the present read order.

c. Interrupt (bit 6). If this bit is a "1" and the associated character is encountered during a Read order, then an interrupt request is issued to the IOP (provided that the interrupt circuitry is enabled).

d. Disable Service Request (bit 7). SYNCHRONOUS: This function is generally used to detect the beginning of a transmission from an external device. Normally, at the beginning of an I/O program to receive data, a CLEAR INTERRUPT and RECEIVE

LOGIC control word is issued to the controller. CLEAR INTERRUPT and RECEIVE LOGIC disables any service request signals to the multiplexer channel and puts the controller into the "hunt" mode (the "hunt" mode means that the controller will "hunt" for the character associated with the disable service request function but no data will be transferred to the IOP). Then an associate function control word is issued to the controller specifying an ASCII SYNC character and the disable service request function. A read order is then issued. At this point, the controller begins waiting for data from the external device. When a SYNC character is encountered and matched with the associate character, "character sync" (i.e., the beginning of a block) is established, but no service request signal is issued for the SYNC character. In fact, further SYNC characters will not generate service request signals. However, the first character that is not a SYNC character will generate a service request signal as will all subsequent characters (with the exception of those associated with the disable SR function).

ASYNCHRONOUS: When asserted and compared, this bit will disable a Service Request; to the HP 3000, it appears as if nothing has happened. This may be used in special cases where null characters are not wanted as entered data.

2-39. Status Word

A 16-bit status word is used to indicate to the software the operational state of a device controller. There are a number of instructions and orders that obtain the status word from a device controller. They are: TIO instruction, SIO instruction, RIO instruction, WIO instruction, End order, and Sense order. The storage location for the stats word is dependent on the reason for obtaining the word:

a. If status is obtained because of a TIO instruction, then the status word is pushed onto the top of the stack (TOS) in the CPU.

b. If status is obtained because of either an SIO, RIO, or WIO instruction, the status word is stored in scratch pad 2 in the CPU. The CPU then checks bit zero of the status word if an SIO instruction is being executed, or bit one if an RIO or a WIO instruction is being executed. If the bit being checked is a "0", then the status word is pushed onto the TOS and the instruction is aborted. If the bit checked is a "1", then execution of the instruction continues.

c. If status is obtained because of an End of Sense order, then the status word is stored in the IOAW location of that order.

The status word format is illustrated in figure 2-2. The meanings of the status word bits are as follows:

a. SIO OK (bit 90. This bit when a "1" indicates the controller is ready to execute an I/O program (i.e., not presently executing an I/O program).

b. RIO/WIO OK (bit 1). This bit when a "1" indicates that either the receive section or the transmit section of the controller is free to execute the appropriate function. Bits three and four of the status word must then be checked to determine if the desired function (i.e., read or write) can be performed. If this bit is a "0", both the receive and transmit sections are busy and, therefore, it is not necessary to check bits three and four.

c. Interrupt Pending (bit 2). This bit when a "1", indicates that an interrupt will occur from the controller if the controller interrupt circuitry is enabled.

d. Read OK (bit 3). This bit when a "1" indicates that the receive section of the controller is ready to perform a read operation.

e. Write OK (Bit 4). This bit when a "1" indicates that the transmit section of the controller is ready to perform a write operation.

f. Byte (bit 5). This bit when a "1" indicates either that the controller is in the "receive pack" mode (refer to paragraph 2-36) and only the first byte has been read (i.e., whole word not assembled) or the controller is not in the receive pack mode (i.e., only one byte per word transferred).

g. Parity Error/Overrun (bit 6). This bit when a "1" indicates that either a parity error was detected on data being received from the external device or an "overrun" condition occurred. An overrun occurs if a data word was not transmitted to the IOP before a new character is received from the external device.

h. Transfer Error (bit 7). This bit if a "1" indicates that a transmission error has been detected by the IOP.

i. Bit 8. SYNCHRONOUS: Not used, always a logic "0". ASYNCHRONOUS: Break is asserted when a break signal is detected at the receiver. The receiver detects the break by sensing a start bit and the absence of a stop bit.

j. Associate Character Detect (bit 9). This bit when a "1" indicates that a character received from the external device matches one of the associate characters stored by the controller (refer to paragraph 2-38).

k. Device Status Lines S0-S5 (bits 10:15). These bits are asserted directly by the modem and their meanings are, therefore modem dependent. (Modem status lines are configured by internal wiring of the interconnecting cable connector hood.)

Device status bits S0-S3 can be programmed to cause interrupts (refer to paragraph 2-37).

2-40. Data Formats

There are two basic data formats associated with the controller, packed and unpacked. With the packed data format, two data bytes (characters) are transferred in one data word, the most significant byte being in bits 0:7. With the unpacked data format, only one byte is transferred in a data word, the byte being in bits 8:15. Data formats are illustrated in figure 2-1. Illustrated are the packed and unpacked formats as well as examples of data words containing characters other than the standard 8-bit ASCII characters.

2-41. INTERRUPTS

The interrupt circuitry on the controller becomes enabled when a Set Mask command is issued, setting the controller Mask FF, and an interrupt control word is issued with bit four asserted (refer to paragraph 2-33). (The Mask FF is also set when an I/O RESET is received.) If the interrupt circuitry is enabled, the controller issues an interrupt request to the IOP if any of the following conditions occur:

a. A SIN (set interrupt) instruction or an interrupt order is issued to the controller.

b. A transfer error, receive parity error, or data overrun is detected.

c. BREAK. (Asynchronous mode only). When BREAK is received, an interrupt will be generated.

d. A data character is received from the external device and receive interrupts have been enabled by receive/transmit control word bit seven.

e. The controller needs a data character for transfer to the external device and transmit interrupts have been enabled by receive/transmit control word bit six (refer to paragraph 2-37).

f. A character is received that matches an associate character being stored by the controller and a function associated with that character is interrupt (refer to paragraph 2-38).

g. The state of a status line (i.e., S0-S3) that has been enabled for interrupt by receive/transmit control word bits 8:11 is changed by the modem (refer to paragraph 2-37).

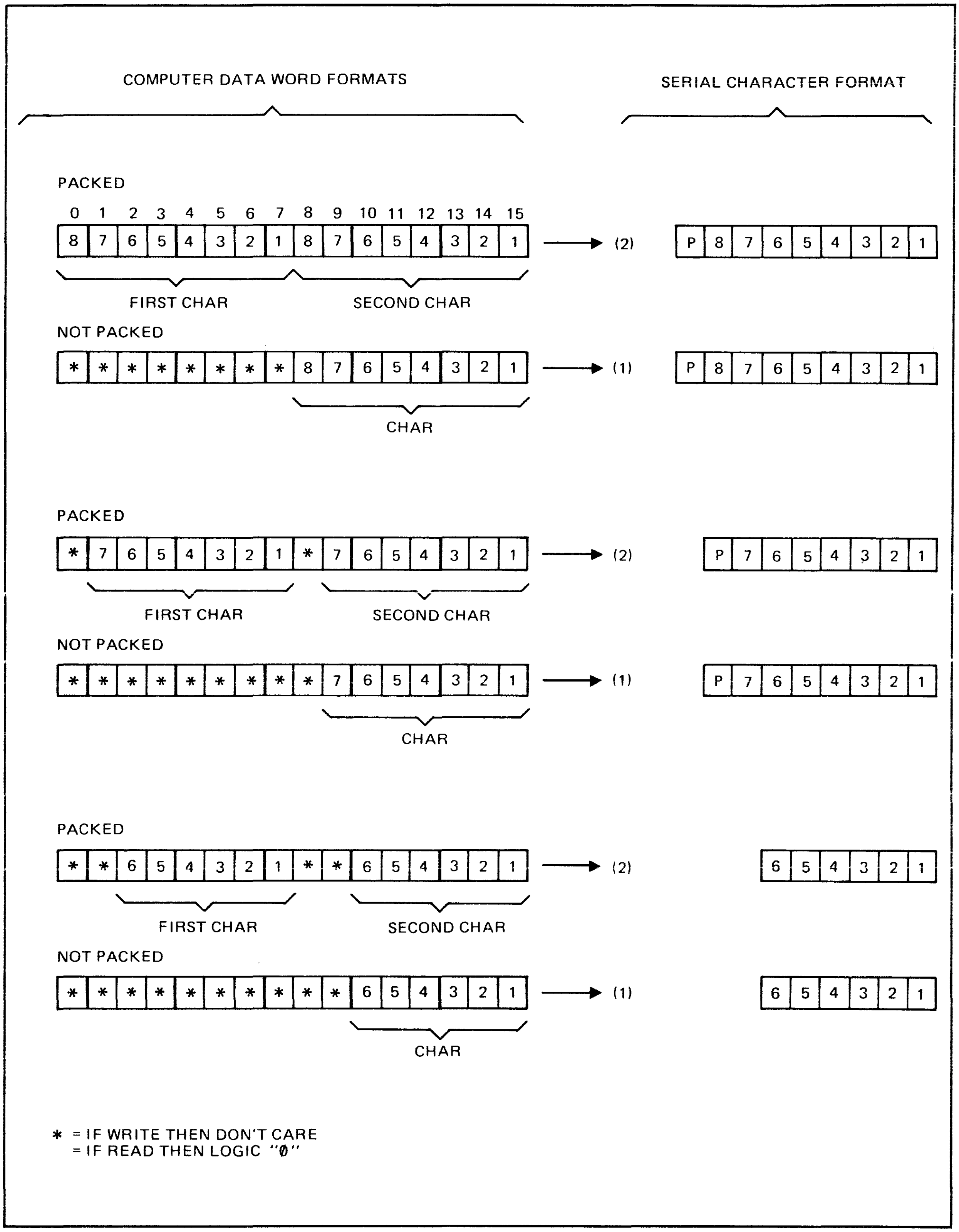


Figure 2-1. Data Formats

2-42. DATA TRANSFER

Data transfer is under control of the I/O driver associated with the controller and external device type. The driver uses CPU instructions to initialize the controller and I/O programs to control the actual transfer of data. Two sample I/O programs are described here, one for receiving data and one for transmitting data. Keep in mind that the controller PCA hardware must be configured by installation of the appropriate interconnecting cable.

2-43. Receive

A typical I/O program to receive synchronous data is listed in table 2-5. The first part of the I/O program clears the controller receiver section, sends the transmit baud to the controller, sets the character length, sends control information to the modem, and puts the controller into the "hunt" mode. In the hunt mode, service request signals and interrupt requests are disabled until the character associated with the disable service request function is received from the external device. The hunt mode is initialized by a Reset Receiver control word, an ASCII "SYNC" associate character and disable service request associated function, and a Read order. At this point, the controller compares incoming data from the external device with the associated "SYNC" character. When a SYNC character is received, "character sync" is established, and further characters are handled as data. In the case of the sample I/O program a Read of one word is specified.

The next section of the sample receive program ensures that at least two "SYNC" characters are received from the external device before any other data is received (this ensures that a legitimate data block is being received). A "SYNC" associate character is again issued to the controller, this time associated with jump conditionally, disable service request, and device end functions. Then a Read order of one word is executed. If a character other than SYNC is received, the Read order terminates but the subsequent Jump Conditional order fails and the I/O program returns to the beginning of the "hunt" mode. If a "SYNC" character is received, the controller issues a device end signal to the multiplexer channel. The next order executed is then a jump conditional, and because a "SYNC" character is read the jump condition is met.

The next section of the I/O program prepares the controller to recognize the actual beginning of the data portion of the block being received from the external device. An associate character control word is issued with a "SYNC" character associated with the disable service request function (this causes further SYNC characters to be disregarded). Then two more associate character control words are issued to the controller: 1) an SOH (start of header) and 2) an STX character (start of text). These two characters are associated with device end and conditional jump functions. Then if the next character is not a "SYNC" character and is either an SOH or STX character the I/O program begins the actual transfer of data. If the next character is not SYNC, SOH, or STX the data block is considered invalid and the I/O program terminates.

Prior to the actual transfer of data to memory from the external device, the controller receives three more associate characters: 1) ETB (end of block, associated with device end), 2) ETX (end of text, associated with device end), and 3) ENQ (associated with device end and conditional jump). A Read order specifying 500 words to be transferred is executed. Data is transferred to memory until one of the associated characters is received from the external device or 500 words have been transferred to memory. If an ENQ character is received, the data block was terminated prematurely by the external device and the I/O program terminates immediately. Otherwise, when the ETB or ETX character is received, the Read order is terminated by the device end signal from the controller or the multiplexer channel. A return residue order is then executed to obain a count of the actual number of words transferred. I/O program execution is then terminated by an End order (obtaining controller status and interrupting the CPU).

2-44. Transmit

A typical I/O program to transmit synchronous data is listed in table 2-6. The two control orders initialize the controller and transmit control inforamtion to the external device. The first write order transmits four SYNC characters to the external device. The first write order transmits four SYNC characters to the external device. The second write order transmits the actual data to the external device (including header characters -SOH or STX- and the block check character -BCC, all being supplied by software). Finally, an End order obtains controller status and interrupts the CPU.

Table 2-5. Typical I/O Program to Receive Synchronous Data

Item	Code(%)	Order	Comment
1.	040000	Control-	
	110000	Reset Receiver	
2.	040000	Control-	
	017537	Xmit Baud Rate	2400 Baud
3.	040000	Control-	
	023004	Data Format & Cont	Odd Parity, Xmit Unpack
4.	040000	Control-	
	030121	Rec/Xmit Cont	Control to Modem
5.	040000	Control-	
	110000	Reset Receiver	
6.	040000	Control-	
	042426	Assoc. Char	SYNC = DEV END/DIS SR
7.	077777	Read -One word	
	Buf 1	Buffer Address	Ignore non SYCN characters

Items "1" through "7" ignore leading non SYNC characters and detect the first SYNC character, enabling the controller logic for further data transfer.

