ICS User's Guide

First Edition

by

Les Acton

This guide documents the software operation of the Prime Computer and its supporting systems and utilities as implemented at Master Disk Revision Level 20.1 (Rev. 20.1) and PRIME/SNA Rev. 1.1 — 20.0 or later.

It is most important that, from Rev. 20, ALL ICS2s (including async only) are at REV. G or later.

Prime Computer, Inc.
Prime Park
Natick, Massachusetts 01760

COPYRIGHT INFORMATION

CREDITS

Editor	Marilyn Kirschen
Project Support	Kathe Rhoades David Roberts
Graphic Support	Terry Bacharz Marlene Bober Marjorie Clark Michael Moyle
Production Support	Michelle Hoyt
Document Preparation	Margie Matteson

PRINTING HISTORY — ICS USER'S GUIDE

Edition	Date	Number	Software Release
First Edition	March 1986	DOC10094-1LA	20.1

This guide is an expanded version of the ICS2 User's Guide, DOC10094-1LA (Rev. 20.0), first printed in August 1985.

CUSTOMER SUPPORT CENTER

Prime provides the following toll-free numbers for customers in the United States needing service:

1-800-322-2838 (within Massachusetts)
1-800-343-2320 (within other states)
1-800-541-8888 (within Alaska)
1-800-651-1313 (within Hawaii)

HOW TO ORDER TECHNICAL DOCUMENTS

Follow the instructions below to obtain a catalog, price list, and information on placing orders.

United States Only	International
Call Prime Telemarketing, toll free, at 800-343-2533, Monday through Friday, 8:30 a.m. to 8:00 p.m. (EST)	Contact your local Prime subsidiary or distributor.

This warning applies to the 9950, the 2250, and to all other 50 Series processors manufactured after October 1, 1983:

WARNING

This equipment generates and uses radio frequency energy and if not installed and used properly, i.e., in strict accordance with the instructions manual, may cause harmful interference to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

This warning applies to all other processors described in this book:

WARNING

This equipment generates and uses radio frequency energy and if not installed and used properly, i.e., in strict accordance with the instructions manual, may cause harmful interference to radio communications. As temporarily permitted by regulation, it has not been tested for compliance with the limits for Class A computing devices pursuant to Subpart J of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference. Operation of this equipment in a residential area is likely to cause interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Contents

3 PRIME HOST SOFTWARE

4 ICS2/3 SYSTEM ARCHITECTURE

5 CONFIGURATION

About This Book

INTRODUCTION

Prime's Intelligent Communications Subsystems, Model 2 (ICS2), and Model 3 (ICS3), enable a Prime 50 Series™ (host) to manage synchronous and asynchronous devices. The ICS2/3's ability to support asynchronous and synchronous communication lines allows a variety of communication protocols to be provided.

The ICS3 provides higher performance, more memory space, and more flexibility in the allowed protocol combinations than the ICS2. The ICS3 is recommended for new installations.

The ICS User's Guide is intended for those who will be supporting the PRIMENET™/X.25 software, bisynchronous-framed RJE, SDLC, and asynchronous protocols on the ICS2/3. Users of this manual should have a comprehensive knowledge of data communication protocols.

This guide provides detailed information about the ICS2 and the ICS3. Limited information is provided on the earlier Intelligent Communications Sybsystem, Model 1 (ICS1). More information on the ICS1 is contained in the references listed under OTHER DOCUMENTATION at the end of this section.

SYNOPSIS OF THE ICS USER'S GUIDE

The ICS User's Guide contains seven chapters and three appendixes.

- Chapter 1: A detailed description of the hardware components of the ICS2/3.
- Chapter 2: A comprehensive study of the various software components resident in the ICS2/3.
- Chapter 3: A short description of the software resident in the Prime host to support the various communication protocols.
- Chapter 4: An overview of the ICS2/3 system architecture.
- Chapter 5: A detailed description of the various ICS2/3 configurations that are available.
- Chapter 6: The current expected performance of the ICS2/3 under normal operating conditions, including the present communication protocol restrictions.
- Chapter 7: Details of the diagnostic tools available.
- Appendix A: Information on the various types of cable used in ICS2/3 configurations, and details of some ICS2/3 edge connectors, together with their pin assignments.
- Appendix B: A listing of the possible error conditions that can occur during operational use, together with an explanation of their possible cause and how to remedy the fault.
- Appendix C: A comprehensive glossary of terms and abbreviations used in communications generally and the ICS2/3 in particular.

HOW TO USE THE ICS USER'S GUIDE

The following is a guide to where information is found for different categories of users:

- Category 1: The user requiring a general overview of the ICS2/3 and its relationship to other Prime products should read the introduction to Chapters 1 (HARDWARE COMPONENTS), 2 (ICS2/3 SOFTWARE), 4 (ICS2/3 SYSTEM ARCHITECTURE), and 5 (CONFIGURATION).
- Category 2: The present user, upgrading to ICS2/3 synchronous support, or the user who needs enough information to install, configure, and initialize the ICS2/3 quickly should read Chapter 5 (CONFIGURATION).

- Category 3: The user requiring information for fault diagnosis should read Chapter 7 (DIAGNOSTICS) and Appendix B (ERROR AND STATUS MESSAGES).

- Category 4: The user requiring detailed knowledge of both hardware and software should read, at least, Chapters 1, 2, 3 (PRIME HOST SOFTWARE), and 4.

OTHER DOCUMENTATION

Other documentation that may be useful in understanding the ICS2/3 and its function in relation to other Prime products includes

- DOC10044-1LA ICS2 User's Guide, Revision 20.0
- DOC3710-193L PRIMENET Guide, Revision 19.3
- UPD3710-31A PRIMENET Guide, Revision 19.4
- UPD3710-32A PRIMENET Guide, Revision 20.0
- DOC5037-3LA System Administrator's Guide, Revision 20.0
- DOC7532-2LA Network Planning and Administration Guide, Revision 19.4
- DOC6053-3LA Remote Job Entry Phase II Guide, Revision 20.0
- DOC8908-2LA PRIME/SNA Administrator's Guide, Revision 19.4.5
- DOC8909-2LA PRIME/SNA Operator's Guide, Revision 19.4.5
- DOC8910-1LA PRIME/SNA Interactive Terminal User's Guide, Revision 19.4.1

and later editions or updates of all the above manuals.

1
Hardware Components

INTRODUCTION

The Intelligent Communications Subsystems, Model 2 (ICS2), and Model 3 (ICS3), are 16-bit Zilog 8001 (Z8001) microprocessor-based controllers, capable of supporting synchronous and asynchronous serial communication lines concurrently.

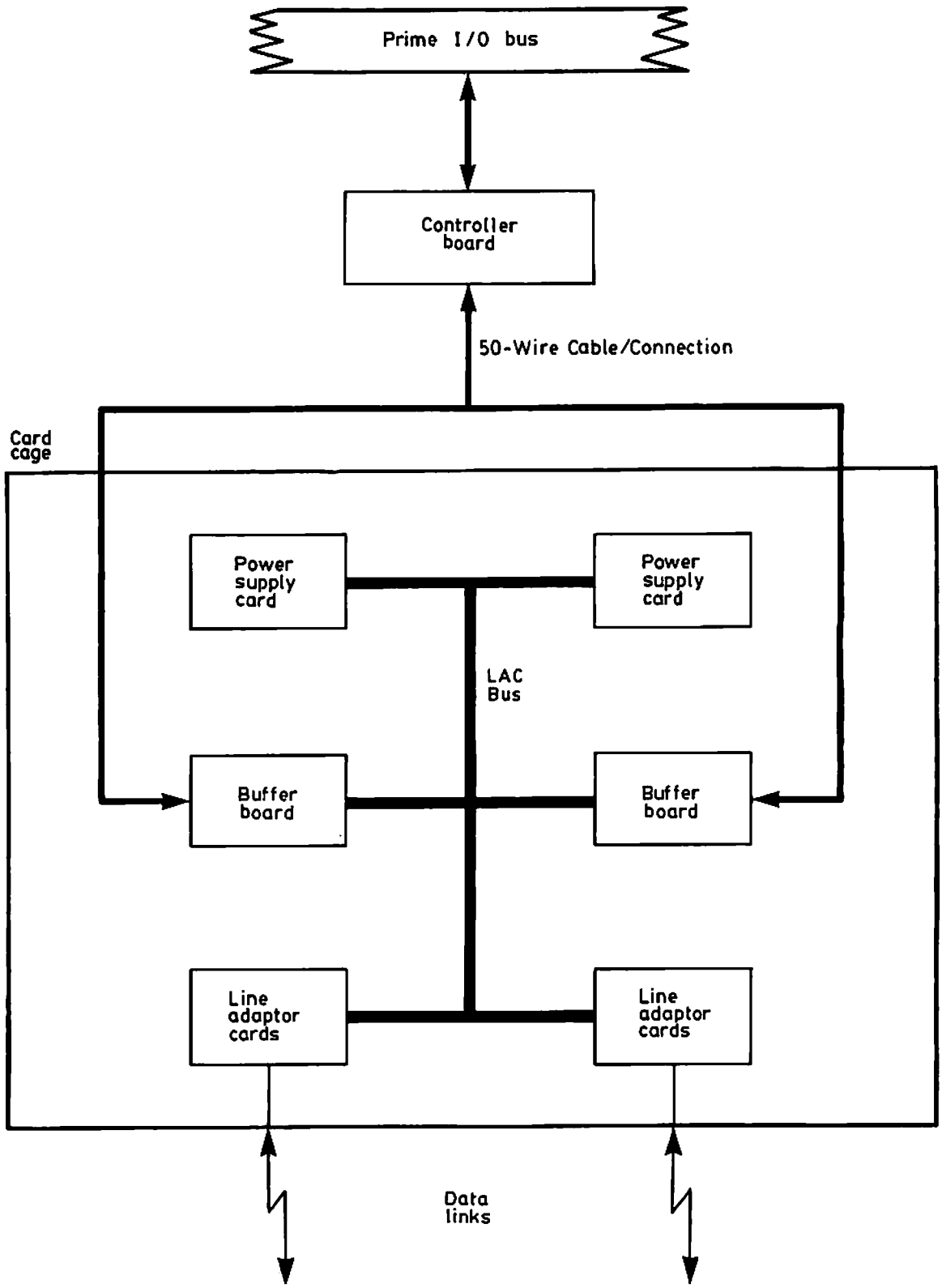
These lines can be configured to support a variety of protocols and electrical interfaces for communication with various terminal devices and controllers, other Prime computers, and the computers of other manufacturers. The ICS2/3 is supported by all current Prime 50 Series processors.

The hardware components are as follows:

- A standard Prime-size controller board
- An eight-slot Line Adapter Card (LAC) card cage for small Prime systems using the ICS3 controller board, and designed to fit inside the CPU cabinet
- A sixteen-slot LAC card cage for medium to large Prime systems using either the ICS2 or ICS3 controller board, and designed to fit in a peripheral cabinet
- A smaller sixteen-slot LAC card cage for medium to large Prime systems using the ICS3 controller board, and designed to fit inside some CPU cabinets or a peripheral cabinet

- A 50-wire cable to connect the controller board to the buffer board in the LAC card cage, either directly or via the Prime bulkhead
- Cables to connect ICS2 LACs to the Prime peripheral bulkhead, and then, to users' terminals, printers, and modems
- Cables to connect ICS3 LACs (using their integral bulkheading) directly to users' terminals, printers, and modems

An overall block diagram of the ICS2/3 is shown in Figure 1-1.



ICS2/3 Overall Block Diagram
Figure 1-1

CONTROLLER BOARD

General Description

The controller board can be logically divided into two functional areas:

- The Z8001 microprocessor, Z8001 support logic, memory storage, microprocessor-bus and associated input/output devices
- The Inter-Bus Controller (IBC)

Memory storage for microprocessor programs and data buffering is as follows. The ICS2 has a 2K x 16-bit word PROM, and up to 256K bytes (128K x 16-bit words) of dynamic RAM.

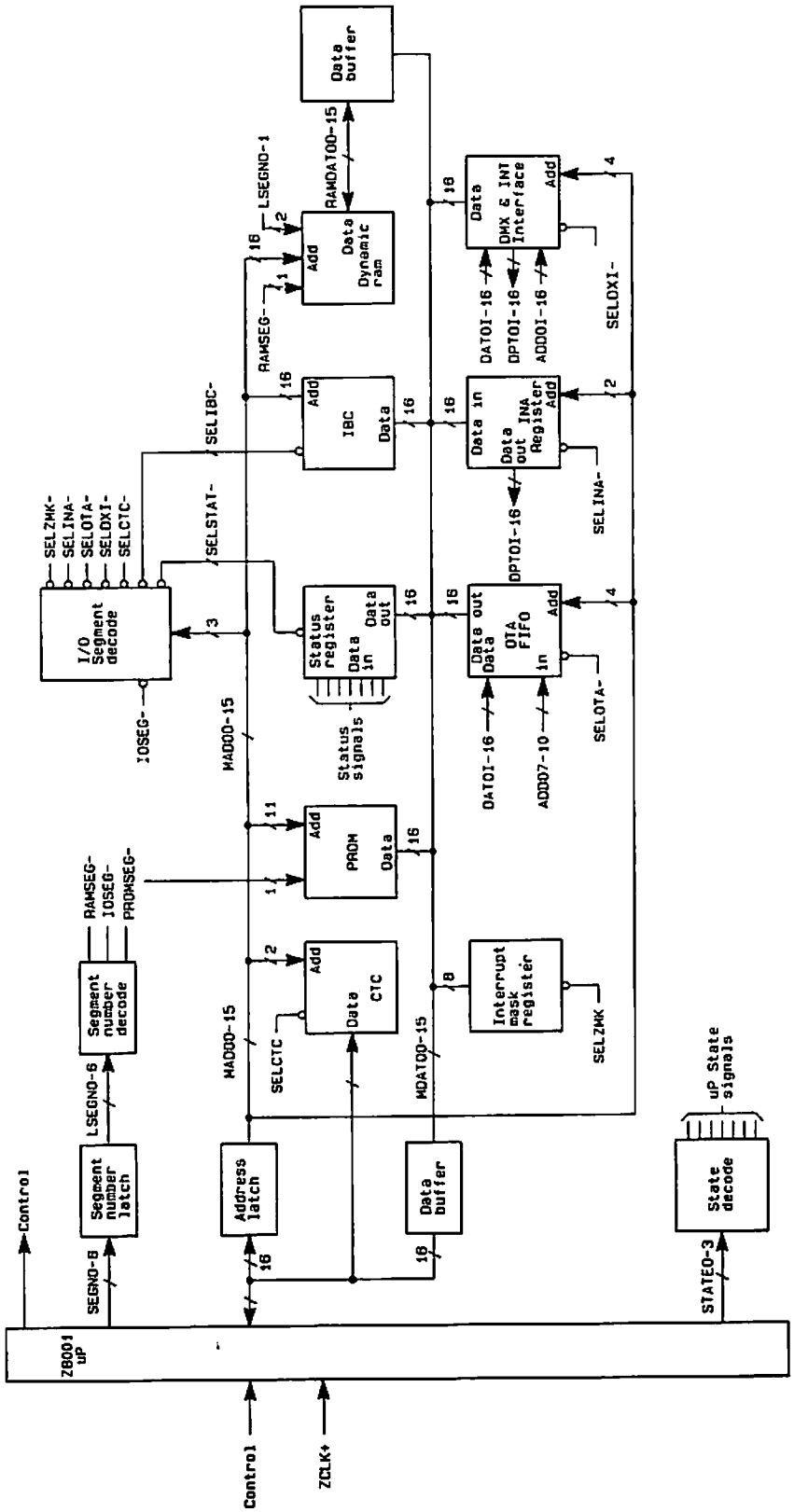
The ICS3 has a 8K x 16-bit word EPROM, and up to 1024K bytes (512K x 16-bit words) of dynamic RAM. The microprocessor communicates with its memory and input/output (I/O) devices over the 16-bit microprocessor bus (uBus).