Item	Code(%)	Order	Comment
8.	040000	Control-	
	046426	Assoc. Char	SYNC = JMP COND/DIS SR/ DEV END
9.	07777	Read-One Word	
	Buf 2	Buffer 2 Address	
10.	004000	Jump- Cond	
	*+3		Jump to item 12 if second SYNC detected
11.	000000	Jump- Uncond	
	*-15		Jump to item 4 if second SYNC not detected (restart "hunt")
12.	040000	Control-	SYNC = DSI SR (ignore further SYNCs)
	040426	Assoc. Char	
13.	040000	Control-	
	056001	Assoc. Char	SOH = DEV END/COND JMP (enable search for start of data)

Table 2-5. Typical I/O Program to Receive Synchronous Data (Continued)

Item	Code(%)	Order	Comment
14.	040000 066002	Control- Assoc. Char	STX = DEV END/COND JMP (enable search for start of data)
15.	077777 Buf 1	Read- One word Buffer 3 address	
16.	004000 *+3	Jump- Cond	Jump to item 18 to start actual data transfer if SOH or STX detected
17.	000000 *+17	Jump- Uncond	Jump to item 26 to end I/O program if SYNC, SOH, or STX not detected
18.	040000 023204	Control- Data Fmt & Cont	Receive Pack

Items "8" through "18" detect the second SYNC character, detect the header character (i.e., SOH or STX), and enable the controller for packing the subsequent data.

Item	Code(%)	Order	Comment
19.	040000 052027	Control- Assoc. Char	ETB = DEV END (enable search for end of data)
20.	040000 062003	Control- Assoc. Char	ETX = DEV END (enable search for end of data)
21.	040000 076005	Control- Assoc. Char	ENQ = COND JMP/DEV END (enable search for premature block termination)
22.	077014 Buf 4	Read- 500 Words Buffer 4 address	1000 Characters
23.	004000 *+5	Jump- Cond	Jump to item 26 to end I/O program if data block terminated prematurely by an ENQ character

Table 2-5. Typical I/O Program to Receive Synchronous Data (Continued)

Item	Code(%)	Order	Comment
24.	010000	Return Residue	
	Residue		Obtain count of actual data read if normal block termination (i.e., ETB or ETX)
25.	077777	Read- one word	
	Buf 5	Buffer 5 Address	Read the BCC (block check character)
26.	034000	End- w/interrupt	
	Status		Interrupt CPU

The end order terminates the I/O program with the five buffers containing the following information:

Buffer 1	SYNC character
Buffer 2	SYNC character
Buffer 3	Header character (i.e., SOH or STX)
Buffer 4	Data (up to 1000 characters packed)
Buffer 5	BCC character

Table 2-6. Typical I/O Program to Transmit Synchronous Data

Item	Code	Order	Comment
1.	040000	Control-	
	023204	Data Fmt & Cont	Odd parity, Xmit Unpack
2.	040000	Control-	
	030125	Rec/Xmit Cont	Control to modem
3.	067776	Write- Two words	
	Buf 6	Buffer 6 Address	Transmits four SYNC characters
4.	06xxxx	Write- block length	
	Buf 7	Buffer 7 Address	Transmit data block to modem
5.	034000	End- w/interrupt	
	Status		Interrupt CPU

THEORY OF OPERATION

SECTION III

3-1. INTRODUCTION

This section contains system-level and block-level descriptions of the HP 30055A Synchronous Single-Line Controller PCA. The system-level description briefly describes controller PCA operation in relation to the HP 3000 Computer System. The block-level description separates the controller PCA into a number of functional logic groups then describes the operation of each group.

3-2. SYSTEM-LEVEL DESCRIPTION

The controller PCA is combined with a Bell System 201, 208, or 209 modem (or equivalent) or to an asynchronous repeater to form a subsystem of the HP 3000 Computer System (see figure 3-1). The component parts of the computer system involved in the operation of the controller PCA and associated modem or asynchronous repeater are:

a. Memory.

b. CPU (central processor unit).

c. IOP (input/output processor).

d. Multiplexer channel.

e. Synchronous single-line controller.

f. Modem or repeater.

3-3. Memory

Located in memory are four groups of information used in an I/O operation: 1) the software and programs that initiate an I/O operation and check the outcome of that operation (i.e., the I/O driver), 2) the device reference table (DRT) that points the hardware to the I/O and interrupt programs (the I/O program contains I/O orders, the interrupt program contains CPU instructions), 3) the I/O program that controls the operation of the controller PCA and modem and, 4) the buffer area used as the pickup and storage during data transfers between the external device and memory.

3-4. CPU

The CPU communicates with memory, through a port shared with the IOP, via the CTL DATA BUS. The CPU fetches, then executes, instructions stored in memory. To implement an instruction, the CPU executes a number of microinstructions. It is the microinstructions that actually control the gates, flip-flops, register, etc. that comprise the CPU. The execution of I/O instructions causes microinstructions to be executed that control the operation of the IOP. (Microinstructions are also used to handle interrupt conditions in the CPU.)

3-5. IOP

The IOP issues direct commands to device controllers as directed by the CPU microinstruction set. The IOP communicates with device controllers via the IOP BUS and wih memory through the port common with the CPU via the CTL DATA BUS. The IOP resolves priorities if more than one IOP BUS transfer is being attempted simultaneously and signals the CPU if any device controller is interrupting. If the CPU and the IOP attempt to communicate with memory at the same time, the IOP has priority over the CPU.

3-6. Multiplexer Channel

The multiplexer channel permits data transfers from up to 16 devices to be multiplexed over the IOP bus on a word-at-a-time basis. Operation of the multiplexer channel and therefore the controller PCA is under control of the I/O program stored in memory associated with a specific device controller.

Execution of the I/O program is initiated when the CPU encounters an SIO instruction and issues a direct SIO command through the IOP to the specified device controller. The device controller upon receipt of the SIO command signals the multiplexer channel to begin fetching and executing the I/O program words. The multiplexer channel then signals the CPU that I/O program execution has been initiated leaving the CPU free to continue processing. Next, the multiplexer channel directs the device controller to send the address of its DRT location in memory to the IOP. Using that address, the multiplexer channel fetches the I/O program pointer from the DRT first word. The I/O program pointer, that points to the next (initially the first) I/O program word to be executed, is temporarily stored in a register in the multiplexer channel. Also, the multiplexer channel increments the I/O program pointer by two then returns the pointer to the DRT first word (so the next DRT fetch will point to the next I/O program order to be executed).

The I/O program word specified by the I/O program pointer is then fetched and subsequently stored in a register in the multiplexer channel (the multiplexer channel contains 16 such registers, one for each possible device connected to a multiplexer channel). The order portion of this word (i.e., the IOCW) is decoded by the multiplexer channel and the appropriate control signals are sent to the device controller via the MUX CHAN BUS. The device controller acknowledge the control signal(s) by performing the appropriate action and issuing an acknowledge signal to the IOP (informing the IOP that the IOCW transfer is completed). Depend-

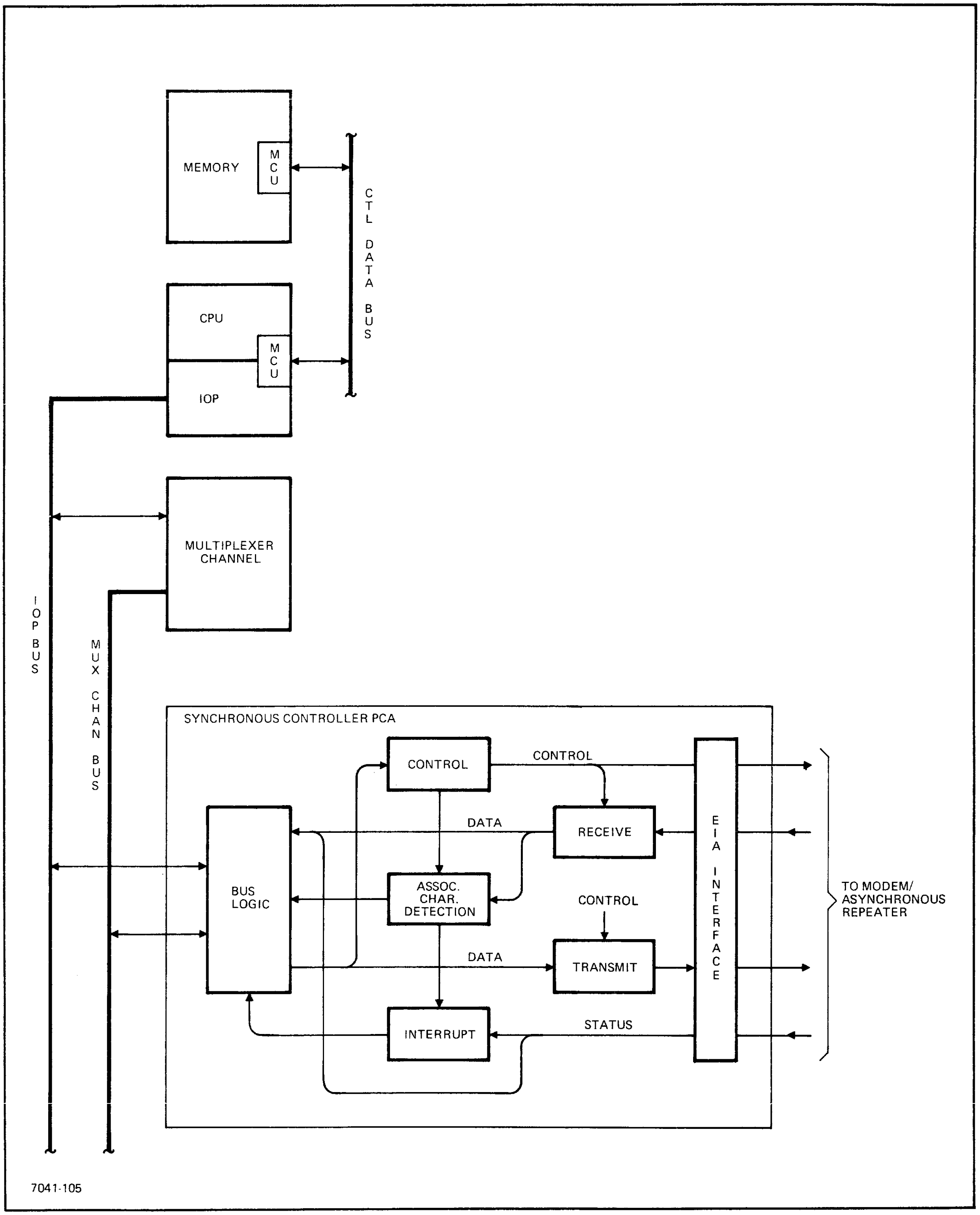


Figure 3-1. System Block Diagram

ing upon the specific order being executed a transfer to or from the IOAW location in memory is initiated. If a control order, a control word from the IOAW location is issued to the device controller along with the appropriate control signal. If a read or write order, the IOAW contains the address of the data buffer in memory. That address is stored in a register in the multiplexer channel (the multiplexer channel also contains one of these registers for each of the 16 possible devices). If a sense or end order, the multiplexer channel orders the device controller to send its status word to the IOP. The status word is stored in the IOAW location in memory. If a jump order, the IOAW location contains the address of the next I/O order to be executed if the jump condition is met. That address is used as the I/O program pointer for the next DRT fetch (and as a result the jump address plus two is stored in the DRT first word).

The sequence of obtaining the I/O program pointer from the DRT first word (i.e., DRT fetch) then using that address to fetch the I/O program words (i.e., IOCW fetch and IOAW fetch) occurs for each I/O program order in the I/O program.

3-7. Controller PCA

The controller PCA provides the means of converting commands, control signals, and control word resulting from the execution of I/O instructions or an I/O program into the signals necessary to control a modem. The controller PCA converts data between the 16-bit parallel data from the computer systems and the bit-serial data required by the modem. Additionally, the controller PCA has the ability to compare characters being received from the external device with up to four pre-specified characters and take some related action if they match.

The controller PCA is programmed through the use of 12 control words. These control words specify such things as the baud rate, character length, control signals to the modem, and characters to be checked against data received from the external device (i.e., associate character recognition). Data and control words are transferred between the controller PCA and the IOP via the IOP BUS. Control signals resulting from the execution of an I/O program are transmitted to the controller PCA via the MUX CHAN BUS. Control, status, and data signals are transferred between the controller PCA and the modem via the interconnecting cable. Certain functions of the controller PCA are also programmed through internal jumpering of the interconnecting cable hood.

Upon command, the controller PCA provides a 16-bit status word to the IOP. The status word indicates the internal state of the controller PCA as well as the states of status lines from the device. (The meanings of the status lines from the modem are dependent on the particular device and internal wiring of the interconnecting cable hood.)

3-8. Modem

The modem converts digital signals from the controller PCA to analog signals for transmission to another modem. Conversely, when receiving from another modem the modem converts the analog signals to the digital signals acceptable by the controller PCA. The modem provides a clock signal to the controller PCA as well as status information such as "ring indicator" (the modem is being called) and "carrier detect" (a modem is on the other end of the line).

3-9. BLOCK-LEVEL DESCRIPTION

The HP 30055A Synchronous Single-Line Controller consists of the seven interrelated functional logic groups shown in figure 3-1. These groups are: bus logic, control, receive associate character recognition, transmit, interrupt, and interfacing circuits. Figure 3-2 illustrates the functional logic groups at a more detailed level.