The I/O devices residing on the uBus include a general purpose Counter-Timer Circuit (CTC), the standard Prime I/O interface logic components, the IBC, and for the ICS3 only, a Serial Communications Controller (SCC).

The IBC is a microprogrammed device whose primary purpose is to move data between three buses:

- The uBus
- The LAC bus
- The Prime I/O bus

Figure 1-2 is a block diagram showing the interconnection of the various components of the controller board.



Controller Board Logic Block Diagram
Figure 1-2

Microprocessor and Support Logic

The 16-bit Zilog Z8001 microprocessor is the main control element of the ICS2/3. Support logic for the Z8001 microprocessor consists of the following:

- Clock driver circuit
- Reset control logic
- State and address decode logic
- Bus buffers
- Microprocessor status register

Clock Driver Circuit: This circuit is necessary to ensure that the stringent timing and level specifications for the Z8001 clock input are met. Clock speeds are 8Mhz for the ICS3 and 4Mhz for the ICS2.

Reset Control Logic: The reset control logic has two functions: it synchronizes the asynchronous reset inputs to the microprocessor clock, and it also ensures the proper resetting of the microprocessor.

State and Address Decode Logic: The state decode logic decodes the four state lines (STATE0-3) from the microprocessor into one of 16 possible microprocessor state output signals. These output states indicate what cycle type is currently being executed. The address decode logic comprises the address latch, segment number latch, and segment number decoder. The output from the segment number latch (LSEGN0-6) is decoded and used in conjunction with the output from the address latch (MAD00-15) to select memory and peripheral devices.

Bus Buffers: These are a necessary requirement due to the limited drive current of the Z8001 microprocessor.

Microprocessor Status Register: The microprocessor status register is a 16-bit read-only register residing on the uBus. The data input consists of a number of status signals, routed from various areas of the controller board logic; these signals provide essential state information to the microprocessor.

Microprocessor Bus (uBus)

The uBus is comprised of 16 bi-directional lines designated ZAD00-15, which connect the microprocessor to its memory and other associated peripherals. Address and data information are multiplexed on the uBus. In addition to the 16 multiplexed address lines, the uBus includes seven address lines for the segment number (SEGN0-6). The segment number address lines increase the byte addressable memory from 64K bytes to a possible 8M bytes.

ICS3 Controller Board LED Lights

The ICS3 controller board is fitted with four LED indicator lights. These will flash or remain on when the following events occur:

- LED 1 (Flash) Warning — LAC card cage power failure
- LED 2 (Flash) Warning — Single-bit memory parity error detected
- LED 3 (On) Fatal — WCS parity error detected
- LED 4 (Flash) RUN light — when VERIFY is completed and the operating system is functioning correctly

All LEDs flash when multiple bit errors are detected.

Memory

Microprocessor PROM: The ICS2 has 4K bytes of PROM, and the ICS3 has 16K bytes of EPROM, both containing

- Initialization routines
- Bootstrap routines
- Self verification routines

Microprocessor RAM: On the ICS2, the microprocessor RAM consists of up to 256K bytes (128 x 16-bit words) of dynamic memory, with even parity checking on each byte. On the ICS3, the RAM consists of up to 1024K bytes (512 x 16-bit words) of dynamic memory with Error Checking and Correction (ECC) logic, capable of detecting multiple-bit errors and correcting single-bit errors. A minimum of 256K of memory is required to run any of the synchronous support options on the ICS2/3.

Input/Output Devices

Counter-timer Circuit: Three independent sections comprise the counter-timer circuit. All sections are implemented using a Zilog Z80A-CTC chip and the microprocessor clock ZCLK+. The sections function as follows:

- Section 0 is an independently programmable timer used by the microprocessor for general timing purposes. It generates non-vectored interrupts to the microprocessor.
- Section 1 is similar to Section 0, but with the added capability of using timeouts on CTC section 0 to count down its time-constant register. This enables cascading to create longer timeout intervals.
- Section 2 is a uniquely configured counter or timer section that, on timeout, generates segment traps to the microprocessor. Section 2 has the ability to use either the clock ZCLK+ or stack references to count down its time-constant register. The configuration of Section 2 enables hardware single-stepping to be implemented. Single stepping is used in diagnostic routines.

Standard I/O Interface Logic: The standard Prime I/O interface is composed of three logic units:

- The OTA FIFO is a 20-bit x 16 word, first-in-first-out memory used to store and buffer OTA instructions received by the ICS2 from the Prime. Because the rate at which the host CPU issues OTA instructions is potentially faster than the microprocessor handling rate, some form of queuing is required.
- The INA register is a 16-bit register written-to by the microprocessor and read by any INA instruction sent from the host CPU (Prime). A non-vectored interrupt informs the microprocessor that data has been read by the host CPU.
- The direct-memory-access interface logic (which incorporates the interrupt interface logic) enables the microprocessor to set up and request direct memory cycles on the Prime I/O bus. The microprocessor initiates DMX cycles by configuring the address, mode, and data registers as appropriate, and then setting up the DMX request logic.

Interrupt interface logic is the mechanism for setting up and requesting Prime vectored interrupts.

Serial Communication Controller (SCC): A dual-channel Z8530-SCC is used on the ICS3 for the main purpose of supporting RASBUS. The primary task of the SCC is to perform serial-to-parallel conversion for the two communication channels. The two channels are used as follows:

- Channel A is used as a general purpose link for direct communication to the Z8001 using an RS232 interface provided on the ICS3.
- Channel B is dedicated to the sole purpose of supporting RASBUS.

Inter-Bus Controller (IBC): The IBC is a specialized, high-speed microcode programmed device whose primary task is to control the flow of data between the three buses under its influence:

- The 16-bit microprocessor bus (uBus)
- The 8-bit LAC bus
- The Prime I/O bus

The IBC also allows a reduction in the logic that would otherwise be required on the LACs.

The interconnection of the IBC is shown in Figure 1-3 and includes the listed logic components, which are described in the following pages.

- Sequencer
- Writeable Control Store and Pipeline Latch
- Independent Action Code (IAC) Decoder
- IBC Bus Control
- ALU (Arithmetic Logic Unit)
- Line File
- RAMX
- Bus Manager
- LAC Bus Interface Logic
- Microprocessor Bus Interface Logic
- DMX Interface Logic
- IBC Command Register

Sequencer: The 12-bit sequencer provides an address into the writeable control store. A 4-bit microcode field on the input of the sequencer selects one of 16 possible sequencer instructions.

The execution of most of the sequencer instructions depends upon the output from the condition code multiplexer logic. This output is a 4-bit microcode field that allows for the selection of one of 16 possible condition codes. These condition codes are states that exist in the IBC and can be tested by the microcode.

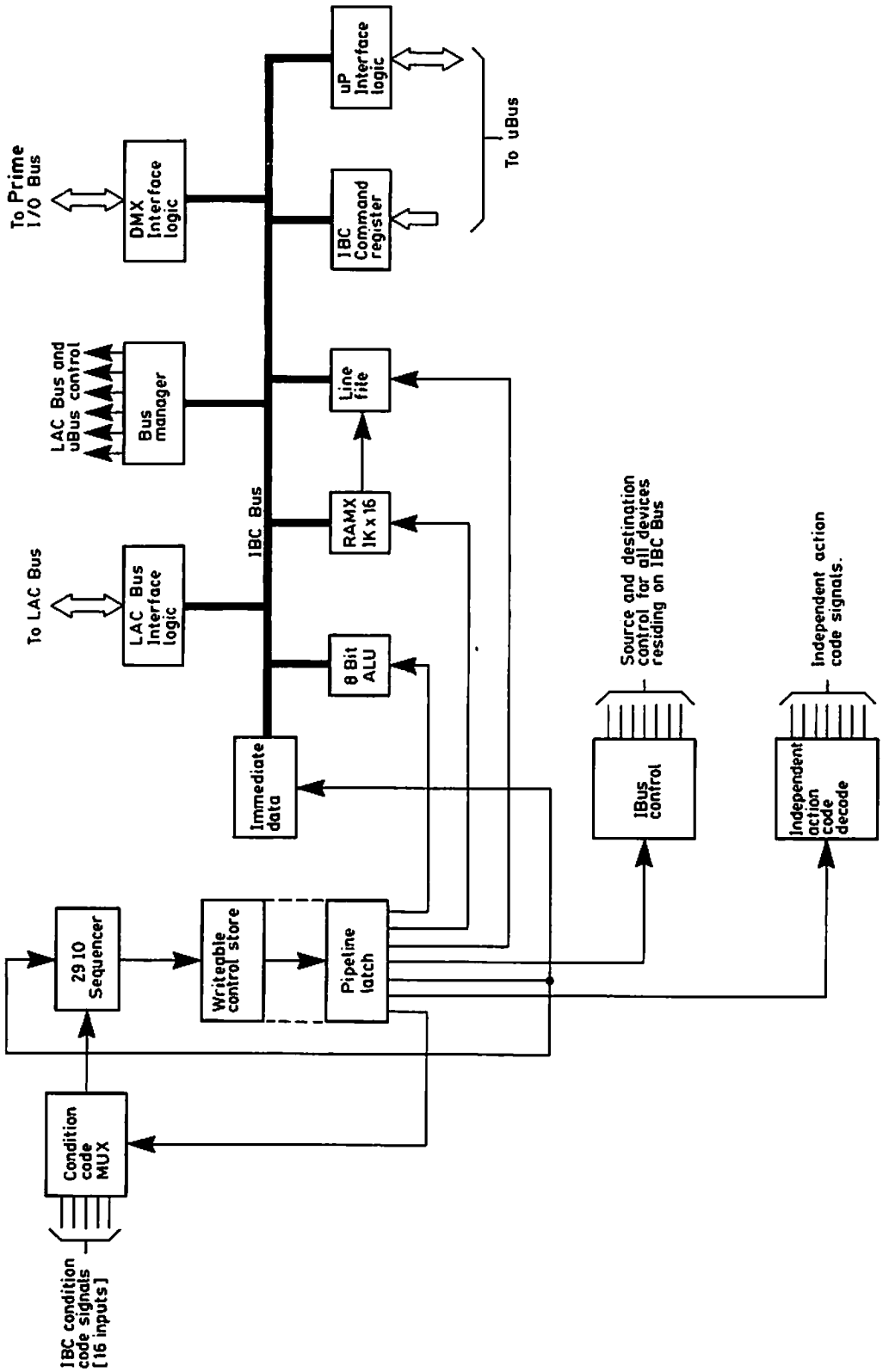
The sequencer can be stopped, started, and single-stepped by the microprocessor; this capability provides a powerful debugging feature for the hardware and software components of the ICS2/3. This facility is available only for Prime internal development personnel, and consequently, is not visible to users.

Writeable Control Store and Pipeline Latch: In the ICS2, the Writeable Control Store (WCS) is a 56-bit wide by 1K deep memory that stores the control code (microcode) for the IBC. In the ICS3, the WCS is 64 bits wide by 4K deep, and contains parity logic. The 56/64-bit output word is latched from the WCS into the pipeline latch on every IBC cycle; from there, it is divided into microcode fields that are used for control purposes throughout the IBC. There are seven microcode fields output from the pipeline:

- Sequencer operation field (Sequencer)
- ALU control field (ALU)
- IBC source and destination control field (IBC Bus Control)
- Line file index field (Line File)
- Independent Action Code (IAC) select field (Independent Action Code Decoder)
- Condition code select field (Condition Code Multiplexer)
- Next address/immediate data field (IBC Bus)

Independent Action Code Decoder: This logic decodes the microcode field from the WCS into IAC signals. These code signals are used to activate control signals in the IBC.

IBC Bus Control: The 16-bit IBC bus interconnects various data sources and destinations throughout the IBC. The IBC bus control specifies, in microcode, the source and destination devices for the movement of any data on the IBC bus.



IBC Logic Block Diagram
Figure 1-3

Arithmetic Logic Unit: The IBC ALU is a general purpose 8-bit ALU residing on the IBC bus. The ALU is capable of performing both arithmetic and logical functions under the control of the microcode input from the pipeline latch.

Line File: The line file is a 2K (ICS2), or 4K (ICS3), by 16-bit RAM residing on the IBC bus. Its function is to store single line DMA pointers and state information. Each line on the LAC bus is allocated a maximum of four blocks of storage, within the line file, to accommodate a maximum of four possible DMA or data movement channels associated with each line.

RAMX: The RAMX is a 1K (ICS2), or 4K (ICS3), x 16-bit high-speed RAM residing on the IBC bus. On the ICS2, the RAM is divided into four 256 x 16-bit sections, named RAM0, through RAM3. On the ICS3, the RAM is divided into sixteen 256 x 16-bit sections, named RAM0 through RAM15.

RAM0 is used to store mapping information that enables physical LAC bus addresses to be mapped to logical addresses used by the line file. RAM0 also supplies the two control bits for determining the line-file block size in the line-file address multiplexing logic.

RAM1, 2 and 3 are used for protocol-specific storage.

RAM4 through RAM15 are available for general purpose storage.

Bus Manager: The bus manager is a small-state machine that incorporates configured PROMs and registers, which generate timing signals for the LAC bus and the uBus. The bus manager is initiated under microcode control, and supports three LAC bus cycle types (read, wait, interrupt acknowledge) and two uBus cycle types (read, write). The ICS3 bus manager consists of Programmable Logic Arrays (PLAs or PALs) with registers, which replace the PROMS and registers of the ICS2.

LAC Bus Interface Logic: The LAC bus interface logic consists of address and data registers, which are used in conjunction with the bus manager, to execute LAC bus read, write, and interrupt acknowledge cycles. The interface logic incorporates line drivers and receivers to interface via the 26-foot controller board to buffer board cable.

Microprocessor Bus Interface Logic: The microprocessor bus interface logic consists of four 16-bit registers, an 8-bit register, a tri-state driver for the four microprocessor state lines, and the microprocessor vectored-interrupt request flip-flop. The logic has two functions:

- To transfer data to and from the Z8001
- To initiate vectored interrupts to the Z8001

DMX Interface Logic: The DMX interface logic comprises address, data, and mode registers, which are configured and read by the IBC when executing I/O bus DMX cycles. The interface logic also includes an IBC DMX request bid flip-flop, an IBC DMX request flip-flop, and an IBC DMX end-of-range flip-flop. This logic is capable of executing the following transfers on the Prime I/O bus:

- DMA — Direct Memory Access
- DMC — Direct Memory Control
- DMT — Direct Memory Transfers
- DMQ — Direct Memory Queue

The DMX interface logic supports an 18-bit DMX address.

IBC Command Register: The IBC command register is a 16-bit register into which is written 16 bits of command data sent from the Z8001 to the IBC. This register is an IBC read-only register.

LAC CARD CAGE ASSEMBLY

The LAC card cage is available in three sizes to accommodate two LAC sizes.

8-LAC ICS3 Card Cage

This card cage houses the smaller (6.5 by 5 inches) ICS3 LACs used on small Prime systems requiring up to eight LACs. The card cage contains its own backplane and buffer board, but is not fitted with a power supply. A separate cable provides power from the CPU power supply. (Refer to Figure 5-4, 8-LAC ICS3 Card Cage, in Chapter 5, CONFIGURATION.)

16-LAC ICS3 Card Cage

This card cage houses (in two groups of eight) up to sixteen of the smaller (6.5 by 5 inches) ICS3 LACs. Each group of eight LACs has its own backplane and buffer board. One power supply, fitted between the two groups of eight LACs, provides power for all sixteen LACs. (Refer to Figure 5-5, 16-LAC ICS3 Card Cage, in Chapter 5, CONFIGURATION.)

ICS2 Card Cage

This card cage houses the larger (10 by 8 inches) ICS2 LACs. It can contain up to sixteen LACs, and a backplane, buffer board, and power supply for each group of eight LACs. (Refer to Figure 5-8 and Figure 5-10 in Chapter 5, CONFIGURATION.)

Buffer Board

The buffer board comes in two sizes; 10 by 8 inches for the ICS2 cage, and 6.5 by 5 inches for the ICS3 cages. The ICS2 buffer board plugs into the slot on the right of each power supply board in the ICS2 LAC cage. On ICS3 LAC cages, the ICS3 buffer board mounts on the back of, and parallel to, the backplane. The primary function of the buffer board is to provide an electrical interface between the 50-wire cable from the controller board and the LAC bus. The LAC bus resides on the backplane of the LAC card cage assembly.

A green LED, located on the front of the ICS2 buffer board, indicates that power is available on the LAC bus backplane.

Additional functions performed by the buffer board are accomplished by the following logic circuits:

- Cable interface
- LAC bus interface
- Parity logic
- Interrupt acknowledge logic

Cable Interface: Interfacing of the 50-wire cable, which connects the controller board to the buffer board, is achieved by using open collector bus transceivers. The cable signal wires are terminated on both ends of the cable.

LAC Bus Interface: The buffer board interface to the backplane is relatively simple, due to the short length of the LAC bus backplane. The majority of the LAC bus control signals are driven directly from the receive side of the bus transceivers.

Parity Logic: The buffer board parity generation/checking logic is used to generate and check parity on the data and address information transmitted over the controller board to buffer board cable. Furthermore, the parity logic checks the cable transceivers and the LAC bus transceivers because no parity checking is carried out on the LAC bus backplane or within the LACs.

Under normal operation, LAC bus read cycles to a nonexistent slot or backplane do not generate a parity error.

Interrupt Acknowledge Logic: The buffer board interrupt acknowledge logic is used to prioritize the LAC bus interrupt requests and to select the highest priority LAC during LAC bus interrupt acknowledge cycles. The buffer board receives separate interrupt request lines from each LAC.

Because two buffer boards and, in turn, two LAC backplanes can be connected to one controller board, additional logic has been incorporated to differentiate the buffer board that should receive the data cycle. A signal, generated in the buffer board, is used to make the determination. As a result, during interrupt acknowledge cycles, the buffer board knows whether or not the cycle is for a slot in its backplane.

In addition to prioritizing LAC bus interrupt requests, the buffer board also modifies the interrupt vector obtained from the LACs during interrupt acknowledge cycles.

Asynchronous RS232 LAC

There are two basic types and sizes of asynchronous LACs. The ICS2 LAC type (10 by 8 inches) uses Zilog's SIO (Serial Input/Output) controller, while the ICS3 LAC type (6.5 by 5 inches) uses Zilog's SCC (Serial Communications Controller). The ICS3 controller board will support both SIO-fitted and SCC-fitted asynchronous LACs. However the ICS2 controller board only supports SIO-fitted asynchronous LACs.

The asynchronous RS232 LAC sends and receives serial streams of data to and from the Data Terminal Equipment (DTE). In asynchronous mode, characters are delimited by start and stop bits that help to synchronize character bit timing.

The following logic circuits comprise the asynchronous RS232 LAC:

- Standard LAC Bus Interface Logic
- Baud Rate Generator
- Timing Generation Logic: SIO (Zilog Serial I/O controller) in the ICS2, and SCC (Zilog Serial Communications Controller) in the ICS3
- SIO/SCCs and Support Logic
- Loopback Control Logic
- Electronic Industries Association (EIA) Interface Circuit

Standard LAC Bus Interface Logic: The asynchronous RS232 LAC uses standard LAC bus interface logic. Because the LAC bus has a multiplexed address and data path, the address is latched on each LAC bus cycle and the bidirectional data is buffered by using tri-state devices.

Each LAC is assigned an 8-bit ID, which is addressed during a read cycle.

Baud Rate Generators: Two baud-rate generators provide the four baud-rate clocks for the communication lines on the asynchronous RS232 LAC. A four-bit code sent by the IBC configures each clock to the desired speed.

The four baud-rate clock outputs are connected directly to the four serial input/output communication lines on the LAC. The speeds for these lines are described in the System Administrator's Guide, Rev. 20, under the ICS JUMPER directive. (The ASYNC JUMPER directive replaces the ICS JUMPER directive; both are currently supported.)

Note

The transmit and receive baud-rate clocks for each communications line are currently constrained to run at the same speed.

SCC Timing Generation Logic: The ICS3 asynchronous LAC uses the Zilog Serial Communications Controller (SCC) chip instead of the Zilog Serial Input/Output (SIO) chip used in the ICS2 asynchronous LAC. This chip performs serial-to-parallel and parallel-to-serial conversion. It enables the SCC to read and write data and interrupt acknowledge cycles.

SIO Timing Generation Logic: The SIO timing generation logic forms the control signals that the SIO requires for read, write, and interrupt acknowledge cycles.

SIOs and Support Logic: Each asynchronous RS232 LAC is fitted with two dual-channel SIO devices. These provide the primary function of serial-to-parallel and parallel-to-serial conversion for the four communication lines. Additional SIO functions include protocol handling, data set status, data set control line handling, and interrupt vector generation.

Note

The SIO devices on the asynchronous RS232 LAC are the only generators of LAC bus interrupt requests.

The SIO support logic basically consists of address decode logic, which is used to select the appropriate SIO (Zilog).

Loopback Control Logic: Each of the asynchronous RS232 LAC's four communication lines has a data loopback capability. Primarily, the data loopback capability loops the transmit data back through receive data on a particular communication line. This loopback gives the ICS2/3 the ability to verify the correct operation of a LAC by transmitting data to the SIO/SCC chip and verifying that the data returns intact via the receiving line. This is a software-enabled hardware loopback.

EIA Interface Circuit: The EIA (Electronics Industries Association) interface circuit provides signal level conversion between TTL signals and the EIA signals. This signal level conversion is required for interfacing to the RS232/V24 configured equipment. Each communication line on the asynchronous RS232 LAC transmits and receives three EIA signals.

Transmitted EIA signals are

- Transmit data
- Request to send (RTS is held high)
- Data terminal ready (controlled by the PRIMOS® operating system)

Received EIA signals are

- Receive data
- Data carrier detect
- Clear to send

Synchronous LAC

There are two sizes of synchronous LACs: 10 by 8 inches, for the ICS2 LAC card cage, and 6.5 by 5 inches, for the ICS3 LAC card cages. The ICS3 controller board supports both sizes, but the ICS2 controller board supports only the ICS2 size.

The synchronous and asynchronous LACs are very similar. Their basic differences are

- The synchronous LAC supports two lines compared to the four-line support of the asynchronous LAC
- The synchronous LAC supports four dataset control signals (modem speed select, DTR, RTS, spare) and four dataset status signals (CTS, DCD, DSR, RI) for each line. The asynchronous LAC supports only two dataset control signals and two dataset status signals per line
- The synchronous LAC has no programmable loopback capability
- The synchronous LAC is available in two versions: V.24 and V.35. The V.24 LAC is intended for applications that support line speeds to a maximum of 19,200 bps, whereas the V.35 LAC is required for the higher speed operations that range from 19.2 to 64 Kbps. The V.24 and V.35 LACs require different cabling to the Prime bulkhead and from the Prime bulkhead to the customer equipment. (See Appendix A.) The ICS3 LACs have integral bulkheading.
- The synchronous LAC uses the Zilog SCC (Serial Communication Controller), while the asynchronous LAC uses either the SCC (for use with ICS3 only), or the SIO Serial Input/Output for use with ICS2 or ICS3 controller boards.

Power Supply Card

The power supply, or power supply card, provides the buffer board(s) and the LACs with regulated voltages. The power supplies can be internally configured for the various voltage/frequency requirements of different countries.

LAC Bus

The LAC bus consists of a synchronous, tri-state, 8-bit bus incorporating a multiplexed address and data path. In addition, the LAC bus includes

- Four slot select bits, which interact with the hardwired slot numbers on the LAC bus backplane and are used for LAC and LAC bus backplane selection
- Two line select bits, which are used for selection of LAC communication lines 1 through 4
- Various control signals

Three types of LAC bus cycles are supported: read, write, and vectored interrupt acknowledge.

EXTERNAL INTERFACES AND CABLING

The ICS2/3 communicates with external equipment via the following interface connectors:

- The Electronics Industries Association (EIA) RS232 interface or the compatible International Telegraph and Telephone Consultative Committee (CCITT) Recommendation V.24
- The CCITT Recommendation V.35
- AT&T's Dataphone Digital Service (DDS)

On an ICS2 LAC, one cable connects a LAC to the Prime bulkhead, and another cable connects the bulkhead to external equipment. On an ICS3 LAC, the cable to the external equipment is attached directly to the integral bulkhead on the edge of the LAC.

EIA RS232/CCITT V.24

Two different cable configurations are required for RS232/V.24 connection.

The ICS2 LACs require a cable from the LAC to a peripheral bulkhead. The LAC connection is via a 50-pin connector. The synchronous cable terminates (at the bulkhead) in two 25-pin D-type connectors, while the asynchronous cable terminates (at the bulkhead) in four 9-pin D-type connectors. Further cables then connect from the bulkhead to external V.24 or V.35 equipment. Cable numbers and pin assignments are listed in Appendix A, ICS2/3 CABLES AND CONNECTORS.

The ICS3 LACs have integral bulkheading. This means that the asynchronous LACs have four 9-pin D-type connectors for direct connection of cables to terminals, printers, or modems. The synchronous LACs (V.24 and V.35), have two 25-pin D-type connectors, for direct connection of cables to modems.

CCITT V.35/DDS

The CCITT Recommendation V.35 interface is for connection to wideband modems using data transmission rates of 19.2 - 64 Kbps.

The pin assignment of the V.35 cables and connectors is given in Appendix A.

ENVIRONMENTAL SPECIFICATIONS

Table 1-1 shows the environmental specifications for the ICS2/3.

Table 1-1
Environmental Specifications

Operational Temperature:	59-90 degrees F (15-32 degrees C)
Operational Humidity:	30%-80% non-condensing
Storage Temperature:	-40 to +140 degrees F (-40 to +60 degrees C)
Storage Relative Humidity:	0%-95% non-condensing
Vibration: Operational:	0.25g; 5 to 200 Hz Horizontal and Vertical
Vibration: Non-operational:	Vertical; 0.33g 5 to 200 Hz Random Horizontal; 0.68g 5 to 200 Hz Random
Altitude Operational:	10,000 feet
Altitude Non-operational:	40,000 feet
Acoustic Noise:	43 db at 125 Hz 34 db at 250 Hz 34 db at 500 Hz
AC Voltage:	104-127 VRMS 208-254 VRMS
AC Frequency:	60 +/- 1 Hz, or 50 +/- 1 Hz

2

ICS2/3 Software

INTRODUCTION

The ICS2/3 software consists of the following:

- ICS2/3 to the PRIMOS operating system communication software
- ICS2/3 operating system
- IBC microcode
- User protocol-specific modules

The software supports HDLC and BSC framing for the PRIMENET/X.25 communications software, BSC-based RJE protocols for IBM 2780/3780 and HASP, SDLC for PRIME/SNA™ software, and asynchronous communications.

The associated software-supported protocols are

- FDX (Full duplex) with High-level Data Link Control (HDLC) using CCITT X.25 protocols and PRIMENET
- HDX (Half duplex) with IBM's binary synchronous communications protocol (BSC) for IBM 2780/3780/HASP RJE support
- FDX with BSC framing using X.25 protocols (EBCDIC and ASCII)
- SDLC for PRIME/SNA communications with SNA hosts
- ASYNC for terminals, printers and switched-line modems

At Rev. 20.1, the ICS3 with 512K byte memory (or larger) will handle any combination of the allowable protocols (ASYNC, HDLC, SDLC, BSCRJE, and BSCX25). The ICS3 with 256K byte memory can handle every protocol combination except the two largest ones, ASYNC_SDLC_HDLC_BSCRJE_BSCX25 and SDLC_BSCX25_BSC_RJE. The ICS2 limits the combination of protocols (see Chapter 5) to

- Any one protocol.
- ASYNC, HDLC, and SDLC.
- HDLC, SDLC, BSC-framed X.25 (BSCX25), and BSC RJE (BSCRJE).
- Non-reverse-flow-control ASYNC with BSC (BSCRJE and/or BSCX25). (Equivalent to pre-Rev. 19.4 functionality for the ICS2 ASYNC.)

ICS2/3 — PRIMOS SOFTWARE

The software interface between the ICS2/3 processes and PRIMOS processes consists of a message protocol using the Interprocess Queuing and Notification Mechanism (IPQNM) to transfer the data, buffers, and commands. The IPQNM consists of a set of routines that run with the calling process and supporting processes, and use logical connection techniques to communicate with each other.

IPQNM uses high and low priority queues to send information over its logical connections. At least one queue must be specified by a process requesting a logical connection, in which case it is designated as a normal or low priority queue. If there are two queues required, the second is designated as a high priority queue.

- The low priority queue is a default or normal priority queue. Information sent over the logical connection using this queue is received by the other process in the sequence it was sent.
- The high priority queue can be specified in either direction for a logical connection. Any information sent on the high priority queue, by a calling process, is given to the other process before any information still remaining on the low priority queue.