3-10. Bus Logic

The controller PCA bus logic is functionally identical to bus logic used by other HP 3000 controllers on the multiplexer channel. That is, the bus logic decodes the direct commands from the IOP and issues the appropriate responses to the IOP and the proper control signals to the controller PCA circuitry. Additionally, the bus logic controls the application of signals from the MUX CHAN BUS to the controller PCA circuitry. The bus logic also contains circuitry to process interrupt requests from the controller PCA interrupt logic, and controls the transfer of control, data, and status between the IOP BUS and the controller PCA logic circuits.

3-11. Control

The control logic stores, and makes available to the remainder of the controller PCA logic and to the modem or asynchronous repeater, signals representing the various control word configurations. When a control word is received as the result of a CIO instruction or a control order, bits 0:3 of the control word are decoded by the control word address decoder. The result determines which of the various registers the control word bit configurations is stored. (Refer to section II for meanings of the control word configurations.)

3-12. Receive

Functionally, the receive logic converts the bit-serial data received from the modem or asynchronous repeater into a 16-bit word to be transferred to the IOP.

Bit-serial data from a modem is input to the de-serializer. Data is clocked into the de-serializer by a receive clock from the external device. The number of sequential bits considered to be a single character is determined by the character length specified in the data format and control word. The data is converted into an eight-bit parallel character by the de-serializer. In asynchronous mode, the start and stop bits are stripped from the data. The read state counter (also clocked by the modem) issues various signals that control the timing of receive and associate character recognition logic.

Initially, during synchronous mode, (i.e., after a CLEAR INTERRUPT AND RECEIVER LOGIC control word or I/O reset is received) the controller PCA is in a "hunt" mode. Functionally, this means that no data will be transferred until a pre-specified character is received from the external device putting the controller in "character sync". Typically this character is an ASCII "SYNC" character and is specified by an associate character and function control word. In asynchronous mode, characters are "framed" by the reception of start and stop bits. The associated function (refer to paragraph 2-38) is disable service request. Additional data (other than SYNC characters will then be transferred by the controller PCA. (Additionally, any further characters associated with a disable service request function will not be transferred.)

If "receive pack" is specified by the data format and control word, then the BYTE FF is toggled with each byte assembled by the deserializer. The state of the BYTE FF determines which eight-bit receive buffer register the newly assembled byte is stored. In this manner a 16-bit word is assembled from two eight-bit characters. The first character is stored in the register corresponding to the eight most significant bits of data word (i.e., bits 0:7), the second character is stored in the register that corresponds to the least significant eight-bits. If packing is not specified, then the BYTE FF does not toggle and the eight-bit character is stored in the least significant eight bits. The receive state counter then signals the bus logic that the data word is ready to be transferred to the IOP (by interrupting the CPU which subsequently issues an RIO instruction) or by a currently pending read order.

Additionally, the receive logic does the following: checks the parity of the incoming data according to bit five of the data format and control word; signals the interrupt circuitry if a data word is ready for transfer (this is typically used during direct I/O transfers only); and signals the interrupt circuitry if data has been lost due to an overrun condition. During asynchronous mode, it also detects break conditions.

3-13. Associate Character Detection

The associate character detection circuitry works in conjunction with the receive logic to compare data being received from an external device with characters stored on the controller PCA.

The comparison character and the associated function are stored in associative memories (also called content addressable memories, or CAMs) by an associate character and function control word (up to four comparison characters and associated functions are permitted). To determine if a character being received matches a character in the associative memory, that character is presented at the input to the associative memory along with a compare strobe (supplied by the receive state counter). If the character matches, the address line corresponding to the storage location of the associate character (i.e., 0:3) is asserted. The asserted line addresses another associative memory containing bits representing the associated function. The configuration on the output of this memory is strobed into a four-bit register by a signal from the receive state counter.

The four possible associated functions are jump, assert device end, interrupt, and disable service request (a particular character can be associated with any or all functions as desired):

a. Jump. If this function is specified, the associated character is matched with a character from the external device, and a conditional Jump order is issued to the controller PCA, a JMP MET signal is issued to the multiplexer channel during the next DRT fetch sequence. This results in multiplexer fetching and executing the order specified by the Jump order IOAW contents.

b. Assert device end. If this function is specified and the associated character is received from the external device, a DEV END signal is asserted to the muliplexer channel. This results in the current Read order being terminated after the present data word is transferred to the IOP.

c. Interrupt. If this function is specified, the associated character matched with a character from the external device, and the controller PCA interrupt circuitry is enabled, an INT REQ signal is isued to the IOP. If the CPU is accepting external interupts, the CPU is interrupted.

d. Disable service request. This function actually has two purposes. One is to put the controller PCA into "character sync" as described in paragraph 3-12 (synchronous mode only). The other function is to cause the controller PCA to ignore some specified character.

3-14. Transmit

Functionally, the transmit logic converts a 16-bit data word from the IOP into one or two bit-serial characters and transmits the characters to the modem or asynchronous repeater.

Execution of a WIO instruction or a write order loads a 16-bit data word from the IOP into the transmit buffers and activates the transmit logic. The transmit state counter, which is specified by the data format and control control word, then the UNPACK

FF is toggled with each character to be transferred to the serializer. The first character to be serialized is taken from the transmit buffer corresponding to the most significant portion of the data word (i.e., bits 0:7). The second character to be serialized, or all characters if unpack mode is not specified, is taken from the buffer corresponding to the least significant portion of the data word.

The number of bits considered to be a single character by the serializer is determined by the data format and control word. The serializer converts the character from the transmit buffer into a bit-serial data stream. After leaving the serializer during synchronous mode, a parity bit is appended to the bit stream, and data is encoded into the format required by the modem. (Both the serializer and the encoder flip-flops are synchronized with a clock from the modem.) The bit-serial data from the encoder is then routed through the interfacing circuits to the modem. After leaving the serializer during asynchronous mode, start, stop, and parity bits are appended to the bit stream and the data is encoded into the required format.

An additional function of the transmit logic, if specified by the data format and control control word, is to "echo" data back to an external device. This is accomplished by simply directing the incoming bit-serial data back to the modem.

3-15. Interrupt

Functionally, the interrupt circuitry, when enabled by the appropriate control words, monitors signals from the controller PCA and modem to determine if an interrupt is being requested.

There are three main classes of interrupts associated with the controller PCA:

a. Data interrupt. A data interrupt request occurs if either an associate characer is detected and the associated function is interrupt; the controller PCA is writing, another data character is required, and transmit interrupts are enabled by the appropriate receive/transmit control word (this is typically used for direct write I/O only); the controller PCA is reading, a word is ready for transfer to the IOP, and receive interrupts are enabled by the appropriate receive/transmit control word (typically used for direct read I/O only); or, during asynchronous mode, when a break signal is detected at the receiver.

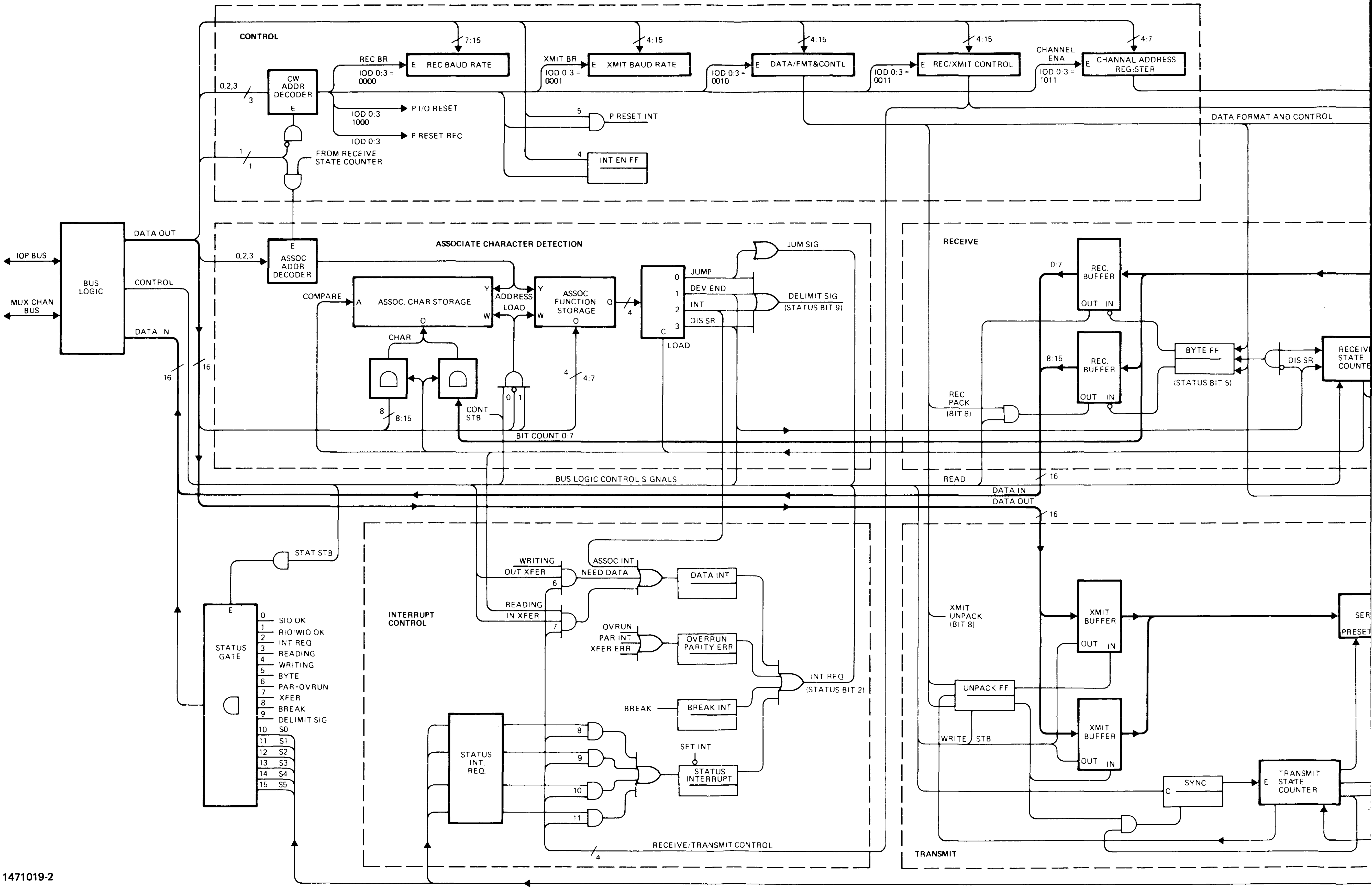
b. Overrun/Parity Error. This interrupt request occurs if a data overrun occurred with data from the external device, a parity error was detected in the data from the external device, or a transmission error signal has been received from the multiplexer channel.

c. Status Interrupt. A status interrupt request occurs if a status line from the device (as configured by the connector cable) changes states and that line is enabled for interrupts by the appropriate receive/transmit control word.

If the interrupt circuitry requests an interrupt and the interrupt control circuitry in the bus logic is enabled (i.e., appropriate interrupt control word received and Mask FF set) an INT REQ signal is asserted to the IOP. If the CPU is accepting external interrupts, CPU processing is interrupted and the CPU begins executing code asociated with the interrupt.

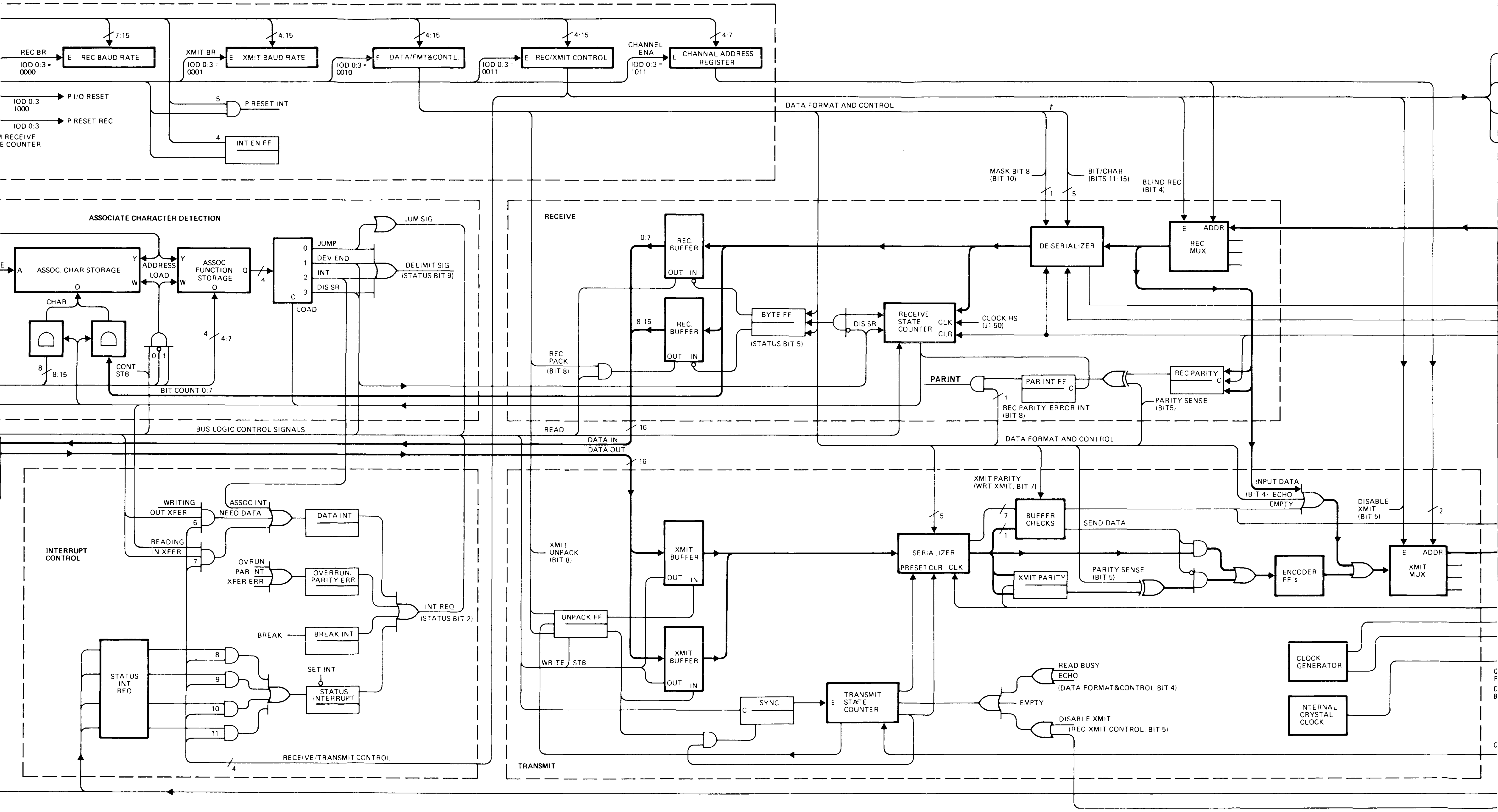
3-16. Interfacing Circuits

The interfacing circuits provide for interfacing the controller PCA TTL logic levels to the logic levels required by the device it will be connected to. The actual configuration of the interfacing circuits is determined by internal jumpering of the interconnecting cable connector hood. Refer to figure 4-2 for connection to EIA RS-232-C type synchronous devices and to figure 4-3 for connection to hardwired devices.