The message protocol consists of a set of control blocks and code words. PRIMOS sends multi-word control blocks to the ICS2/3 to define line configuration and to control synchronous communication lines. The line configuration informs the ICS2/3 about line characteristics such as protocol, data set order, parity selection, number of bits per character, and special characters to be detected. PRIMOS controls the synchronous communication lines by sending information about synchronous I/O operation for the specified communication line, such as transmit/receive, and data set control.

The IPQNM creates the logical connections between communicating processes from information received from the calling process. A maximum of 64 logical connections can be initiated; however, logical connections 0 and 1 are used only for the following specific functions:

- Logical connection 0
  - Initializes the ICS2/3's IPQNM data structure upon request from the calling process
  - Deletes or initializes a logical connection between an ICS2 process and a PRIMOS process
  - Notifies the ICS2/3 process, including logical connection 1, with the event code placed on the event queue
- Logical connection 1
  - Returns transmit buffers to the PRIMOS Buffer Server process
  - Obtains receive buffers from the PRIMOS Buffer Server process

PRIMENET/X.25 Support Software

The ICS2/3 PRIMENET/X.25 software enables PRIMOS to configure lines on the ICS2/3 to support HDLC and BSC-framed X.25 protocols. All ICS2/3-supported PRIMENET/X.25 lines are full duplex.

Note

Half-duplex lines support a Prime-specified BSC protocol used solely for connection to other Prime machines. This is usually referred to as Half Duplex PRIMENET. It will not be supported on the ICS2/3 at Rev. 20.1.

Using PRIMENET/X.25 enables inter-machine communication via any Public Data Network (PDN) that supports the X.25 protocol or via any point-to-point/Prime-to-Prime link that uses synchronous lines.

PRIMENET/X.25 follows a layered construction with Level III (packet/network-level) and Level II (framed/link-level) software residing in Prime, and Level I (physical-level) residing in the ICS2/3 as shown in Figure 2-1. Some of the sequence-number handling of Level II is performed in the ICS2/3.

The ICS2/3 uses the IPQNM to communicate with PRIMOS over high and low priority queues. Each X.25 line has one logical connection from the ICS2/3 to the Prime, and it shares two common free-pool buffers on the host.

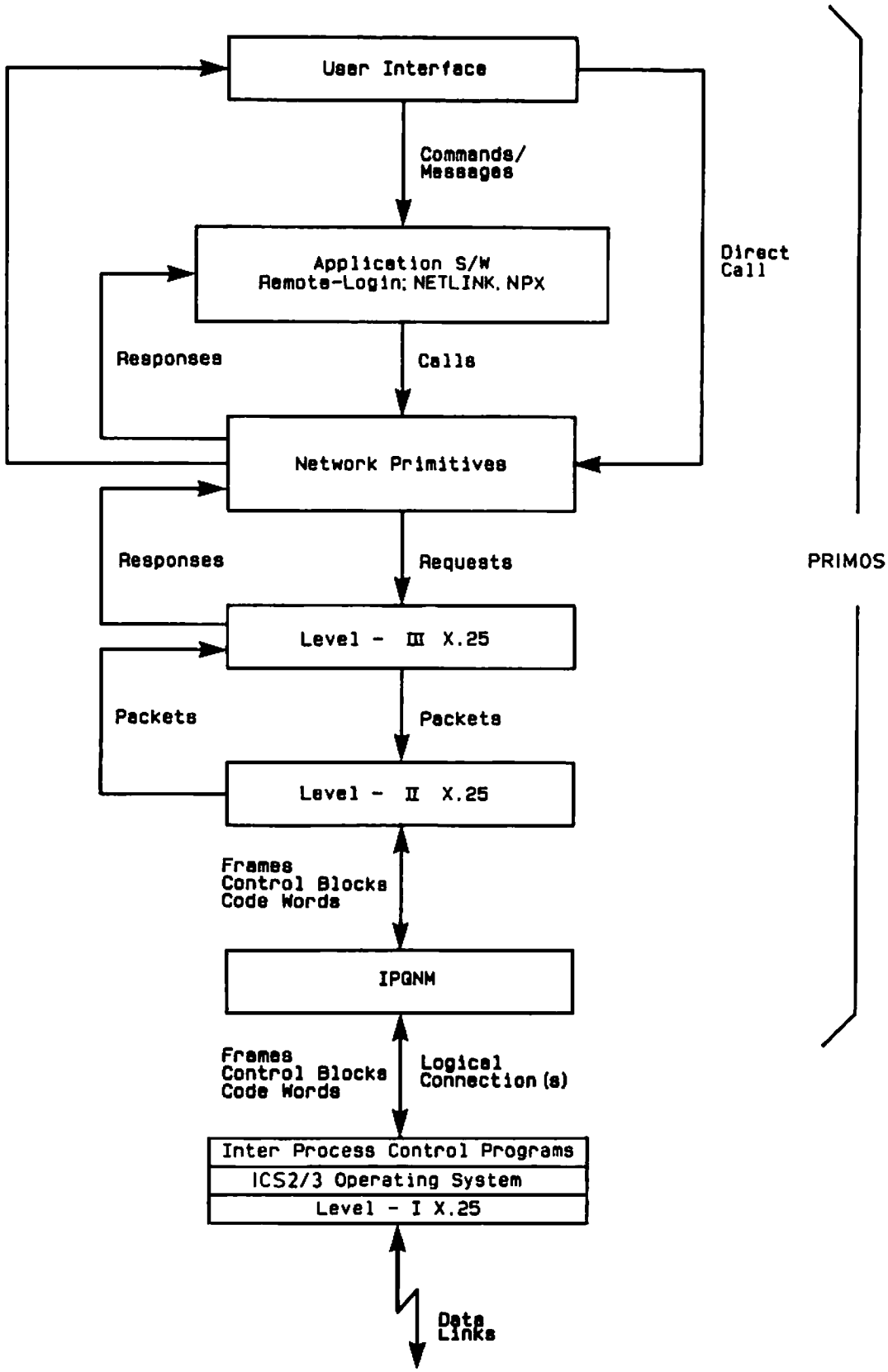
All control messages, X.25 unnumbered frames, and X.25 supervisory frames from PRIMOS are put on the high priority queue, and all X.25 information frames from PRIMOS are put on the low priority queue. All control messages from the ICS2/3 are put on the high priority queue, and all X.25 frames from the ICS2/3 are put on the low priority queue.

Users can invoke PRIMENET/X.25 in two ways. These are

- Through Prime-supplied applications software, such as Remote File Access and remote login, which provide a user-level interface. These packages invoke the network primitives from within PRIMOS and provide a more transparent (friendly) interface to the user.

- By direct call to the network primitives, provided by the user protocol-specific inter-process communication facility. This enables the users to write protocol-specific software modules to customize communication interfaces.

Figure 2-1 shows the PRIMOS -- ICS2/3 software interconnection.



PRIMENET/X.25 Environment
Figure 2-1

RJE Support Software

Remote Job Entry (RJE) emulators allow Prime computers to emulate various RJE terminals/workstations, thus enabling communications between Prime and other host computers. Communications are normally accomplished by the transfer of files from one computer system to another via synchronous links in accordance with the specific RJE data link control protocol.

Prime RJE emulators support the following RJE stations via the ICS2/3:

- 2780 (IBM)
- 3780 (IBM)
- HASP (IBM)

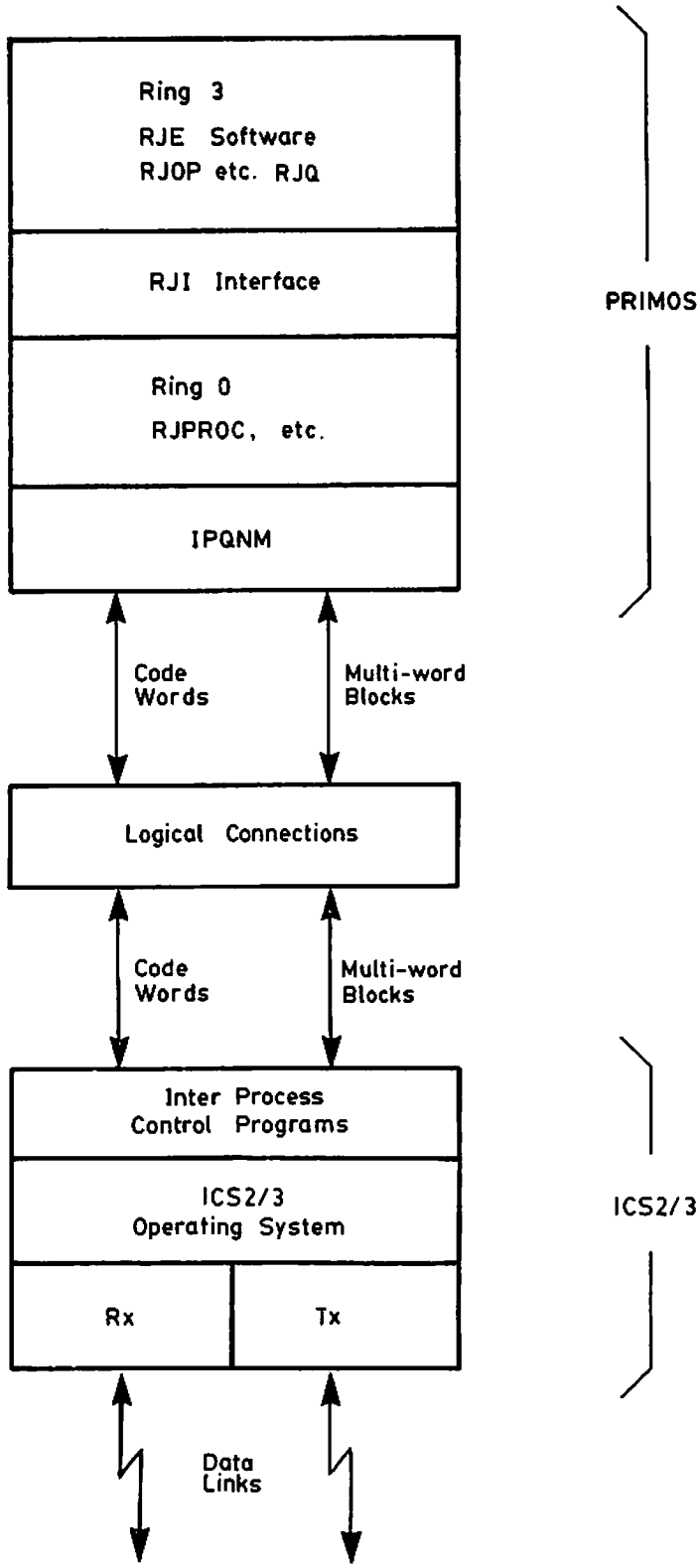
SNA RJE is supported in that the ICS2/3 supports SDLC; however, no support specific to SNA RJE is performed in the ICS2/3.

The ICS2/3 acts as an interface between the Prime systems and RJE hosts connected via synchronous communication links. The ICS2/3 receives data over the communication lines, packs the data in the form of blocks (after doing preliminary checks), and sends the data block to the Prime system. In a similar way, it receives the data and control information from PRIMOS and then transmits it to the specified communication line.

The RJE support software on the ICS2/3 handles the data set monitoring and control for all synchronous communication lines connected to it. ICS2/3 — PRIMOS communication is performed via IPCNM, using one logical connection per line. The RJE support software is responsible for the following functions, which are performed by the ICS2/3 and coordinated by Ring 0 process in PRIMOS:

- Reception and transmission of data over the lines
- Communicating with PRIMOS via IPCNM
- Detecting special character sequences
- Sending line status to the Prime host
- Cyclic redundancy check (CRC) generation and checking
- Detecting loss of synchronization
- Handling dataset control and responses

The RJE environment showing the software functional interconnection is illustrated in Figure 2-2.



RJE Environment
Figure 2-2

ICS2/3 OPERATING SYSTEM

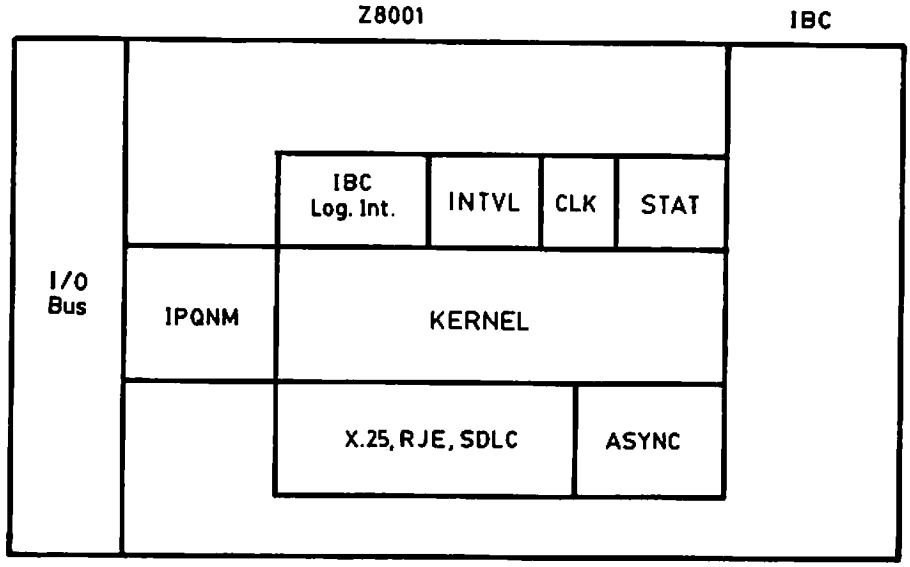
The ICS2/3 OS consists of basically two groups of routines: the kernel operating system and other higher-level processes, such as the following:

- Communication monitors/control processes, such as X.25 and RJE
- Real time clock process (CLK)
- Interval timer process (INTVL)
- OTA handling process
- Boot code
- Status handling software (STAT)

The operating system is based on a multitasking, multipriority timesliced kernel operation, together with additional routines for process-to-process communications, such as IPQNM:

- ICS2/3-to-PRIMOS
- Process-to-process within the boundaries of the ICS2/3

Figure 2-3 illustrates the ICS2/3 OS functional interconnection.



ICS2/3 OS Functional Interconnection
Figure 2-3

The Kernel

The kernel is the lowest level of the operating system structure and is responsible for allocating the resources of the ICS2/3. The ICS2/3 resources consist of the configured memory, the Z8001, and the various interrupts. ICS2 memory is either 128K or 256K bytes, while the ICS3 can have 256K, 512K, or 1024K bytes. The kernel can be considered as part of the processes running underneath the operating system because the kernel tables are shared among all the processes. Kernel code may be executed either on behalf of processes or as part of the actual system. Furthermore, interrupts can occur at any time, even while kernel code is executing; so, kernel data and tables are designed for re-entrant access and modification.

The kernel's primary responsibility is the scheduling of the various processes. Each process is assigned a software priority at process login time, thus allowing the operating system to arbitrate and schedule the processes. The operating system is not responsible for handling I/O operations because each process is responsible for performing its own I/O operations. The kernel does, however, provide standard routines to enable the processes to perform I/O operations correctly.