1471019-2

AUG 1978


7:15
4:15
4:15
4:15
4:7
REC BR
E REC BAUD RATE
IOD 0:3 = 0000
XMIT BR
E XMIT BAUD RATE
IOD 0:3 = 0001
E DATA/FMT&CONTL
IOD 0:3 = 0010
E REC/XMIT CONTROL
IOD 0:3 = 0011
CHANNEL ENA
E CHANNAL ADDRESS REGISTER
IOD 0:3 = 1011
IOD 0:3 1000
P I/O RESET
5
P RESET INT
IOD 0:3
P RESET REC
I RECEIVE
E COUNTER
4
INT EN FF
DATA FORMAT AND CONTROL
MASK BIT 8 (BIT 10)
BIT/CHAR (BITS 11:15)
BLIND REC (BIT 4)
1
5
ASSOCIATE CHARACTER DETECTION
E
A ASSOC. CHAR STORAGE
Y
W
O
ADDRESS
LOAD
Y
ASSOC FUNCTION STORAGE
W
O
Q
4
0
1
2
3
C
JUMP
DEV END
INT
DIS SR
LOAD
JUM SIG
DELIMIT SIG (STATUS BIT 9)
CHAR
4
4:7
8
8:15
0
1
CONT STB
BIT COUNT 0:7
BUS LOGIC CONTROL SIGNALS
RECEIVE
0:7
REC. BUFFER
OUT IN
8:15
REC BUFFER
OUT IN
REC PACK (BIT 8)
BYTE FF
(STATUS BIT 5)
DIS SR
RECEIVE STATE COUNTER
CLK
CLR
CLOCK HS (J1:50)
DE SERIALIZER
E ADDR
REC MUX
REC PARITY
C
PARINT
PAR INT FF
C
1
REC PARITY ERROR INT (BIT 8)
PARITY SENSE (BIT5)
READ
16
DATA IN
DATA OUT
16
DATA FORMAT AND CONTROL
INTERRUPT CONTROL
WRITING
OUT XFER
6
READING
IN XFER
7
ASSOC INT
NEED DATA
DATA INT
OVRUN
PAR INT
XFER ERR
OVERRUN PARITY ERR
BREAK
BREAK INT
INT REQ (STATUS BIT 2)
SET INT
STATUS INTERRUPT
STATUS INT REQ.
8
9
10
11
RECEIVE/TRANSMIT CONTROL
4
XMIT UNPACK (BIT 8)
XMIT BUFFER
OUT IN
UNPACK FF
XMIT BUFFER
OUT IN
WRITE
STB
SYNC
C
E TRANSMIT STATE COUNTER
TRANSMIT
XMIT PARITY (WRT XMIT, BIT 7)
5
SERIALIZER
PRESET CLR CLK
7
1
BUFFER CHECKS
SEND DATA
XMIT PARITY
PARITY SENSE (BIT 5)
ENCODER FF's
INPUT DATA
(BIT 4) ECHO
EMPTY
DISABLE XMIT (BIT 5)
2
E ADDR
XMIT MUX
READ BUSY
ECHO
(DATA FORMAT&CONTROL BIT 4)
EMPTY
DISABLE XMIT
(REC/XMIT CONTROL, BIT 5)
CLOCK GENERATOR
INTERNAL CRYSTAL CLOCK

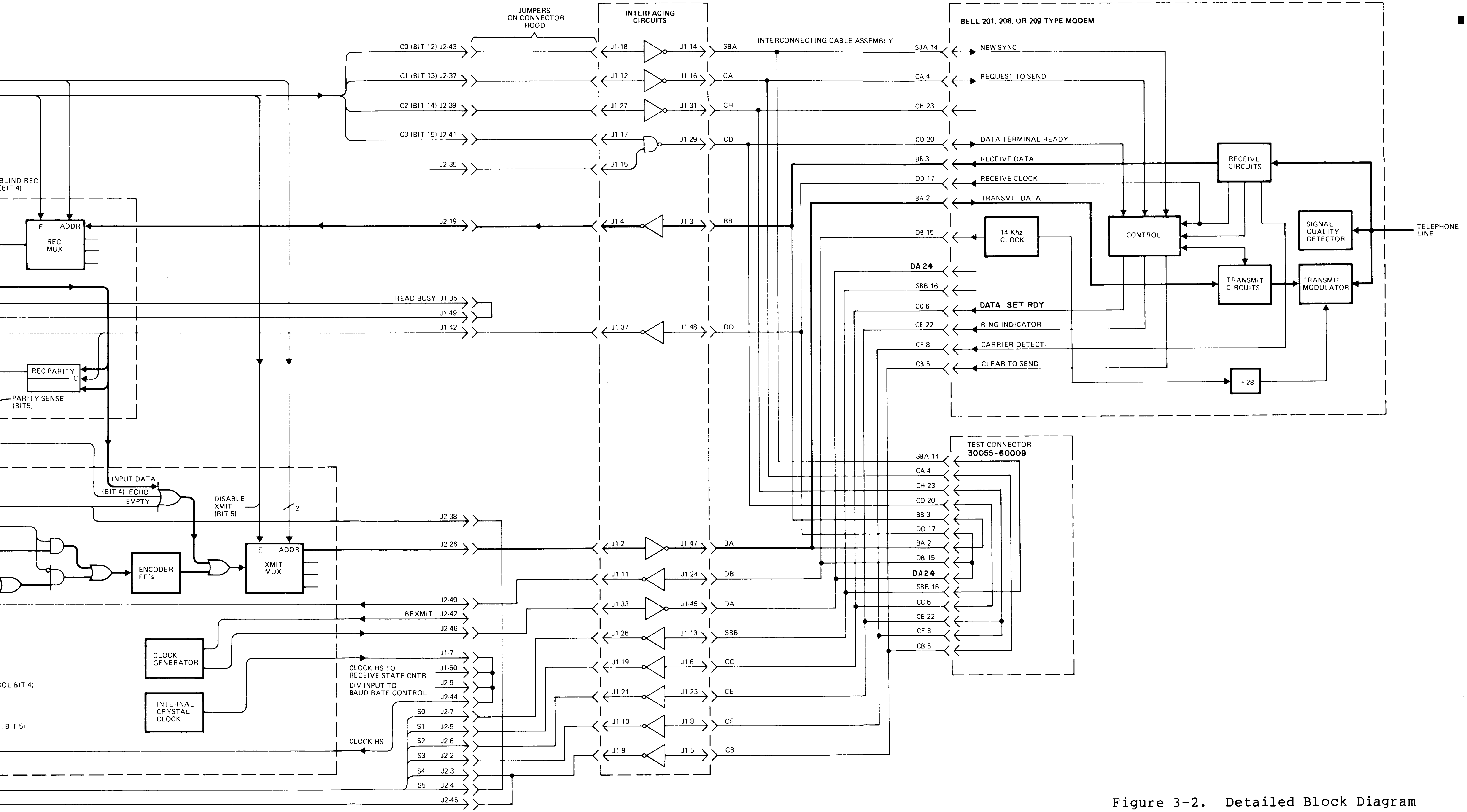
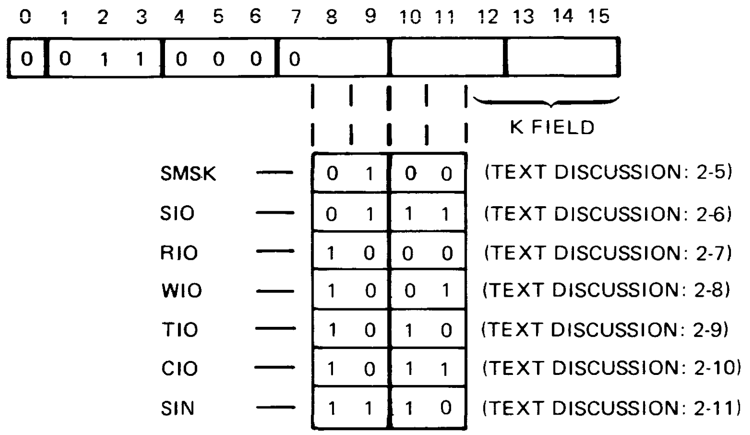


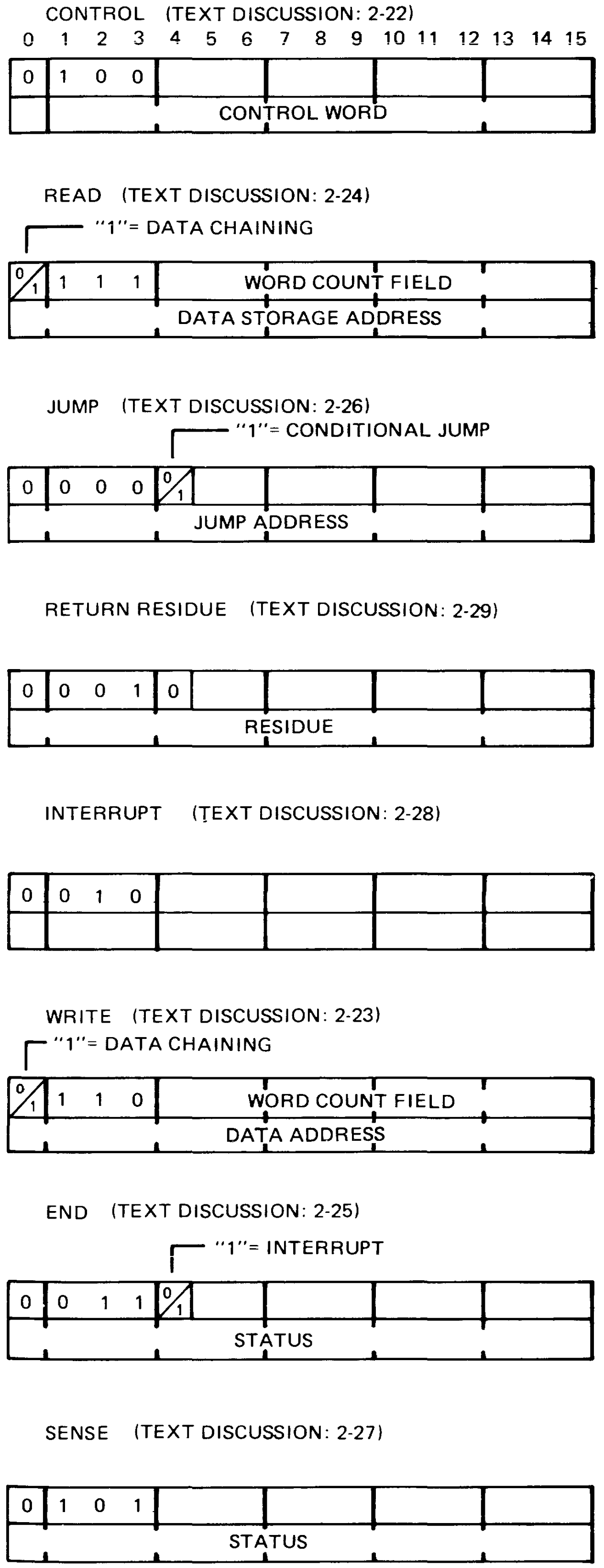
Figure 3-2. Detailed Block Diagram

I/O INSTRUCTION FORMAT



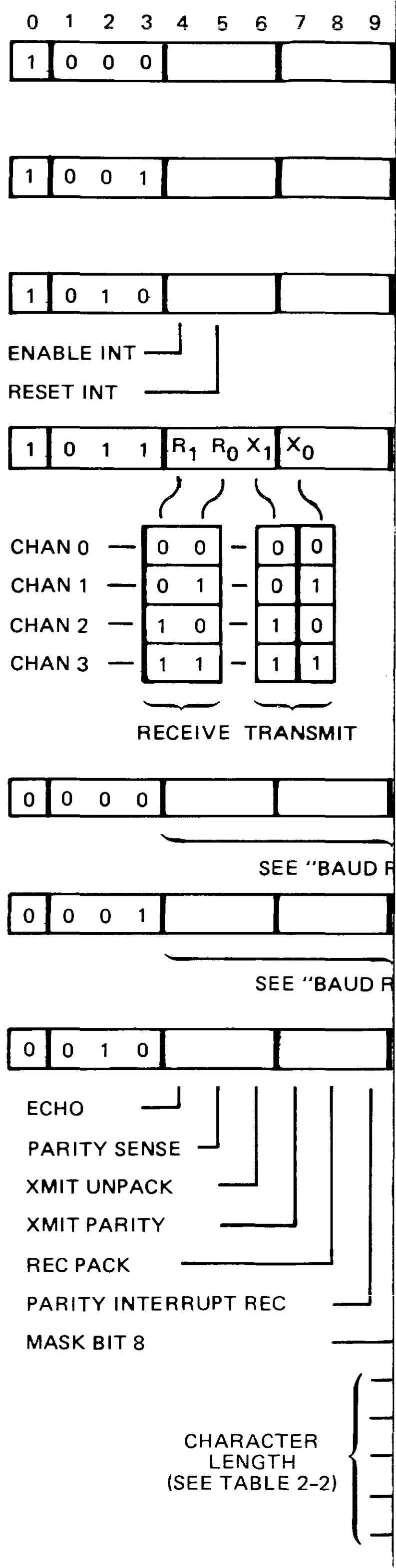
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0 0 1 1 0 0 0 0
K FIELD
SMSK — 0 1 0 0 (TEXT DISCUSSION: 2-5)
SIO — 0 1 1 1 (TEXT DISCUSSION: 2-6)
RIO — 1 0 0 0 (TEXT DISCUSSION: 2-7)
WIO — 1 0 0 1 (TEXT DISCUSSION: 2-8)
TIO — 1 0 1 0 (TEXT DISCUSSION: 2-9)
CIO — 1 0 1 1 (TEXT DISCUSSION: 2-10)
SIN — 1 1 1 0 (TEXT DISCUSSION: 2-11)

I/O ORDER FORMAT



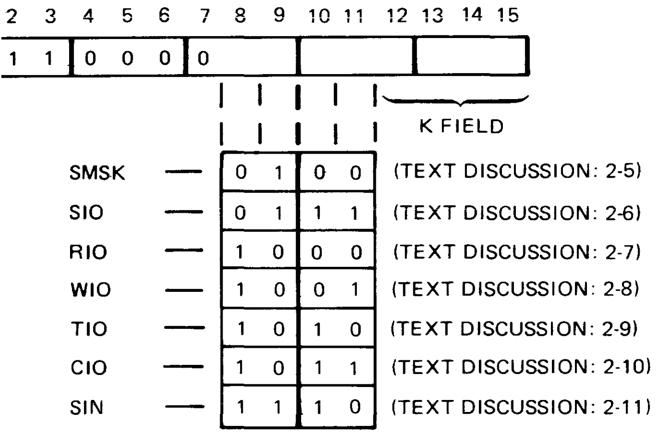
CONTROL (TEXT DISCUSSION: 2-22)
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
0 1 0 0
CONTROL WORD
READ (TEXT DISCUSSION: 2-24)
"1"= DATA CHAINING
0/1 1 1 1 WORD COUNT FIELD
DATA STORAGE ADDRESS
JUMP (TEXT DISCUSSION: 2-26)
"1"= CONDITIONAL JUMP
0 0 0 0 0/1
JUMP ADDRESS
RETURN RESIDUE (TEXT DISCUSSION: 2-29)
0 0 0 1 0
RESIDUE
INTERRUPT (TEXT DISCUSSION: 2-28)
0 0 1 0
WRITE (TEXT DISCUSSION: 2-23)
"1"= DATA CHAINING
0/1 1 1 0 WORD COUNT FIELD
DATA ADDRESS
END (TEXT DISCUSSION: 2-25)
"1"= INTERRUPT
0 0 1 1 0/1
STATUS
SENSE (TEXT DISCUSSION: 2-27)
0 1 0 1
STATUS

CONTROL WORD

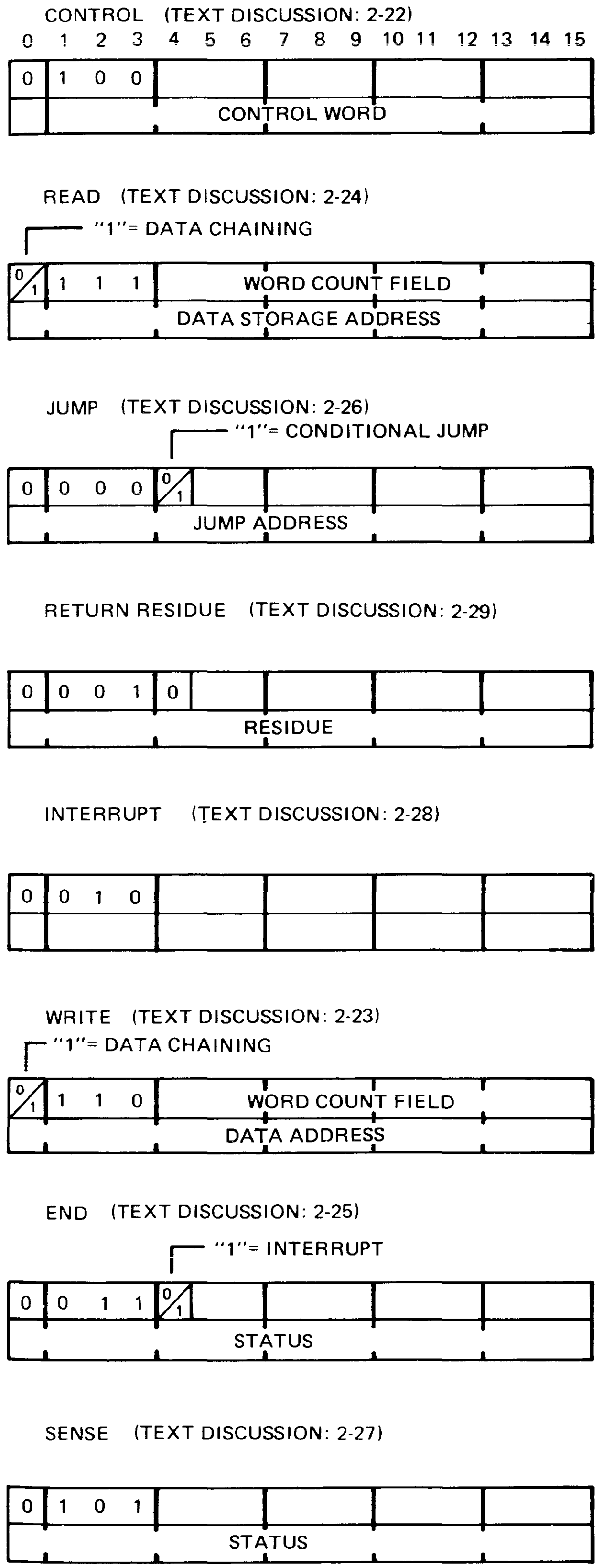


1471019-1

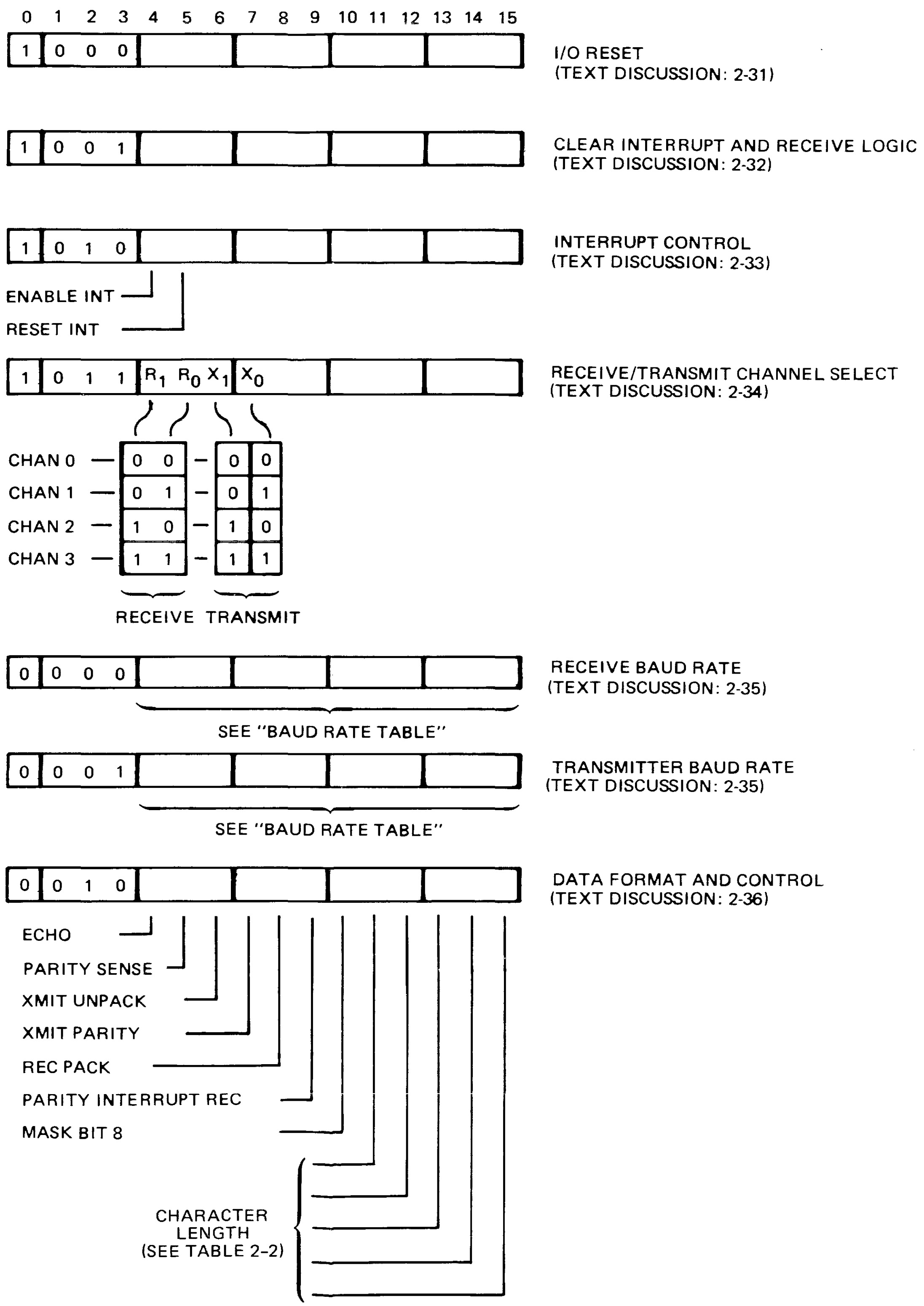
RUCTION FORMAT

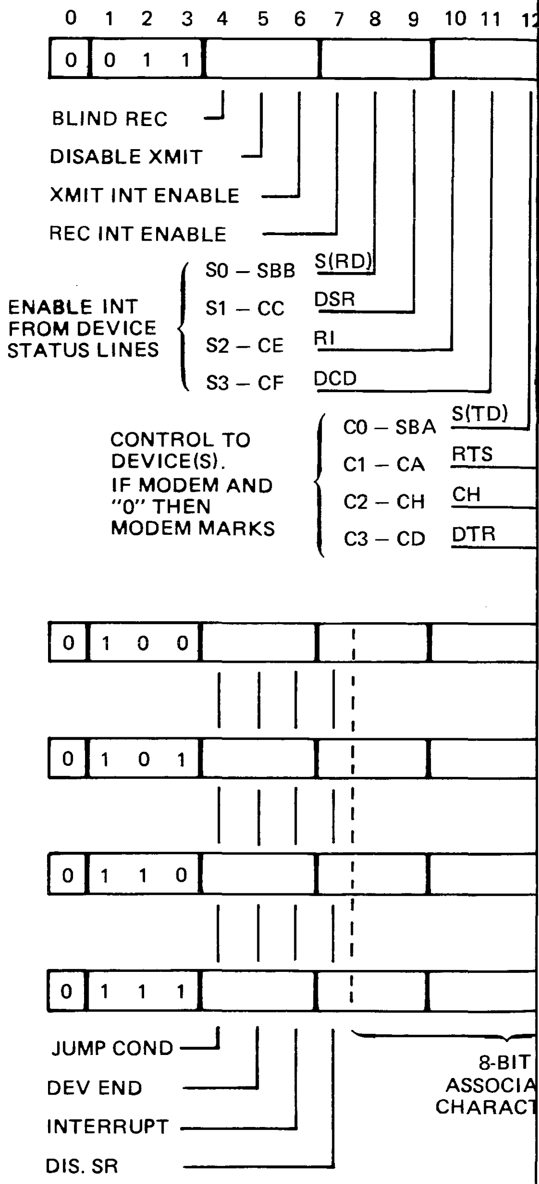

2 3 4 5 6 7 8 9 10 11 12 13 14 15
1 1 0 0 0 0
K FIELD
SMSK — 0 1 0 0 (TEXT DISCUSSION: 2-5)
SIO — 0 1 1 1 (TEXT DISCUSSION: 2-6)
RIO — 1 0 0 0 (TEXT DISCUSSION: 2-7)
WIO — 1 0 0 1 (TEXT DISCUSSION: 2-8)
TIO — 1 0 1 0 (TEXT DISCUSSION: 2-9)
CIO — 1 0 1 1 (TEXT DISCUSSION: 2-10)
SIN — 1 1 1 0 (TEXT DISCUSSION: 2-11)