The kernel has a number of data blocks and linked lists. Each user process consists of a process control block, which contains the process priority; a link to the next process control block; a semaphore pointer; process elapsed time; the current program counter; and saved registers. The kernel maintains a ready list of linked processes ready to be resumed. When a process is to be resumed, the kernel restores the registers, program counter, and flags from the process control block. The ready list is a multipriority list of which there are currently four different priority levels. However, among the process control blocks of the same priority level, dispatching is first-come first-served. When a process has "completed" (which requires the occurrence of a specific event), it waits on a semaphore. The kernel then places the process control block on the appropriate wait list and invokes the next ready process. Note that there is no multiple wait; that is, a process can only sequentially wait on a number of different events. When the event associated with a semaphore (triggered by an interrupt) occurs, the kernel notifies the process by removing it from the wait list and returning it to the ready list.

Process Scheduling: A process or task is the basic logical unit controlled by the operating system. A process, which is a logically complete execution path through the code, may demand the use of the system resources via the kernel routines. A process is not the same thing as a program. A program is merely a section of code, whereas multiple processes may proceed through the same code.

The kernel uses a preemptive scheduling algorithm. A running task can be suspended by a higher priority task, by interrupt code (unless the task specifically disables interrupts), or by a system clock that limits the run time of a task (the time slice allocated to that task at login). This scheduling system ensures that no task can capture all the processing time, and that every process gets a chance to run.

The current configuration of the operating system allows 64 processes on four priority levels.

Semaphores: The ICS2/3 OS uses semaphores to synchronize its operation, which results in the following:

- Serialization of all accesses to the state variables, collecting all accesses into the critical regions guarded by that semaphore
- Guarding against illegal transitions

Assigning a semaphore to a process serializes all accesses and protects against illegal transitions; the semaphore is decremented after each critical region that precedes a possible illegal transition. The decremented semaphore will block the process if it was not incremented inside the critical region. Any cases of ambiguity have to be resolved by using the appropriate scheduling discipline. The ICS2/3 employs semaphores for the following:

- Clock and timers
- Time slice
- Login
- Free space lock
- OTAs, INAs
- IBC
- DMX
- LC0, LC1
- IBC error status
- Various other resources, such as queues

Operating System Services: The ICS2/3 OS also contains routines to do the following:

- Allocate/deallocate free memory according to Knuth's boundary control word scheme
- Handle INAs, OTAs, interrupts, DMX, queue manipulation, and process-to-process communications
- Provide logical interrupt and interface capabilities between the IBC and Z8001 code

Cold Start

At cold start, PRIMOS initiates the PROM code in the ICS2/3 by issuing an Output Control Pulse (OCP '14). This checks the logic and verifies the Z8001 memory. After checking the Z8001 memory, PRIMOS carries out the downline load. This entails copying the complete downline load file into the Z8001 memory, starting at segment 40H (hex notation). The downline load file consists of the following:

- Z8001 OS and protocol handlers
- Z8001 verify code (runtime verify)
- IBC verify microcode
- Operating IBC microcode

On completion of the diagnostic checks, a 16-word table containing the LAC card cage configuration is sent to PRIMOS. Also, the ICS2/3 software initializes the kernel data bases, which are derived from the table of user processes. These user processes include the following:

- Clock manager
- OTA handling process
- Buffer handling process
- Protocol managers
- Communications monitor/operating system diagnostic process for Prime customer service personnel
- IBC status handling process

Warm Start

Warm start enables ICS2/3 recovery after the initial boot. PRIMDS initiates the PROM code in the ICS2/3 by issuing an Output Control Pulse (OCP '17). Warm start is similar to a cold start, except that the PROM code in the Z8001 does not run a full diagnostic test.

IBC MICROCODE

The IBC is an intelligent multi-channel DMA (Direct Memory Access) device controlled by a sequencer and ALU (see Chapter 1). It interfaces the Z8001 to the data links via the LAC bus, and also interfaces to/from the Prime memory via the Prime I/O bus.

The IBC microcode is booted by the Z8001 during PRIMDS cold and warm starts. The IBC microcode, when booted, should contain all the protocols that will be used on the ICS2/3. It should include those protocols to be brought up after a PRIMDS cold start.

The Z8001 performs several tasks with respect to the IBC. The first two tasks of configuring IBC pointers and configuring the LAC bus are accomplished during ICS2/3 initialization, that is, before the ICS2/3 operating system is fully up.

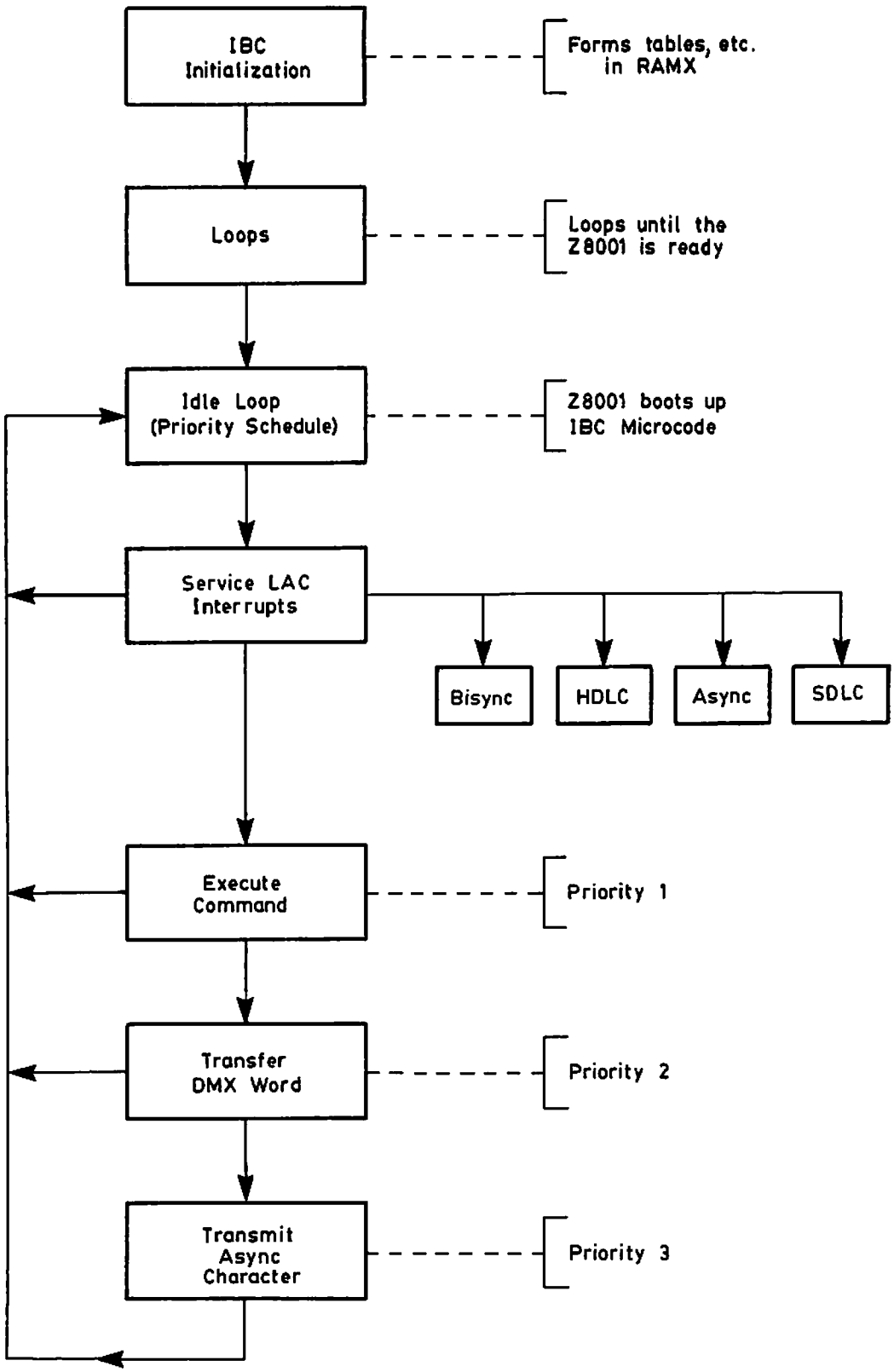
First, the Z8001 supplies both primary and secondary status buffers for the IBC. The status buffers are used by the IBC to pass status (for instance, good CRC or bad CRC) back to the Z8001.

Second, the Z8001 must characterize the LAC bus. Characterization of the LAC bus means determining LAC locations and type of LAC.

LACs are of three basic types: ICS2 four-line asynchronous LACs, ICS3 four-line asynchronous LACs, and ICS2/3 two-line synchronous LACs. ICS2 asynchronous LACs use Zilog's Serial Input/Output (SIO) chip, whereas the ICS3 asynchronous LACs use Zilog's Serial Communications Controller (SCC). They have different IDs so the IBC code can identify which type are installed. ICS3-type LACs are usable only on an ICS3, whereas the ICS2 type can be used with either the ICS2 or ICS3 controller board. To characterize the LAC bus, the Z8001 runs channel programs to read the LAC ID registers for each possible backplane slot. It builds a table, mapping physical lines on the LACs as they are found to logical line numbers.

Third, the Z8001 must initialize the MAP RAM, write the line file, and issue the initialize-a-line command for each physical line. For each line, it then runs channel programs to initialize the communications lines to establish protocol type, baud rate, and loop back control. The MAP RAM establishes the mapping between the logical and physical lines. Initialization of a line is the function of the individual protocol handler process and not of the ICS2/3 OS.

The IBC microcode functional interconnection is shown in Figure 2-4.



IBC Microcode Functional Interconnection
Figure 2-4

USER PROTOCOL-SPECIFIC MODULES

At Rev. 20.1, the ICS2 and ICS3 will support HDLC and BSC-framed PRIMENET/X.25, BSC-based RJE, SDLC, and asynchronous protocols. To achieve this, the ICS2/3 software incorporates four types of user protocol-specific software modules:

- X.25 modules
- RJE modules
- SDLC
- Asynchronous modules

Descriptions of the protocol-specific modules are included in the following pages to give the technical reader a basic understanding of the protocol-dependent software in the ICS2/3 and its function. This information is not necessary for the configuration and operation of the ICS2/3.

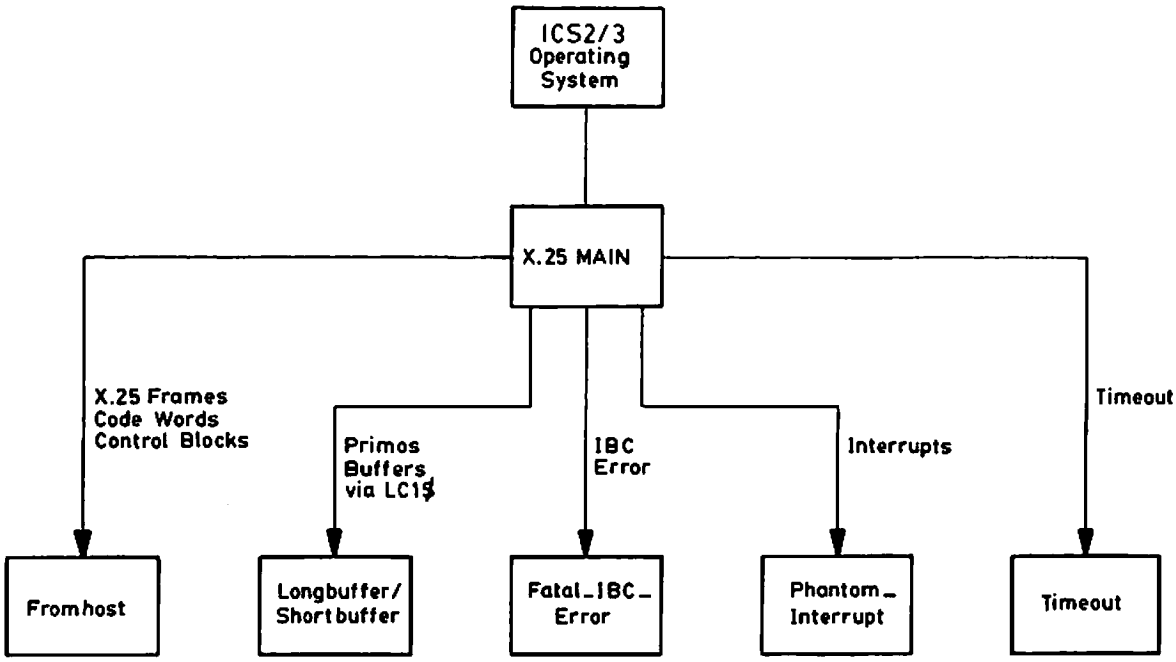
X.25 Protocol-specific Modules

The following text gives brief descriptions of the various X.25 software modules resident in the ICS2/3. Figure 2-5 illustrates the software module interconnection of the ICS2/3 X.25 protocol.

When the ICS2/3 has been booted, it sets up logical connections to the Prime CPU for the maximum number of X.25 lines that can be supported by an ICS2/3, together with a single logical connection to the buffer server process for the purpose of accessing PRIMOS buffers. When the Prime CPU attempts to start up an X.25 line, it places a configuration block on the high priority queue for that line.

The main X.25 process, X25MAIN, is notified of this event, dequeues the block, and attempts to configure the line. If the attempt succeeds, then X25MAIN awaits notification of another event, such as

- A code word from the host
- An X.25 information frame to be transmitted
- An X.25 unnumbered or supervisory frame to be transmitted
- An interrupt received from the IBC for the specific line
- An IBC error
- A buffer received from the host
- A timeout



Overview of the ICS2/3 X.25 Software
Figure 2-5

From the item enqueued on the event queue, X25MAIN can determine the cause of the notification and then call the appropriate function to process it.

If the event that caused notification is an X25 frame to be transmitted, then the Z8001 DMXs the frame into ICS2/3 memory and queues the frame for output by the IBC via the appropriate line.

When the IBC receives a frame from the line, the frame is buffered in ICS2/3 memory and the IBC interrupts the Z8001. The Z8001 then DMXs the frame to the host and notifies the host that a good frame is available.

Descriptions of the main ICS2/3 modules follow:

Module	Description
X25MAIN	This is the main ICS2/3 X.25 process. Its primary purpose is to set up logical connections to the host for each X.25 line, and logical connections within the ICS2/3 to the buffer server process, the IBC logical interrupt handling process, and the timer process. X25MAIN waits for notification of an event (for example, initialization of a line) or for a block to transmit, and then calls the appropriate routine. Furthermore, X25MAIN sets up three special queues to enable rapid determination of outstanding buffer requests: • Large PRIMOS buffers • Small PRIMOS buffers • ICS2/3 transmit buffers
FROMHOST	This routine processes a notification from the host and decides whether the host has sent a code word, control block, or transmit frame.
TXMSG:	This routine controls the transmission of an X.25 message over the synchronous link.
IBC_INI	This routine initializes the IBC to handle the configuration block for either HDLC or BSC-framed X.25.
CODE1BLK	This routine configures the specified line by using the control block sent from the host. The host is notified of successful completion of initialization or of any error conditions that are detected.

Module	Description
PHANTOMINTERRUPT	This routine is called by X25MAIN to process IBC interrupts, namely • A completed frame transmission • A received message • a changed data channel status
FLUSH_XMIT_Q	This routine flushes the transmit queue by returning all transmit buffers to the free pool and setting the chain pointer to zero.
GET_BUFFER	This routine consists of two functions that obtain a long or short PRIMOS buffer Queue Control Block (QCB) respectively.
STATISTICS	This routine updates the specified counters block field by 1, and where necessary, updates the interrupt status block. The ICS2/3 collects large numbers of statistics that may be made available to the user in future releases of PRIMOS.
APPEND_TX_CHAIN	This routine appends a buffer to the transmit chain that is pointed-to from the Synchronous Line Control Block.
CTLWORD	This routine processes control words sent from the host to the X.25 process on the ICS2/3.
SHORTBUFFER	This routine processes short PRIMOS buffers sent from the host.
DECON	This routine is called when a line is ready for complete deconfiguration; that is, when there are no outstanding requirements for host buffers.
LONGBUFFER	This routine processes long PRIMOS buffers sent from the host.
TIMEOUT	This routine is called when X25MAIN receives notification of a timeout.
DUMP_COUNTERS	This routine dumps counters to the host using large PRIMOS buffers.

Module	Description
ICS2_FATAL_IBC_ERROR	This routine handles any serious IBC errors by emptying the transmit and receive chains of all lines, returning the buffers to the free pool, and resetting the lines.
CODE_WORD	This routine sends a code word to the host by enqueuing the code word on the host's high priority queue.
UPD_NR	This routine updates the N(R) fields in all the buffers queued for transmission by the IBC.
UPD_INTS	This routine inserts a character into the interrupt status block. When the block becomes full, it attempts to DMX it to the host.
DUMP_INTERRUPTS	This routine dumps the interrupt block to the host.

RJE Protocol-specific Modules

The following paragraphs provide a simplified explanation of the RJE logic in the ICS2/3. This description does not apply to PRIME/SNA RJE.

RJE in the ICS2/3 is a collection of state-driven modules. Logical states are changed when particular events occur, such as

- A block sent from PRIMOS
- Data Set Status (DSS) changing
- IBC-controlled line activity, such as, transmit completion, receive starting, and receive completion.

When an RJE session starts, PRIMOS establishes a logical connection with the ICS2/3 RJE process and sends a configuration block for the line concerned. This configuration block is validated and a configuration-accepted response is returned to PRIMOS.

Once the line configuration has been accepted, either PRIMOS or the remote host can send data; whichever end starts sending data causes the logic to "flip-flop" between transmit and receive states. This reflects the half-duplex nature of RJE protocols.

For example, if PRIMOS sends the ICS2/3 a block, it will be transmitted, and then the ICS2/3 RJE process will look for receive data. When input from the line occurs, the received data/response is sent to PRIMOS, and the software waits for another block from PRIMOS.

Responsibility for detailed checking of protocol violations is left with the appropriate RJE support in PRIMOS; the ICS2/3 is concerned only with receiving and transmitting data which conforms to general RJE-defined formats.

When the RJE session is completed, PRIMOS sends a deconfigure command, and the ICS2/3 releases the line for use by other protocols.

The following sections briefly describe the RJE software modules which reside in the ICS2/3.

RJE_MAIN: The RJE_MAIN routine is the primary RJE control process in the ICS2/3. It sets up one logical connection with PRIMOS for every physical line that RJE can support (currently eight). Furthermore, it has one semaphore that, together with its event queue, enables RJE_MAIN to control the following:

- All RJE activity for logical connections, such as, data blocks to/from PRIMOS
- All input and output activity on the lines, for example, Binary Synchronous Communications (BSC) blocks and control sequences

The input/output activity on a line is controlled using the following data structures:

- LCCB -- Logical Connection Control Block
- SLCB -- Synchronous Line Control Block

The SLCB contains line-oriented information such as state, timer, and LCCB pointer. Each line may transit seven states, where each state incorporates procedures to handle specific events determined by received or transmitted data or dataset changes.

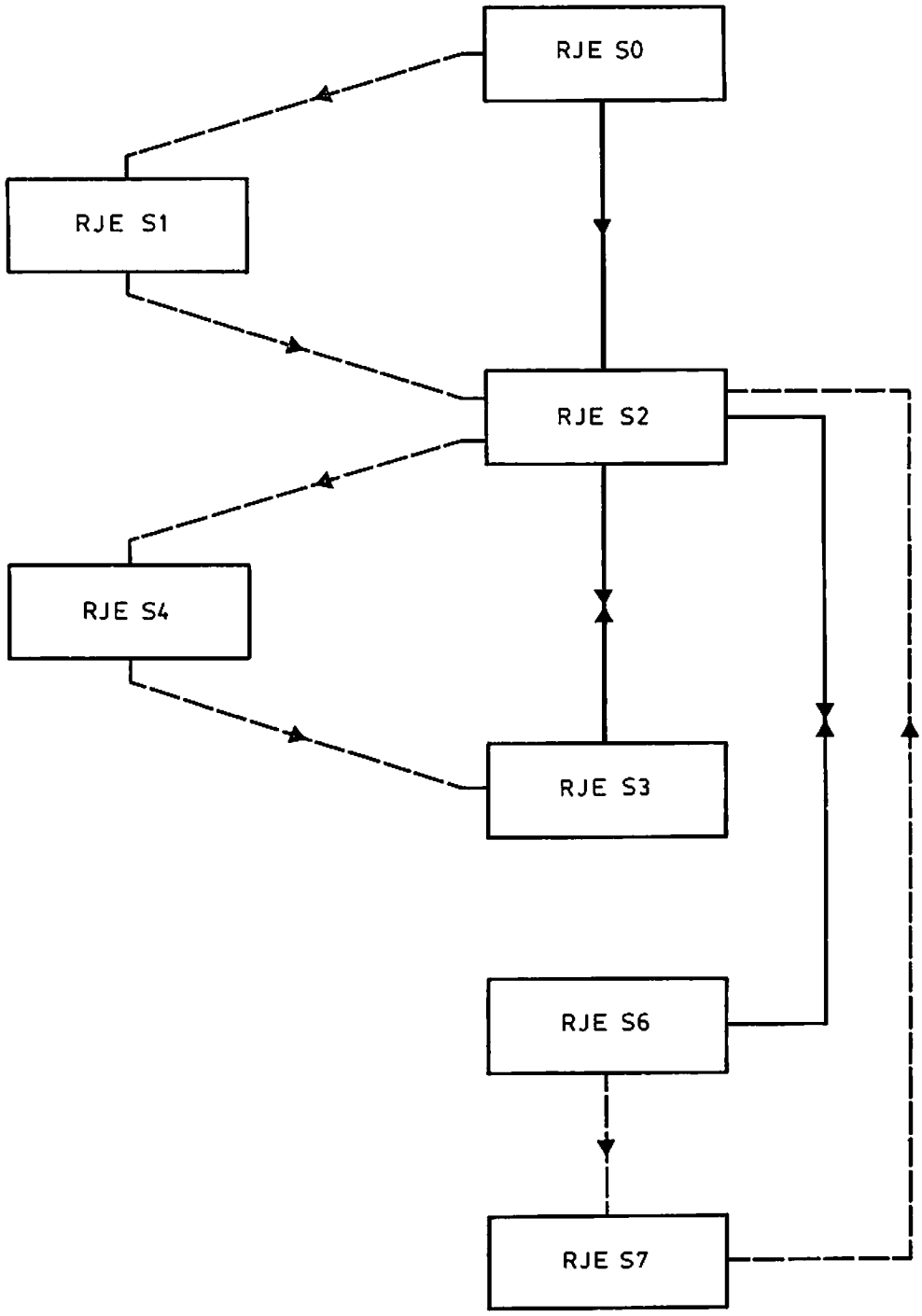
The line state flow, during a normal RJE session, is shown in Figure 2-6. Note that the following exception conditions usually abort the connection to the remote site.

- More than one output string, at any one time from PRIMOS, is considered a protocol error (because BSC is HDX).
- Either a deconfiguration command or a logical connection delete causes the line to be deconfigured.

State Descriptions: The following descriptions, together with Figure 2-6, detail the possible states for an RJE line and some of the events that can cause state transitions. For simplicity, only the basic system is described; the error and exception paths are not mentioned.

State	Description
RJE_S0	(Configure the Line): A line will have its SLCB set up and its statistics initialized. If DSS is good for receive, the system will transit to RJE_S2, otherwise to RJE_S1.
RJE_S1	(Wait for good RCV DSS): The line's DSS is periodically checked for good RCV DSS. When RCV DSS is good, the system transits to RJE_S2.

State	Description
RJE_S2	(Initiate Receive, Transmit, Deconfigure): DSS good for receive is assumed on entry to this state, meaning that the host has data to transmit, which it does when transition to RJE_S3 occurs. If the host has no data to transmit, and if the IBC indicates a Receive_Started status, the RJE system will receive by transiting to RJE_S6.
RJE_S3	(Transmit, wait for transmit completion): Data is being transmitted by the IBC, and when transmission is complete the RJE system will turn around and wait for received data in RJE_S2.
RJE_S4	(Wait to Transmit): This state is entered when the host has data to transmit. The system waits until DSS is ready to transmit and then transits to RJE_S3.
RJE_S5	This state is not used at Rev 20.1.
RJE_S6	(Receive, wait for receive completion): Data is being received. When receipt is complete, and if DSS is idle, the RJE system transits to RJE_S2. If DSS is not idle, the RJE system transits to RJE_S7.
RJE_S7	(Wait for Idle DSS): The system waits for DSS to become idle. When DSS becomes idle, the RJE system transits to RJE_S2.



RJE Software States Interconnection
Figure 2-6

SDLC Protocol-specific Modules

This section briefly describes PRIME/SNA software and its use of SDLC protocol implemented in the ICS2/3. For a more detailed overview, please refer to the PRIME/SNA System Administrator's Guide.

SDLC (Synchronous Data Link Control) is the line protocol used by PRIME/SNA for connection to an SNA host.

SDLC protocol can be used in both point-to-point and multipoint configurations on half-duplex and full-duplex electrical connections. An IBM host (not necessarily other SNA hosts) restricts switched line connections to half-duplex, point-to-point.

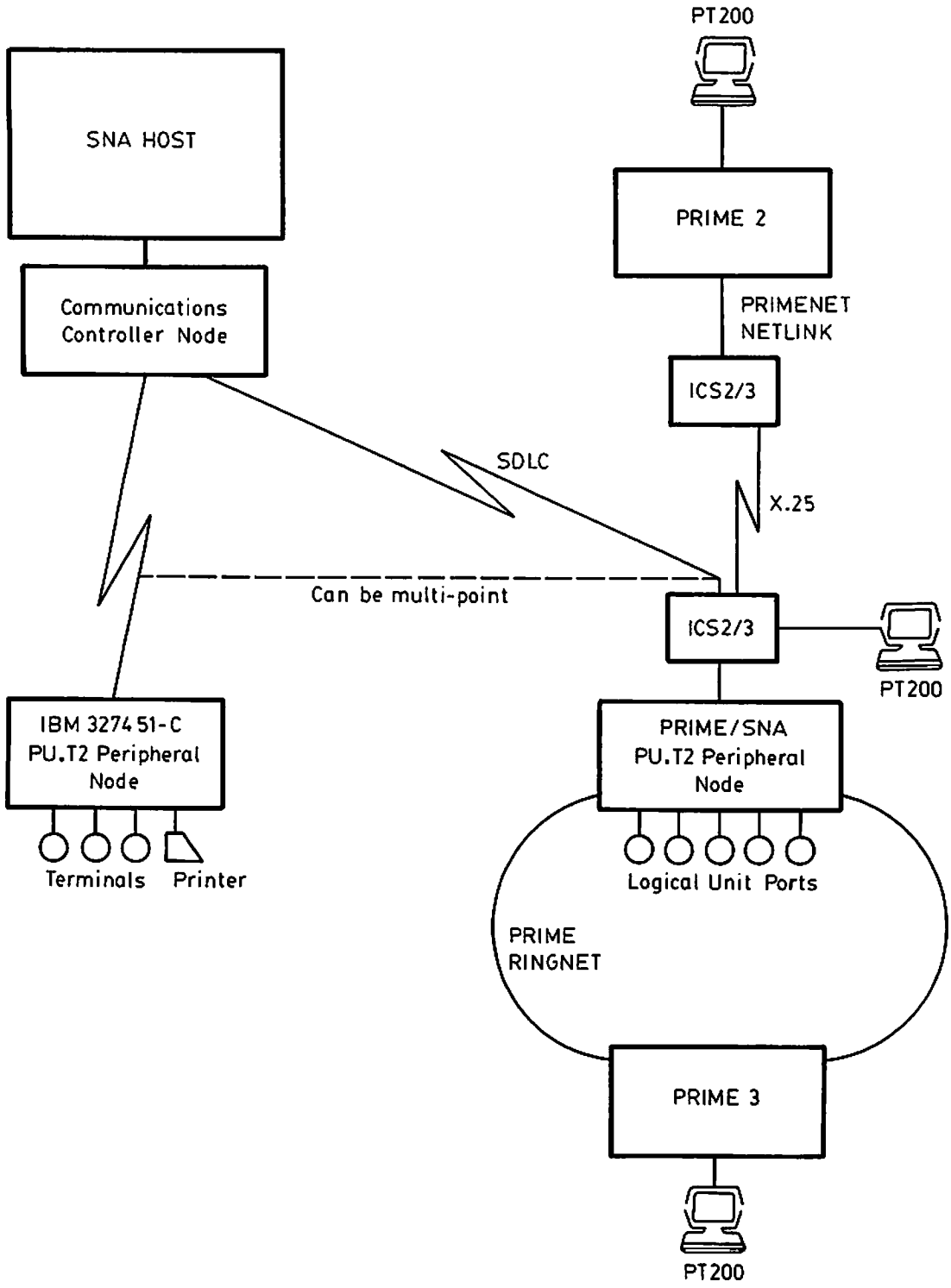
A Prime system using PRIME/SNA emulates an SNA Physical Unit Type 2 (PU.T2) Peripheral Node. A peripheral node, by SNA definition, is one that requires the support of an SNA host.

PRIME/SNA uses SDLC as its data link control for communicating with an SNA host. As such, for each peripheral node that PRIME/SNA is supporting, there is a unique SDLC secondary station communicating with an SDLC primary station.