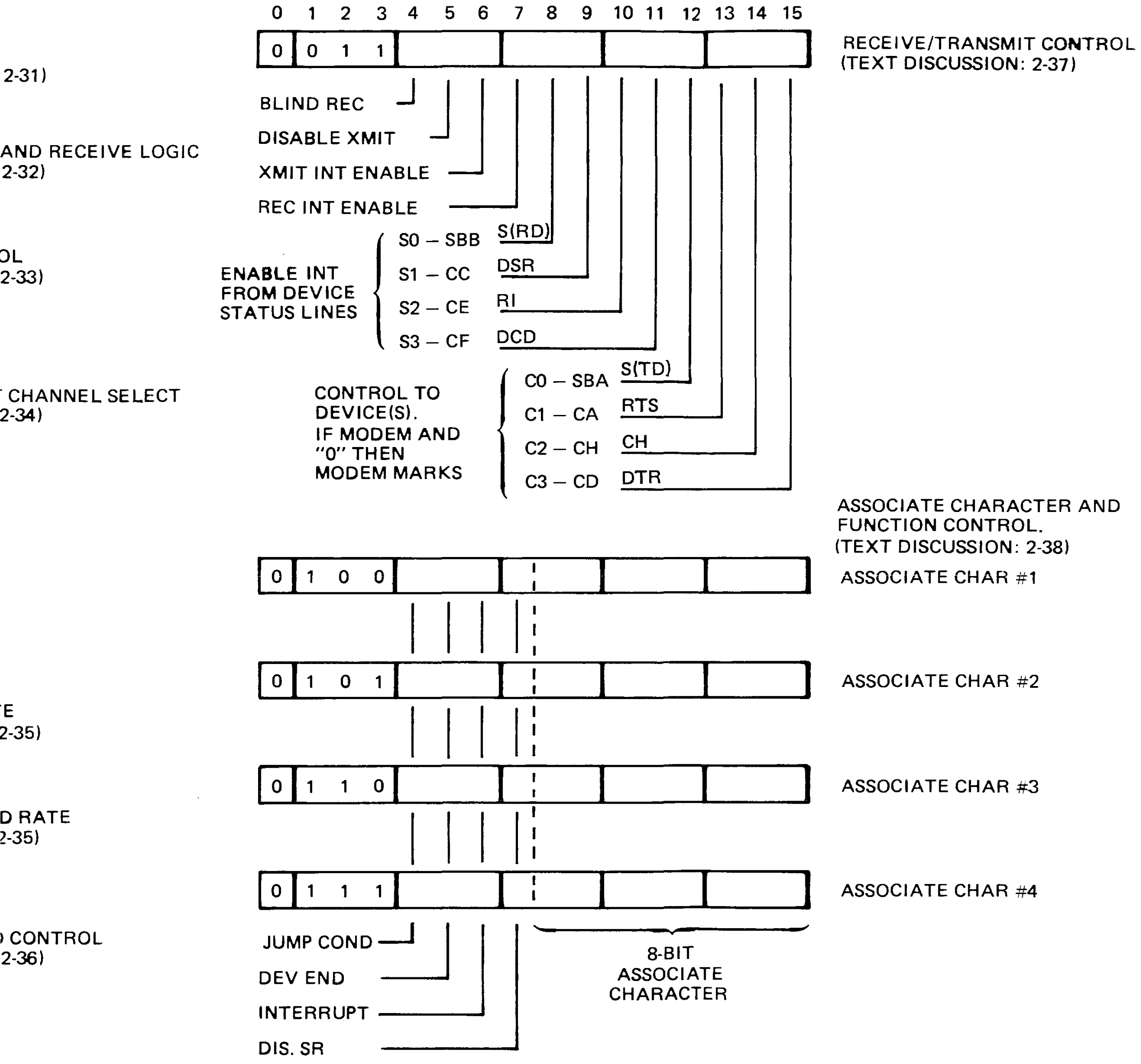
I/O ORDER FORMAT



CONTROL WORD







CONTROL WORD CONFIGURATIONS FOR COMMONLY USED BAUD RATES

BAUD RATE	CONTROL WORD (BINARY)												CONTROL WORD (OCTAL)
	4	5	6	7	8	9	10	11	12	13	14	15	
75	1	1	1	1	1	1	1	1	1	0	1	0	7772
110	0	0	1	1	0	1	1	0	0	1	1	1	1547
134.7	0	0	0	0	1	1	1	0	1	1	1	1	0357
150	1	1	1	1	1	1	1	1	0	1	0	1	7765
300	1	1	1	1	1	1	1	0	1	0	1	1	7753
600	1	1	1	1	1	1	0	1	0	1	1	1	7727
1050	0	0	0	1	0	0	1	0	1	1	1	1	0457
1200	1	1	1	1	1	0	1	0	1	1	1	1	7657
2400	1	1	1	1	0	1	0	1	1	1	1	1	7537
4800	1	1	1	0	1	0	1	1	1	1	1	1	7277
9600	1	1	0	1	0	1	1	1	1	1	1	1	6577
19200	1	0	1	0	1	1	1	1	1	1	1	1	5377
38400	0	1	0	1	1	1	1	1	1	1	1	1	2777
48000	1	1	0	1	1	1	1	1	1	1	1	1	6777
96000	1	0	0	1	1	1	1	1	1	1	1	1	4777

STATUS FORMAT

(TEXT DISCUSSION: 2-39)

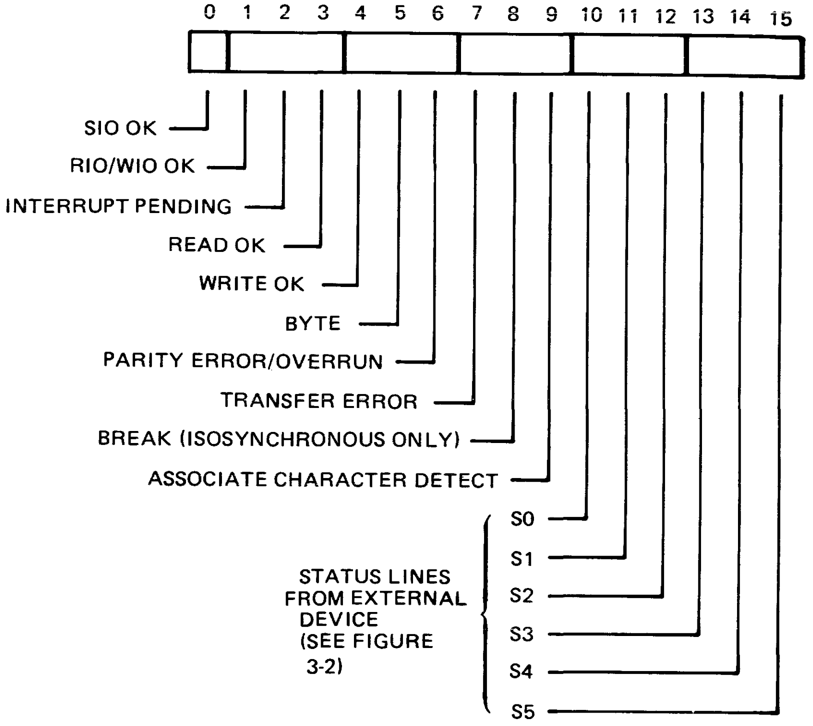


Figure 2-2. Programming Information

MAINTENANCE

SECTION IV

4-1. INTRODUCTION

This section contains the following service related information:

a. Safety precautions.

b. IOP BUS/MUX CHAN BUS signal information.

c. Interconnecting cable signal information.

d. A list of required servicing equipment.

e. A list and description of maintenance aids.

4-2. SAFETY PRECAUTIONS

When the controller PCA is being installed, removed, or placed on an extender PCA, the computer system DC POWER switch must be set to the STANDBY position to remove power from the PCA connectors.

When the controller PCA is being connected to a modem be sure that no power is present at the controller PCA connectors and that the modem power is off.

Failure to observe the precautions may result in damage to the controller PCA or computer system components.

4-3. IOP BUS/MUX CHAN BUS SIGNAL INFORMATION

Table 4-1 lists the IOP BUS and MUX CHAN BUS signals associated with controller PCA operation. (See figure 1-1 for locations of controller PCA connectors and figure 4-1 for connector numbering.) For a list of signals on a particular connector, refer to the detailed diagram set no. DD-510. The following information is provided in table 4-2:

a. The signal mnemonic.

b. The bus on which the signal is found.

c. The connector and pin number through which the signal enters the controller PCA (i.e., CONN=connector).

d. A brief description of each signal.

Table 4-1. IOP/MUX CHAN BUS Signal List

"BUS" LEGEND:

IOP = IOP Bus
MUX = Multiplexer Channel Bus
PBB = Power Bus Backplane

"CONN" LEGEND:

```
P2  07
|    |
|    |
|    |
|    |-- CONNECTOR PIN NO.
|------- CONNECTOR NO.
```

MNEMONIC	BUS	CONN	DESCRIPTION
ACK SR	MUX	P207	Acknowledge Service Request
CHAN ACK	MUX	P209	Channel acknowledge
CHAN SO	MUX	P201	Channel service out
DEV NO 00	IOP	P308	Device no., bit 0
DEV NO 01	IOP	P309	Device no., bit 1
DEV NO 02	IOP	P311	Device no., bit 2
DEV NO 03	IOP	P312	Device no., bit 3
DEV NO 04	IOP	P314	Device no., bit 4
DEV NO 05	IOP	P315	Device no., bit 5
DEV NO 06	IOP	P317	Device no., bit 6
DEV NO 07	IOP	P318	Device no., bit 7
DEV END	MUX	P205	Device end
DEVNO DB	MUX	P211	Device number to data base
IOCMD 00	IOP	P304	I/O command, bit 0
IOCMD 01	IOP	P305	I/O command, bit 1
IOCMD 02	IOP	P305	I/O command, bit 2
IOD 00	IOP	P320	I/O data, bit 0
IOD 01	IOP	P321	I/O data, bit 1
IOD 02	IOP	P323	I/O data, bit 2
IOD 03	IOP	P324	I/O data, bit 3
IOD 04	IOP	P326	I/O data, bit 4
IOD 05	IOP	P327	I/O data, bit 5
IOD 06	IOP	P329	I/O data, bit 6
IOD 07	IOP	P330	I/O data, bit 7

Table 4-1. IOP/MUX CHAN BUS Signal List (Continued)

MNEMONIC	BUS	CONN	DESCRIPTION
IOD 08	IOP	P332	I/O data, bit 8
IOD 09	IOP	P333	I/O data, bit 9
IOD 10	IOP	P335	I/O data, bit 10
IOD 11	IOP	P336	I/O data, bit 11
IOD 12	IOP	P338	I/O data, bit 12
IOD 13	IOP	P339	I/O data, bit 13
IOD 14	IOP	P341	I/O data, bit 14
IOD 15	IOP	P342	I/O data, bit 15
IOD PRTY	IOP	P301	I/O data, parity bit
IORESET	PBB	P111	I/O reset
INT ACK	IOP	P350	Interrupt acknowledge
INT POLL IN	PBB	P148	Interupt poll in
INT POLL OUT	PBB	P144	Interrupt poll out
INT REQ	IOP	P344	Interrupt request
JMP MET	MUX	P214	Jump condition met
P CMD 1	MUX	P243	Programmed command 1
P CONT STB	MUX	P250	Programmed control strobe
P READ STB	MUX	P250	Programmed read strobe
P STATUS STROBE	MUX	P248	Programmed write strobe
REQ	MUX	P222	Request for SIO initiation
SI	PBB	P149	Service in
SIO ENABLE	MUX	P212	Asserted when MUX CHAN present
SO	PBB	P153	Service Out
SR 0	MUX	P241	Service request line 0
SR 1	MUX	P240	Service request line 1
SR 2	MUX	P239	Service request line 2
SR 3	MUX	P238	Service request line 3
SR 4	MUX	P237	Service request line 4
SR 5	MUX	P235	Service request line 5
SR 6	MUX	P236	Service request line 6
SR 7	MUX	P233	Service request line 7
SR 8	MUX	P232	Service request line 8
SR 9	MUX	P231	Service request line 9
SR 10	MUX	P229	Service request line 10
SR 11	MUX	P228	Service request line 11

Table 4-1. IOP/MUX CHAN BUS Signal List (Continued)

MNEMONIC	BUS	CONN	DESCRIPTION
SR 12	MUX	P227	Service request line 12
SR 13	MUX	P226	Service request line 13
SR 14	MUX	P225	Service request line 14
SR 15	MUX	P224	Service request line 15
SET INT	MUX	P249	Set interrupt
SET JMP	MUX	P244	Set jump
TOGGLE INXFER	MUX	P216	Toggle in transfer (FF)
TOGGLE OUTXFER	MUX	P218	Toggle out transfer (FF)
TOGGLE SIO OK	MUX	P219	Toggle SIO OK (FF)
TOGGLE SR	MUX	P217	Toggle service request (FF)
XFER ERROR	MUX	P221	Transfer error

4-4. INTERCONNECTING CABLE SIGNAL INFORMATION

Figures 4-2 and 4-3 illustrate the wiring of the interconnecting cables and the internal wiring of the connector hoods. Also illustrated is the internal wiring of the test connector assembly. (Refer to paragraph 4-6 for a description of the test connector.)

4-5. REQUIRED SERVICING EQUIPMENT

The test connector assembly (supplied with the controller PCA) is required if the Stand-Alone HP 30055A Synchronous Single-Line Controller diagnostic is to be run. Other items that may prove helpful are:

a. An extender PCA.

b. A logic probe.

c. A logic clip.

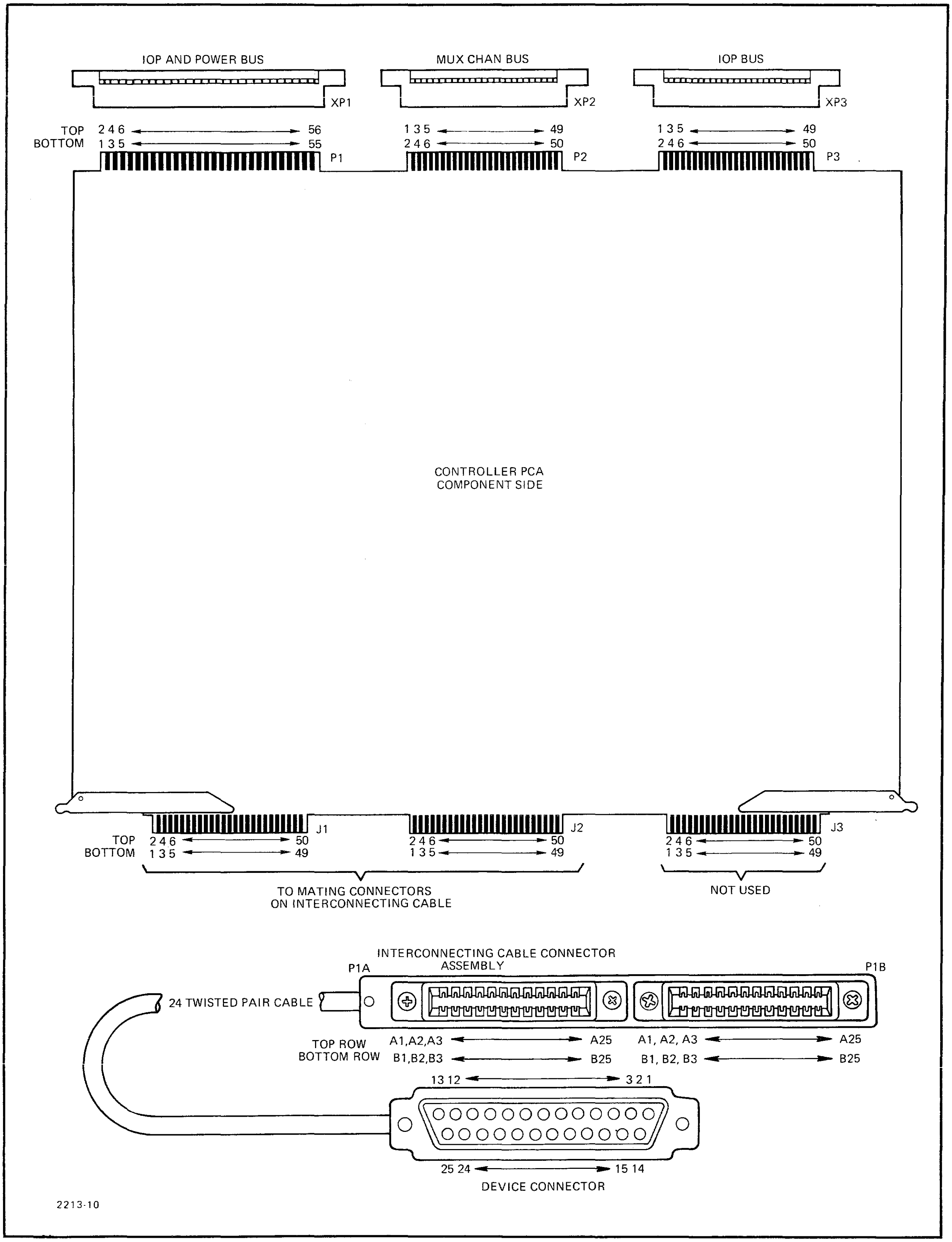


Figure 4-1. Connector Numbering

4-6. MAINTENANCE AIDS

A test connector assembly, part no. 30055-60009, is supplied with the controller PCA for diagnostic testing. This test connector assembly mates with connector P2 on the interconnecting cable assembly (see figures 1-1 and 4-1 for connection of the test connector to the interconnecting cable). The test connector assembly loops output signals back to the input signal lines and provides a means to test the operation of the controller PCA and the interconnecting cable assembly without the use of the external device. See figure 4-2 for internal wiring of the test connector.

Three stand-alone diagnostic programs have been designed to test the controller PCA operation through use of the test connector. Diagnostics D434A and D434B are for use when the controller is installed in a Series II or Series III computer system. D434A verifies the basic HP 30055A controller while D434B verifies Option 001 (asynchronous operation). The diagnostic manual with part number 30055-90008 explains how to run both of these programs. An additional diagnostic, D334A, verifies the controller in a pre-Series II Computer System. To run this diagnostic, use the manual with part number 30055-90004.

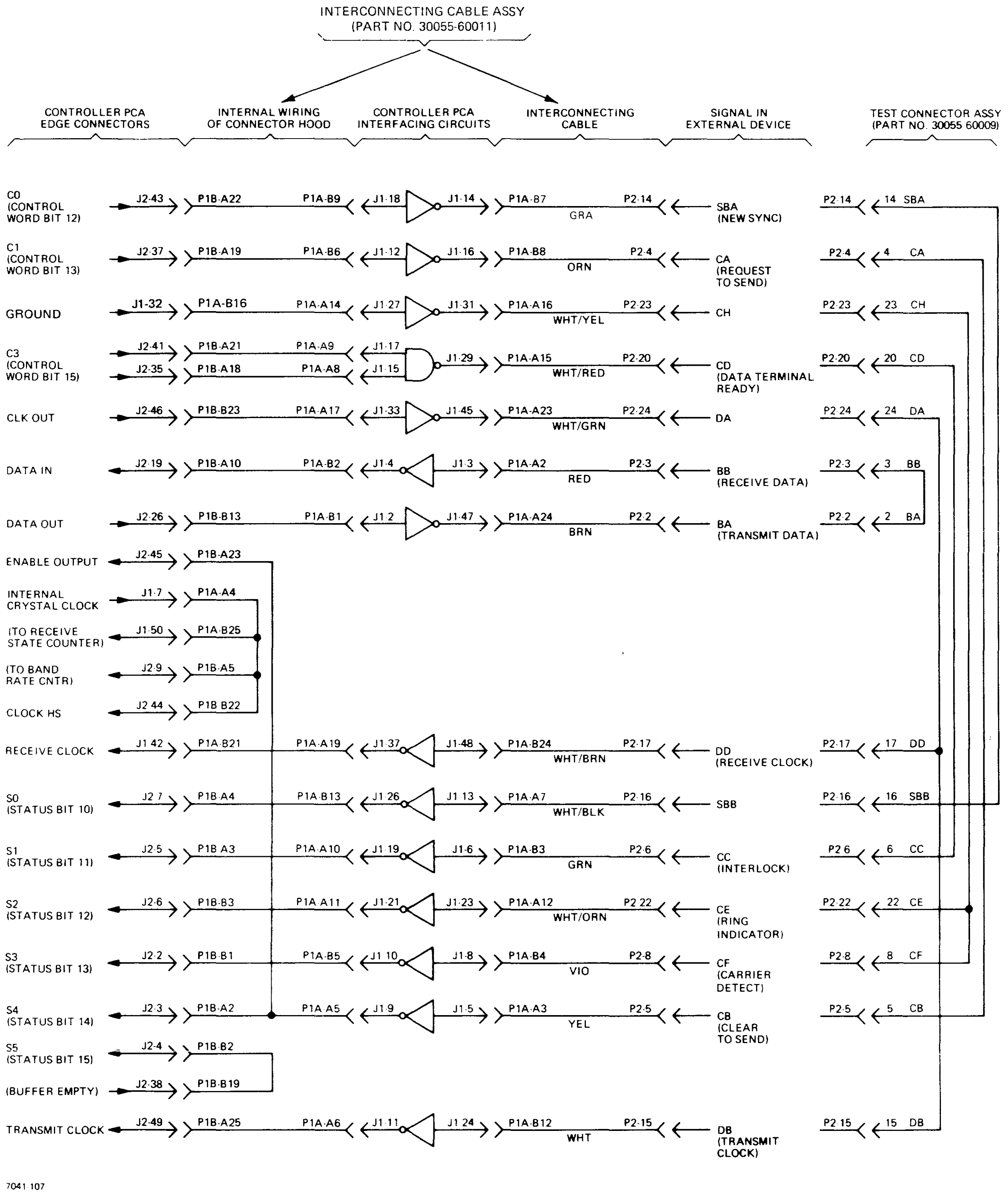

INTERCONNECTING CABLE ASSY
(PART NO. 30055-60011)
CONTROLLER PCA EDGE CONNECTORS
INTERNAL WIRING OF CONNECTOR HOOD
CONTROLLER PCA INTERFACING CIRCUITS
INTERCONNECTING CABLE
SIGNAL IN EXTERNAL DEVICE
TEST CONNECTOR ASSY (PART NO. 30055 60009)
C0 (CONTROL WORD BIT 12) J2·43 P1B·A22 P1A·B9 J1·18 J1·14 P1A·B7 GRA P2·14 SBA (NEW SYNC) P2·14 14 SBA
C1 (CONTROL WORD BIT 13) J2·37 P1B·A19 P1A·B6 J1·12 J1·16 P1A·B8 ORN P2·4 CA (REQUEST TO SEND) P2·4 4 CA
GROUND J1-32 P1A-B16 P1A·A14 J1·27 J1·31 P1A·A16 WHT/YEL P2·23 CH P2·23 23 CH
C3 (CONTROL WORD BIT 15) J2·41 P1B·A21 P1A·A9 J1·17 J2·35 P1B·A18 P1A·A8 J1·15 J1·29 P1A·A15 WHT/RED P2·20 CD (DATA TERMINAL READY) P2·20 20 CD
CLK OUT J2·46 P1B·B23 P1A·A17 J1·33 J1·45 P1A·A23 WHT/GRN P2·24 DA P2·24 24 DA
DATA IN J2·19 P1B·A10 P1A·B2 J1·4 J1·3 P1A·A2 RED P2·3 BB (RECEIVE DATA) P2·3 3 BB
DATA OUT J2·26 P1B·B13 P1A·B1 J1·2 J1·47 P1A·A24 BRN P2·2 BA (TRANSMIT DATA) P2·2 2 BA
ENABLE OUTPUT J2·45 P1B·A23
INTERNAL CRYSTAL CLOCK J1·7 P1A·A4
(TO RECEIVE STATE COUNTER) J1·50 P1A·B25
(TO BAND RATE CNTR) J2·9 P1B·A5
CLOCK HS J2·44 P1B·B22
RECEIVE CLOCK J1·42 P1A·B21 P1A·A19 J1·37 J1·48 P1A·B24 WHT/BRN P2·17 DD (RECEIVE CLOCK) P2·17 17 DD
S0 (STATUS BIT 10) J2·7 P1B·A4 P1A·B13 J1·26 J1·13 P1A·A7 WHT/BLK P2·16 SBB P2·16 16 SBB
S1 (STATUS BIT 11) J2·5 P1B·A3 P1A·A10 J1·19 J1·6 P1A·B3 GRN P2·6 CC (INTERLOCK) P2·6 6 CC
S2 (STATUS BIT 12) J2·6 P1B·B3 P1A·A11 J1·21 J1·23 P1A·A12 WHT/ORN P2·22 CE (RING INDICATOR) P2·22 22 CE
S3 (STATUS BIT 13) J2·2 P1B·B1 P1A·B5 J1·10 J1·8 P1A·B4 VIO P2·8 CF (CARRIER DETECT) P2·8 8 CF
S4 (STATUS BIT 14) J2·3 P1B·A2 P1A·A5 J1·9 J1·5 P1A·A3 YEL P2·5 CB (CLEAR TO SEND) P2·5 5 CB
S5 (STATUS BIT 15) J2·4 P1B·B2
(BUFFER EMPTY) J2·38 P1B·B19
TRANSMIT CLOCK J2·49 P1B·A25 P1A·A6 J1·11 J1·24 P1A·B12 WHT P2·15 DB (TRANSMIT CLOCK) P2·15 15 DB
7041·107