PRIME/SNA to SNA Host Connection: Figure 2-7 shows the connection between an SNA host and a Prime system using PRIME/SNA. Three Prime systems are shown. One is running PRIME/SNA and has direct access to the SNA host; the other two Prime systems have access to the SNA host by using either PRIMENET/NETLINK (Prime 2) or Prime RINGNET™ (Prime 3) to communicate with PRIME/SNA in the first Prime system. Each of the three PT200™ terminals shown has access to the SNA host as an interactive terminal. An IBM 3274 51C Peripheral Node is also shown to illustrate the fact that the Prime running PRIME/SNA is emulating a PU.T2 Peripheral Node.

The PRIME/SNA Interactive Subsystem enables the Prime to emulate the IBM 3270 Information Display System (using the Prime PT200 terminal as the interactive device) and IBM 3287 and 3289 printers. The PRIME/SNA RJE Subsystem enables the Prime to emulate the IBM 3770 Remote Job Entry workstation. The SNA network regards these subsystems as Logical Units (LUs). The subsystems comply with the SNA definition of an LU by implementing the facilities of specific LU types.

The PRIME/SNA Server subsystem allows the Prime to have from one up to eight peripheral nodes attached to the SNA network.



Connection between Prime (using PRIME/SNA) and IBM Host
Figure 2-7

SDLC Protocol Handling in the ICS2/3: All of the SDLC protocol handling is done in the ICS2/3 without interrupting the Prime CPU. The PRIME/SNA Server, which executes in the Prime CPU as a Ring 3 phantom process, allows the PRIME/SNA administrator to configure the information needed to communicate with SNA host systems. Much of this information is passed to the SDLC support when a connection is started so that correct line and SDLC station handling is performed.

SDLC support is possible whenever the downline load file in the ICS2/3 contains SDLC capability. (See the SYNC CNTRLR directive in Chapter 5.) In the ICS3, SDLC can be supported together with any combination of allowable protocols from the selection of HDLC, ASYNC, BSCX25, and BSCRJE. A 512K-byte RAM is required for all protocols to run together; the 256K-byte RAM will handle any combination except all protocols together. The ICS2 also has minor limitations due to its maximum memory size of 256K bytes. (See USER PROTOCOL-SPECIFIC MODULES, earlier in this chapter, and Chapter 6, PERFORMANCE AND RESTRICTIONS.)

SDLC protocol support is accomplished, in the ICS2/3, by three processes, SDLMAN (SDLC protocol handler), and SDLMTR (interrupt handler for SDLMAN, which consists of two processes, one for receive, and one for transmit).

SDLMAN: SDLMAN is the main component of the SDLC handler. It sends and receives messages to and from the PRIME/SNA SERVER, via the IPQNM. These messages include the following:

- Link start and stop commands.
- Link active commands.
- Link active acknowledgement.
- Data messages. SDLMAN formats the frames (except for the FLAGS and FCS, which are generated by the LAC hardware).
- Link statistics requests.
- Link error.

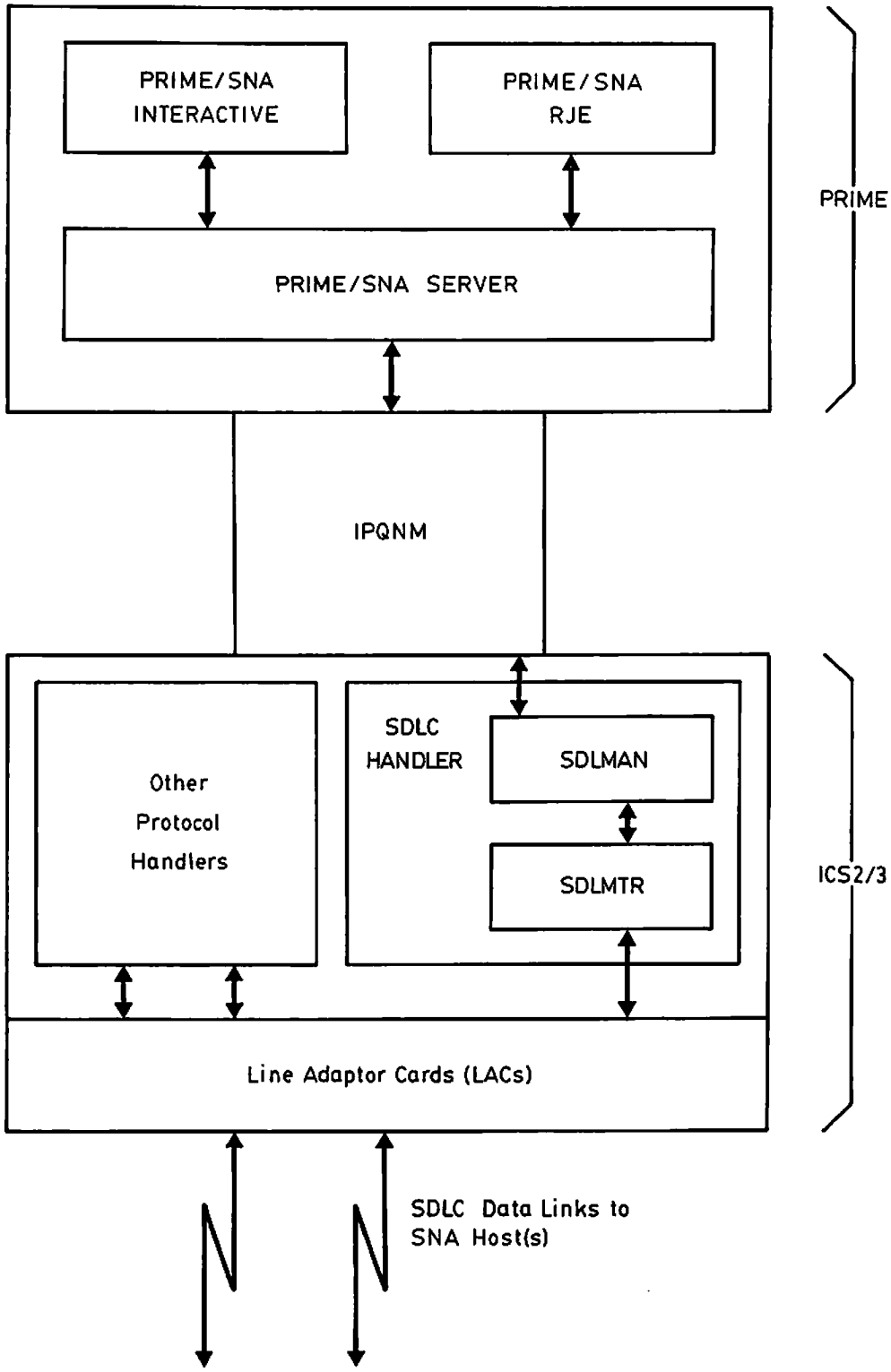
SDLMTR: SDLMTR is the interrupt handler for SDLMAN. It performs the following functions:

- Sends and receives messages to and from SDLMAN
- Handles synchronous LAC initialization, including NRZI line signalling and internal (ICS2/3) or external (modem) clocking
- Initializes line status and handles RTS, CTS, DTR, RI, and DSR
- Sends and receives messages to and from the remote SNA host
- Checks that the maximum or minimum number of bits are present in frames
- Supervises data buffers and informs SDLMAN which should be saved and which are available for use

IPQNM: The IPQNM provides the path for messages between the PRIME/SNA Server and SDLMAN, and for process-to-process communication on the ICS2/3 controller. The data structure used between the Server and SDLMAN is the SDLMAN Message Block (SDLMB). The SDLMB contains information and data about processing that is about to occur or has already occurred.

Protocol Tokens: The protocol token, SDLC, used in the SYNC CNTRLR command, causes the required downline load file to be loaded into the ICS2/3 memory from the DOWN_LINE_LOAD* directory. Files are named ICS2_NN.DL or ICS3_NN.DL for the ICS2 and ICS3 respectively, where NN is the specific file number for a particular protocol or combination of protocols.

Figure 2-8 illustrates the PRIMOS — ICS2/3 software connection.



PRIMOS — ICS2/3 Software Connection
Figure 2-8

Asynchronous Protocol-specific Modules

Asynchronous protocol processing by the ICS2/3 is possible whenever the downline-loaded file in its memory contains ASYNC protocol capability. Most of the asynchronous protocol processing is done in PRIMOS. (See Chapter 3, PRIME HOST SOFTWARE, for more detail.) Support for the ICS controllers is provided by a Device Interface Module (DIM), called ASYNDM. (A similar DIM, called AMLDIM, supports asynchronous protocol processing for the AMLC.) Figure 2-9 shows the connection of the ICS2/3 for asynchronous processing in PRIMOS.

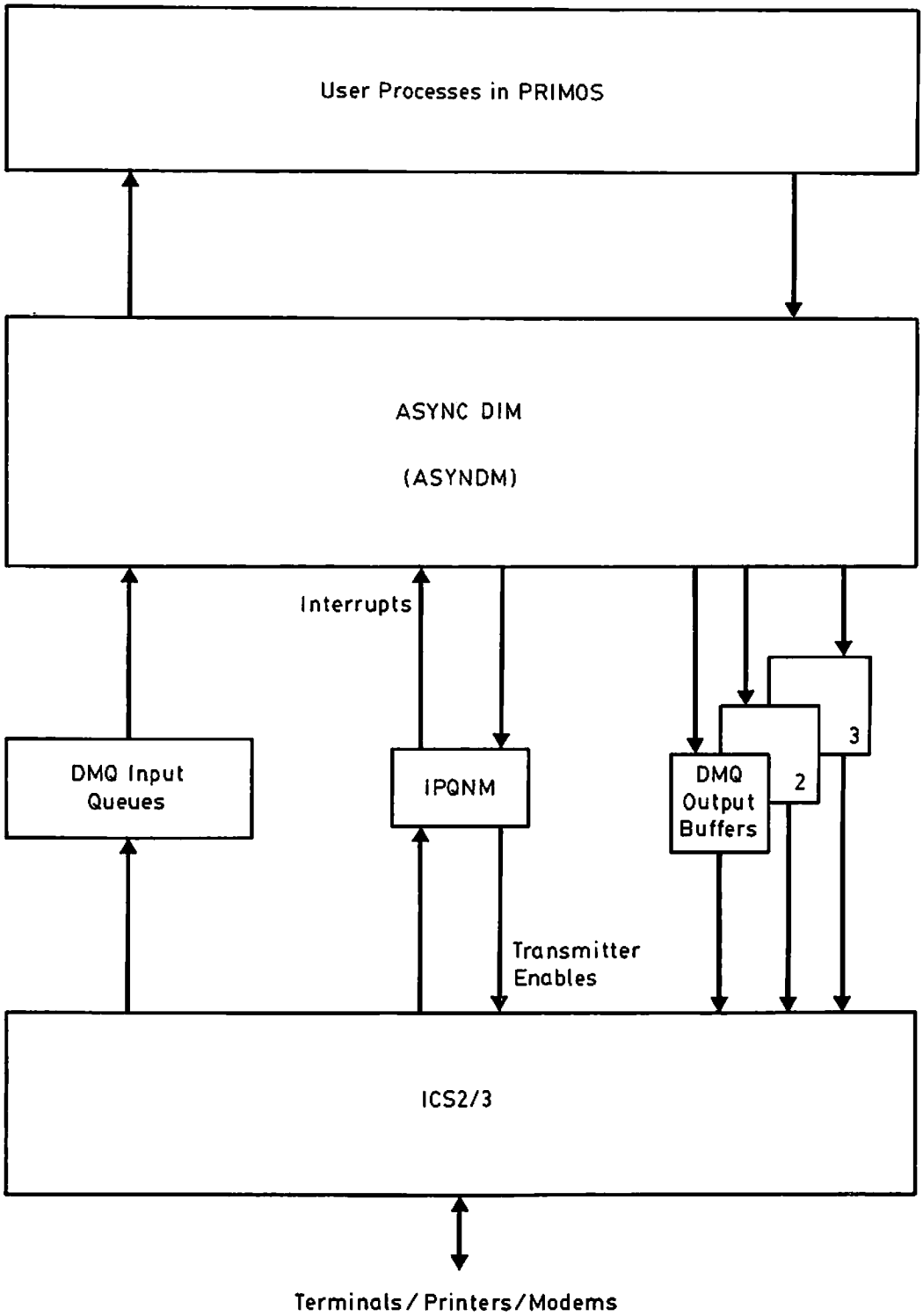
Each ICS2/3 is considered as if it were from one to eight "virtual" ICS1s, each with up to eight asynchronous lines.

Interrupt-driven Processing: Asynchronous support operates at the character level, handling one character at a time as it is received from the terminal. Characters are transferred from the ICS controller to the Prime memory using the Prime I/O bus. Once in memory, the ASYNDM process is started by periodic interrupts from the ICS controller, and processing of input data is initiated. These periodic interrupts are passed to PRIMOS using the IPQNM. The interrupt rate is set using the ICS INTRPT directive, which specifies the number of interrupts per second. The default and minimum is '12 (10 decimal), which gives an interrupt rate of ten per second. The maximum is '144 (100 decimal), which gives an interrupt rate of 100 per second. A balance/compromise has to be reached with the interrupt rate being set high enough to clear the amount of input data and yet not leading to over-interruption of the ASYNDM process to the detriment of other processes.

Input Data Queues: Data from the ICS to the Prime is buffered, in DMQ input queues, under the control of a buffer manager. The ICS1 uses a single buffer for all of its eight asynchronous lines. The ICS2/3 uses two input queues for each "virtual" ICS1-to-PRIMOS connection. That is, up to eight asynchronous lines are multiplexed into a pair of input queues.

These queues are managed as a double-buffer scheme where the ICS2/3 can be writing to one queue while PRIMOS is reading from the other.

The directive, ICS INPQSZ n, is used to vary the size of all the input queues from the default of '77 (63 decimal). The value of n must be in octal, one less than a power of two (such as '177 or '377), and less than or equal to '1777. Increasing the size of the input queues may be necessary to avoid losing ICS-to-Prime data when the input rate or volume is high, such as when several terminals are doing page transmissions.



Asynchronous Protocol in the ICS2/3
Figure 2-9

Reverse Flow Control (RFC) can also be used to regulate the flow of data from terminals. When using RFC, the ICS controller sends XOFF (CONTROL S) and XON (CONTROL Q) to the terminal to stop and resume transmission, respectively. The aim is to prevent overrun of the DMQ input queues. RFC is enabled by setting bit eleven of the configuration word in the AMLC command directive. (See the System Administrator's Guide.)

The queue size is the number of sixteen-bit words in the queue. The right half of the word contains the character; the left half of the word contains the line ID and flags that indicate if the right half contains a normal character or a special control character.

Output Data Queues: Data from PRIMOS to the ICS is buffered in a DMQ output buffer; one buffer for each line. The AMLBUF command varies the size of this DMQ buffer (and other buffers) for a specified line. IPQNM commands, to enable the transmitter, assist in the transfer of transmit data from the DMQ output buffers to the ICS.