Figure 4-2. HP 30055-60011 Cable and Test Connector Wiring Diagram

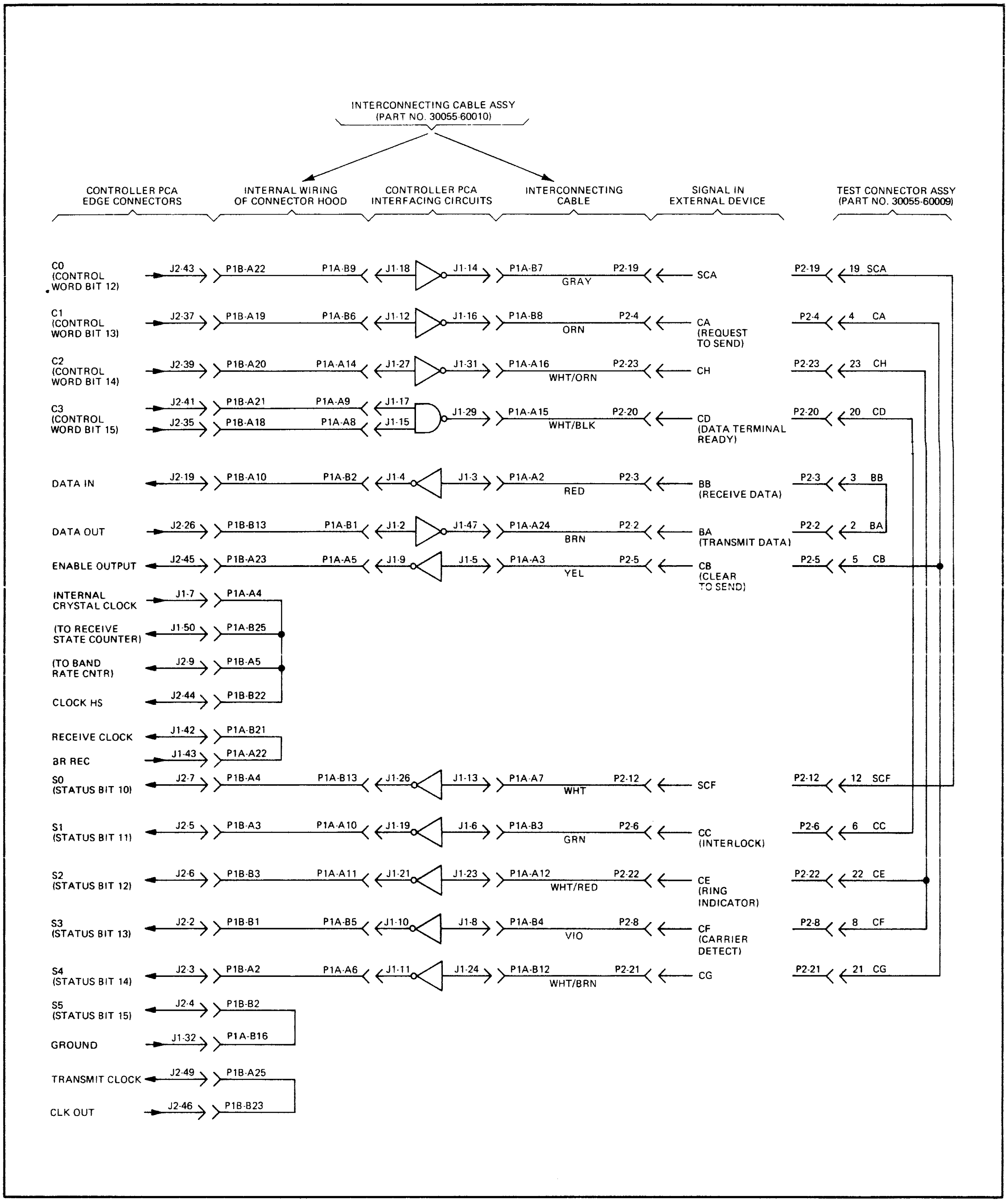

INTERCONNECTING CABLE ASSY
(PART NO. 30055-60010)
CONTROLLER PCA EDGE CONNECTORS
INTERNAL WIRING OF CONNECTOR HOOD
CONTROLLER PCA INTERFACING CIRCUITS
INTERCONNECTING CABLE
SIGNAL IN EXTERNAL DEVICE
TEST CONNECTOR ASSY (PART NO. 30055-60009)
C0 (CONTROL WORD BIT 12)
J2-43 P1B-A22 P1A-B9 J1-18 J1-14 P1A-B7 P2-19 GRAY SCA P2-19 19 SCA
C1 (CONTROL WORD BIT 13)
J2-37 P1B-A19 P1A-B6 J1-12 J1-16 P1A-B8 P2-4 ORN CA (REQUEST TO SEND) P2-4 4 CA
C2 (CONTROL WORD BIT 14)
J2-39 P1B-A20 P1A-A14 J1-27 J1-31 P1A-A16 P2-23 WHT/ORN CH P2-23 23 CH
C3 (CONTROL WORD BIT 15)
J2-41 P1B-A21 P1A-A9 J1-17
J2-35 P1B-A18 P1A-A8 J1-15
J1-29 P1A-A15 P2-20 WHT/BLK CD (DATA TERMINAL READY) P2-20 20 CD
DATA IN
J2-19 P1B-A10 P1A-B2 J1-4 J1-3 P1A-A2 P2-3 RED BB (RECEIVE DATA) P2-3 3 BB
DATA OUT
J2-26 P1B-B13 P1A-B1 J1-2 J1-47 P1A-A24 P2-2 BRN BA (TRANSMIT DATA) P2-2 2 BA
ENABLE OUTPUT
J2-45 P1B-A23 P1A-A5 J1-9 J1-5 P1A-A3 P2-5 YEL CB (CLEAR TO SEND) P2-5 5 CB
INTERNAL CRYSTAL CLOCK
J1-7 P1A-A4
(TO RECEIVE STATE COUNTER)
J1-50 P1A-B25
(TO BAND RATE CNTR)
J2-9 P1B-A5
CLOCK HS
J2-44 P1B-B22
RECEIVE CLOCK
J1-42 P1A-B21
BR REC
J1-43 P1A-A22
S0 (STATUS BIT 10)
J2-7 P1B-A4 P1A-B13 J1-26 J1-13 P1A-A7 P2-12 WHT SCF P2-12 12 SCF
S1 (STATUS BIT 11)
J2-5 P1B-A3 P1A-A10 J1-19 J1-6 P1A-B3 P2-6 GRN CC (INTERLOCK) P2-6 6 CC
S2 (STATUS BIT 12)
J2-6 P1B-B3 P1A-A11 J1-21 J1-23 P1A-A12 P2-22 WHT/RED CE (RING INDICATOR) P2-22 22 CE
S3 (STATUS BIT 13)
J2-2 P1B-B1 P1A-B5 J1-10 J1-8 P1A-B4 P2-8 VIO CF (CARRIER DETECT) P2-8 8 CF
S4 (STATUS BIT 14)
J2-3 P1B-A2 P1A-A6 J1-11 J1-24 P1A-B12 P2-21 WHT/BRN CG P2-21 21 CG
S5 (STATUS BIT 15)
J2-4 P1B-B2
GROUND
J1-32 P1A-B16
TRANSMIT CLOCK
J2-49 P1B-A25
CLK OUT
J2-46 P1B-B23

Figure 4-3. HP 30055-60010 Cable and Test Connector Wiring Diagram

INSTALLATION

SECTION V

5-1. INTRODUCTION

This section contains information regarding the installation of the controller into an existing HP 3000 computer system.

5-2. PROCEDURE

(1) Refer to paragraph 1-6 and verify that all parts were received.

(2) Configure the controller PCA jumpers as specified on the subsystem configuration form. See figure 5-1 for jumper locations.

a. DRT Address Parity. This jumper is removed or installed so that the total number of jumpers installed in positions W1 and W4 through W11 is even.

b. Service Request Priority (W2). Jumper wire W2 selects one of 16 service request (i.e., SR) lines to the multiplexer channel. Each controller serviced by a particular multiplexer channel must be connected through a separate service request line. The controller SR connection should be of a lower priority than magnetic tape controller (i.e., higher SR line numerically).

c. Group Interrupt Mask (W3). Jumper wire W3 selects the group interrupt mask bit the controller responds to during a SMSK instruction. The jumper can be associated with any of the 16 bits of the transmitted mask word, to a permanent enable position, or a permanent disable position.

d. Device number (W4 through W11). The configuration of these eight jumper wires determines the device number and, therefore, the DRT address associated with the controller PCA. A logic "1" is represented by the absence of a jumper wire and, conversely, a logic "0" is represented by the installation of a jumper wire. Check with the system manager to determine the device number.

(3) Set the system DC power switch to the STANDBY position.

(4) Install the controller PCA into the PCA module associated with the multiplexer channel.

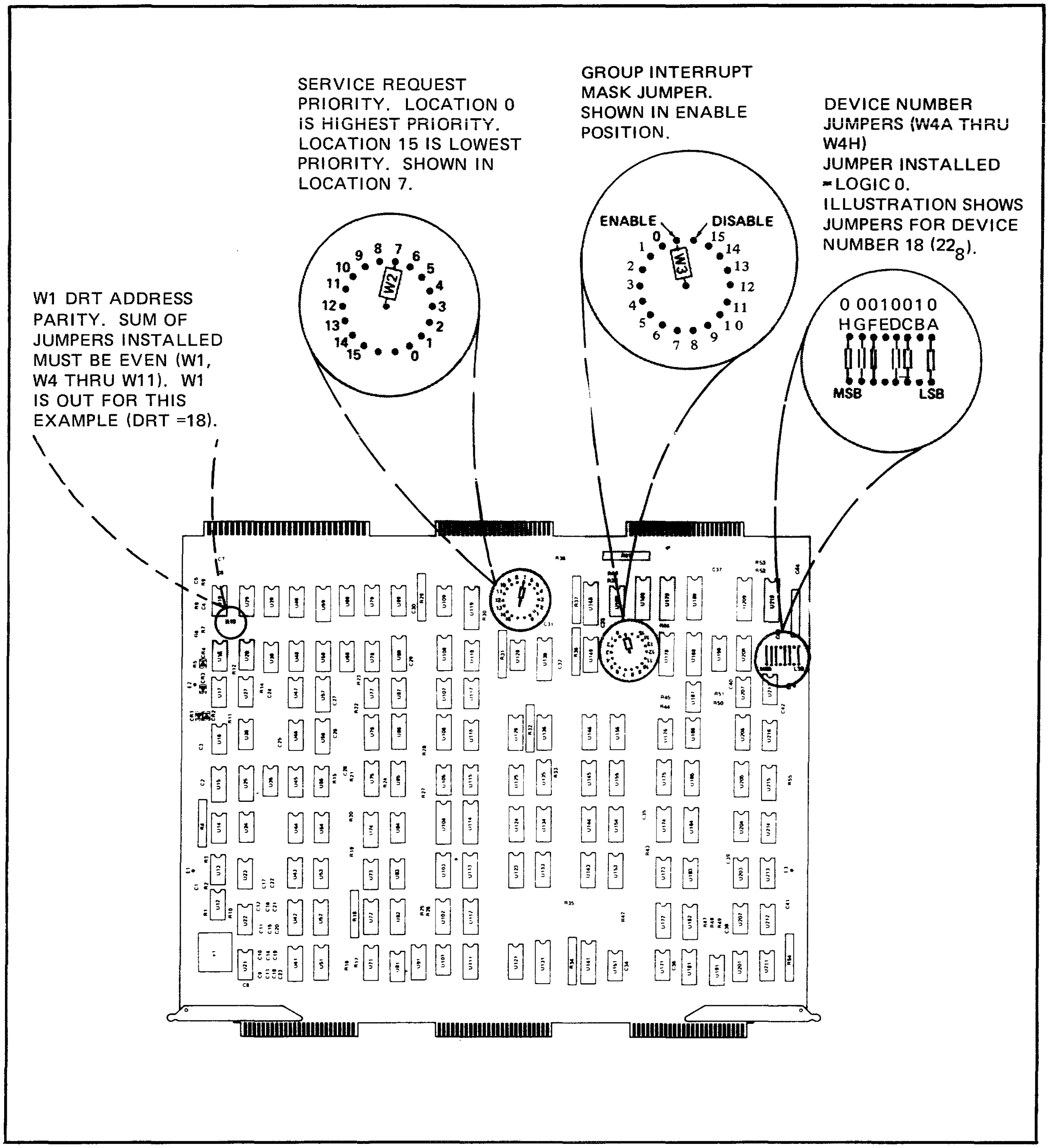
(5) Install the interconnecting cable between the controller PCA and the modem or asynchronous repeater.

(6) Install the interrupt poll (orange and white twisted pair). The synchronous single-line controller should come immediately after the asynchronous terminal controller in the interrupt poll series.

(7) Set the system DC power switch to the ON position.

(8) Run the HP 30055A Synchronous Single-Line Controller diagnostic to ensure proper operation of the controller.

(9) Return the system to the customer.



SERVICE REQUEST PRIORITY. LOCATION 0 IS HIGHEST PRIORITY. LOCATION 15 IS LOWEST PRIORITY. SHOWN IN LOCATION 7.

GROUP INTERRUPT MASK JUMPER. SHOWN IN ENABLE POSITION.

DEVICE NUMBER JUMPERS (W4A THRU W4H) JUMPER INSTALLED = LOGIC 0. ILLUSTRATION SHOWS JUMPERS FOR DEVICE NUMBER 18 (22_8).

W1 DRT ADDRESS PARITY. SUM OF JUMPERS INSTALLED MUST BE EVEN (W1, W4 THRU W11). W1 IS OUT FOR THIS EXAMPLE (DRT =18).

Figure 5-1. Jumper Locations

INDEX

READER COMMENT SHEET

HP 30055A
Synchronous Single-Line Controller
Installation and Service Manual

30055-90001 APR 1979

We welcome your evaluation of this manual. Your comments and suggestions help us improve our publications. Please use additional pages if necessary.

Is this manual technically accurate?	**Yes** □	**No** □	(If no, explain under Comments, below.)
Are the concepts and wording easy to understand?	**Yes** □	**No** □	(If no, explain under Comments, below.)
Is the format of this manual convenient in size, arrangement, and readability?	**Yes** □	**No** □	(If no, explain or suggest improvements under Comments, below.)

Comments:

FROM:

Name ____________________

Company ____________________

Address ____________________

____

____

FOLD FOLD

FIRST CLASS
PERMIT NO. 102(
SANTA CLARA
CALIFORNIA

BUSINESS REPLY MAIL

No Postage Necessary if Mailed in the United States. Postage will be paid by

Data Communications Support Group
Hewlett-Packard Company
General Systems Division
19447 Pruneridge Avenue
Cupertino, California 95014

FOLD FOLD

Part No. 30055-90001
Printed in U.S.A. 12/77
DCOM. 030/040. 30055-90001
Update No. 2 Incorporated 10/79

HEWLETT hp PACKARD

Sales and service from 172 offices in 65 countries.
5303 Stevens Creek Blvd., Santa Clara, California 95050