PRIMOS Commands to the ICS: Commands from PRIMOS to the ICS are conveyed through the IPQNM. These commands relate to the following:

- Setting the line configuration, line number, reverse flow control, line looping, number of stop bits, parity, and character length
- Setting the baud rate between 50 and 19,200 bps
- Setting line control, transmit enable/disable, echo on/off, receive enable/disable, reporting receiver disable
- Interrupt polling and control, periodic interrupts on/off, polling period
- Dataset control, raising/lowering the dataset control lead

3
Prime Host Software

INTRODUCTION

The Prime host software associated with the ICS2/3 provides software support for the establishment and control of PRIMENET/X.25 software, RJE, SDLC, and asynchronous operations.

PRIMENET/X.25 IN PRIME HOST

PRIMENET is a generic term for the complete networking facilities available to users of a Prime computer system. PRIMENET/X.25 enables inter-machine communications via any Public Data Network (PDN) that supports X.25 protocol or via any point-to-point/Prime-to-Prime link that uses synchronous lines. PRIMENET/X.25 caters for Full Duplex (FDX) communications and Half Duplex (HDX) operation. Half duplex PRIMENET, however, is not supported on the ICS2/3 at Rev. 20.1.

Note

PRIMENET also supports a ring network that utilizes node controllers for local area computer configurations.

There are four functional activities in PRIMENET/X.25 processing:

- Initialization
- Startup
- Active Processing
- Shutdown

Initialization

PRIMENET/X.25 initialization includes two procedures:

- Boot from a cold start
- Restart/Recovery from a warm start

Boot: A boot is a cold start operation that starts a full system initialization. Booting includes allocation of system resources and downline loading of any intelligent controllers configured for the network. It also initializes the Device Interface Modules (DIMS) and saves information for the warm start process.

Restart/Recovery: A restart, or recovery, is a warm start operation that is concerned with restarting or recovering PRIMENET and/or any of the attached communication subsystems. Typically, a line controller might have gone down; therefore, under those circumstances, an operator may initiate a warm start, rather than going through the full boot procedure. When a warm start commences, routines are activated to initiate system and process aborts. Communication subsystems are restarted by first halting active processes, then disconnecting the logical connections to the ICS2/3. The software modules are reloaded and the logical connections reestablished.

Startup

Startup is activated by the Network Manager (after initialization) when the operator inputs a START_NET command. Startup involves allocating network-oriented resources and firing up network processes. These processes control and manage the whole framework of the network giving users X.25 support facilities. Some of the activities performed at startup are described briefly under Restart/Recovery. One of the more important aspects of startup is reading the network configuration file to determine what controllers, protocols, and resources are allocated to the network. This file (created by the System Administrator) enables the correct data structures to be set up and determines the network's operational environment.

Active Processing

Two mechanisms are available which enable users to invoke PRIMENET/X.25 when PRIMENET is active. The mechanisms operate

- Through applications software such as Remote-login, NETLINK, and NPX. These packages invoke the network primitives from within and provide a more transparent (friendly) interface to the user.

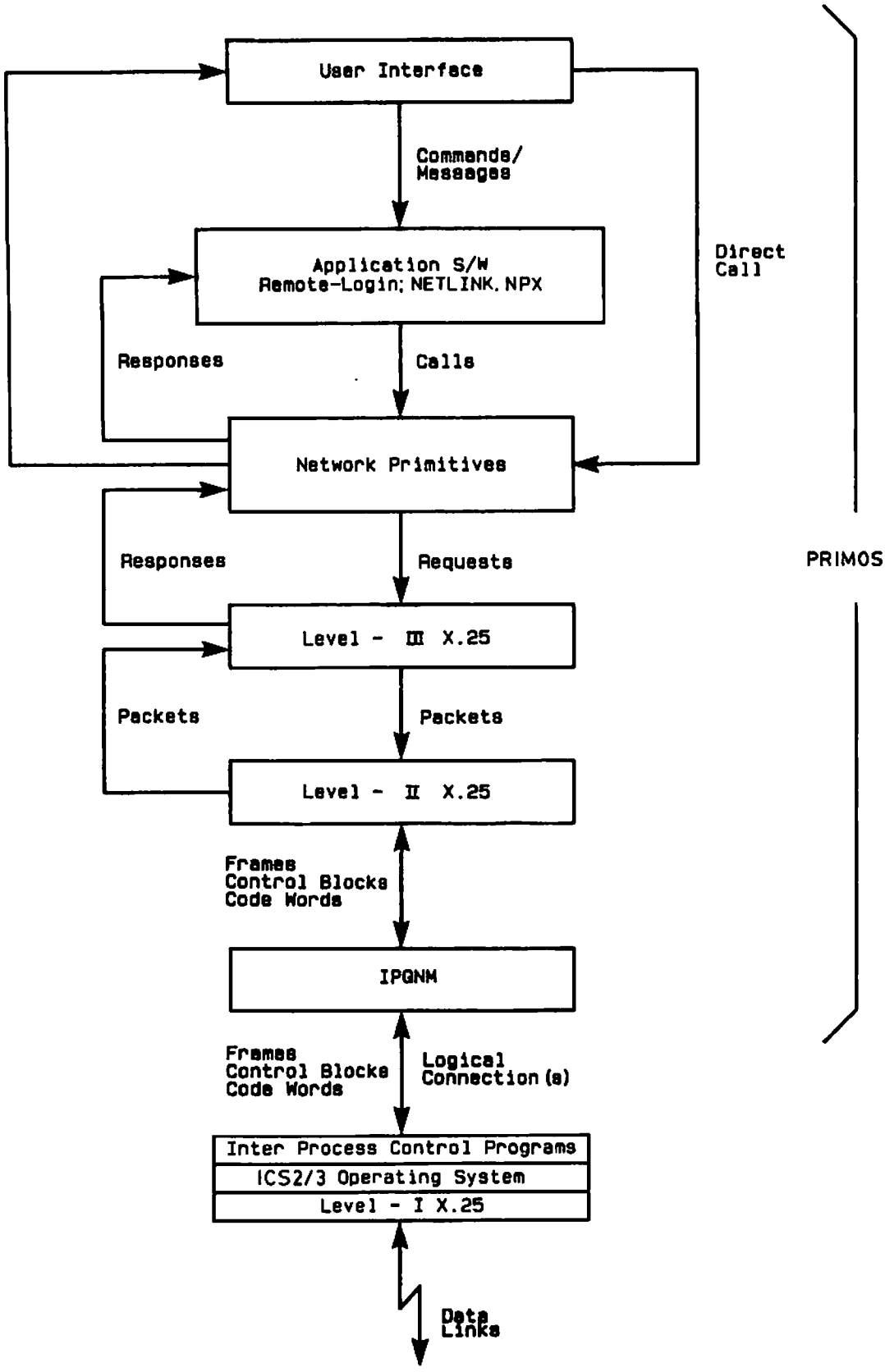
- By direct calls to the network primitives from the user's software (user interface). This access to the network primitives enables the user to write customized communication interfaces.

The two mechanisms are shown in Figure 3-1.

The network primitives request the X.25 software to perform the message packaging and communications control required across the data link. Virtual circuits are established (or disconnected) upon request to provide logical FDX links to processes in other (or the same) machines, thus enabling the transfer of information. Flow control for virtual circuits prevents incoming data packets from saturating the Prime host. This flow control is achieved by the Prime host controlling the rate at which it accepts the packets. There is usually a finite network-dependent limit on the number of data packets which may be in the network at any one time for a given virtual circuit.

Shutdown

Shutdown occurs either when PRIMENET is turned off by using the STOP_NET command or when the PRIMENET process (NETMAN) is logged-out. At shutdown (for a specific host), X.25 support and all lines are brought down, and network resources returned to the system. In addition, all data bases are cleaned up, queues flushed, controllers deconfigured, and all logical connections deleted.



PRIMENET/X.25 Invoking Mechanisms
Figure 3-1

RJE PROTOCOL IN PRIME HOST

Prime's Remote Job Entry (RJE) products allow Prime systems to emulate the standard remote job entry terminals of IBM, ICC, and others. The Prime system can queue the jobs that are sent to run on a remote system, and receive the output from those jobs. These emulator operations involve the software modules detailed below.

Figure 3-2 shows the interaction between the various RJE software modules in the Prime system.

RJQ

Remote Job Queue (RJQ) is a user interface program that allows users of any RJE emulator to queue files for transmission to remote sites. It also enables them to examine and modify the queues of files waiting for transmission.

RJOP

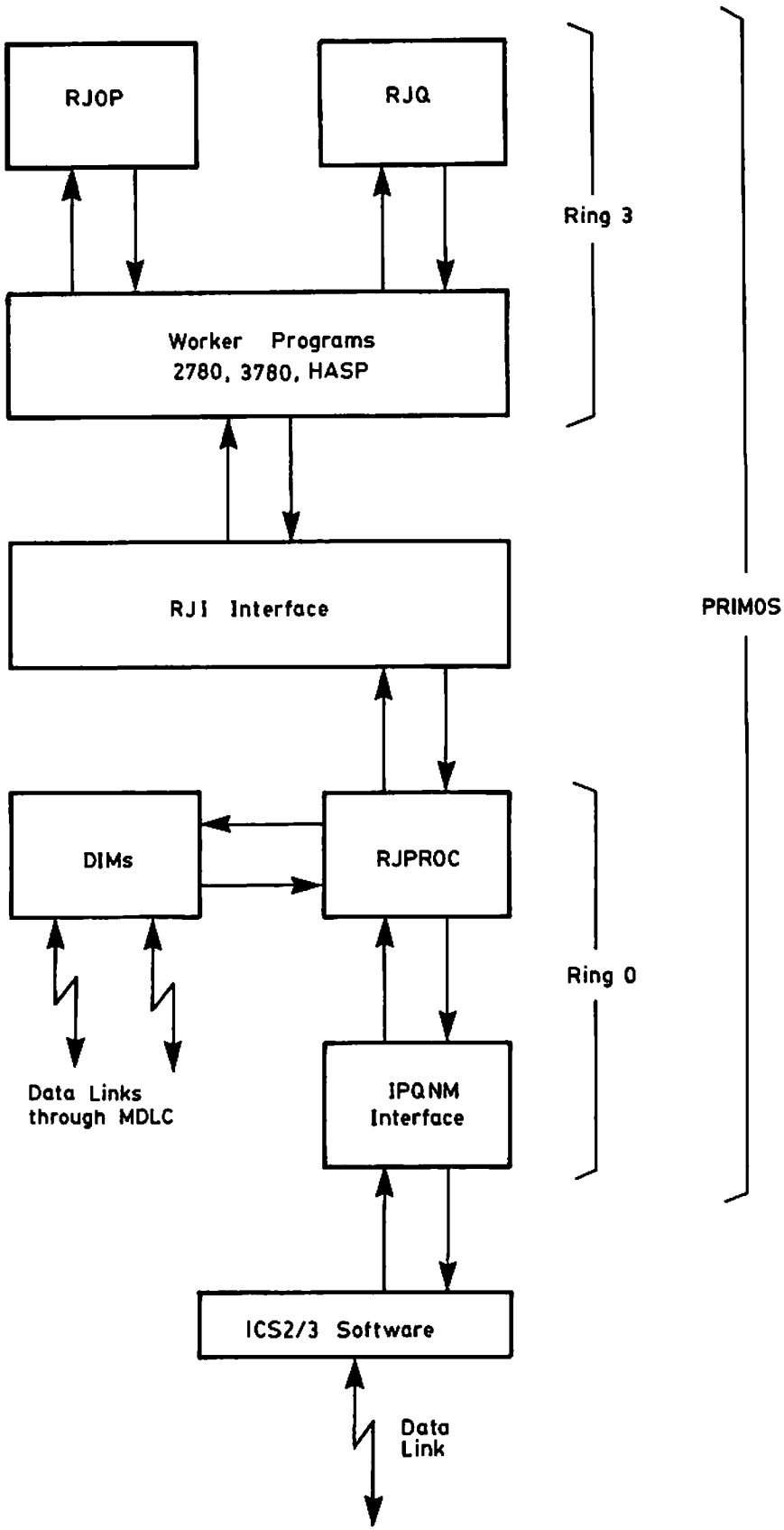
Remote Job Operator Interface (RJOP) is an operator interface program that enables an operator to communicate interactively with the system in order to monitor and control emulators. The RJQ command, with all its functions, is supported by RJOP in a similar manner. RJOP commands obtain the statistical information regarding transactions with an RJE station. RJOP provides an interface for more than one emulator at a time.

The Worker Process

The worker programs perform the necessary code translation and formatting of the files to be transmitted. The worker programs queue the files for the protocol handler, in the form of blocks for transmission to remote stations. Similarly, they perform code translation on the data received from the protocol handler, and then write the translated data, (in the form of a file) to an appropriate place, for instance, a printer spool queue or Punch directory.

RJI

Remote Job Interface (RJI) is a generic term describing an interface between Ring 0 and Ring 3 software.



Prime RJE Software
Figure 3-2

RJPROC

RJPROC is the protocol-handling process that receives data from and transmits data to one or more remote sites, using the appropriate line protocol. It performs the protocol-handling functions of the 2780, 3780, HASP, GRTS, and XBM emulators, in addition to providing the input/output interface for those protocols. RJPROC consists of an event handler, protocol-specific code for each protocol, and the supporting line routines.

On the ICS2/3, RJPROC supports only 2780, 3780 and HASP protocols (on synchronous lines); however, on the MDLC, RJPROC supports all the above protocols.

IPQNM

All the input/output operations using the ICS2/3, are performed by RJPROC through the standard IPQNM interface. This IPQNM interface consists of a set of routines that enable processes within PRIMOS to communicate with processes in the ICS2/3, using queuing and notification methods. (See Chapter 2, ICS2/3 SOFTWARE.)

SDLC SUPPORT FROM THE PRIME HOST

The Prime CPU, using the PRIME/SNA software Server Subsystem, works with the ICS2/3 by performing the following functions:

- Downline loading software into the controller at warm start and cold start time
- Processing requests to start and shut down lines
- Supervising controller activity and event logging

PRIME/SNA Server Subsystem

The Server subsystem is the major software component of PRIME/SNA. It operates as a Ring 3 phantom, communicating with other processes through the IPQNM. The Server subsystem, which operates in both the Prime system and the ICS2/3, enables the Prime system to emulate one or more (up to eight) SNA peripheral nodes. The Server subsystem controls the peripheral nodes and the attached logical units (LUs) to ensure conformity with SNA protocol. The Prime can share a multipoint line with other SNA (PU.T2) peripheral nodes.

The Server is PRIME/SNA's kernel and it performs the following tasks:

- Handles one to four SDLC lines
- Supports one to eight remote system connections
- Supports a peripheral node's logical units (ports)
- Helps operations staff administer PRIME/SNA
- Provides configuration of PRIME/SNA peripheral nodes and LU ports

When the PRIME/SNA Server subsystem is started (see Chapter 5, CONFIGURATION), the Server sends information (for each SDLC line being started) to the ICS2/3, such as

- Synchronous line number
- Signal encoding to be used (NRZ or NRZI)
- Line interface to be used (V.24 or V.35)
- HDX or FDX electrical connection
- Internal or external clock for the modems
- Point-to-point or multipoint operation
- Switched or leased line
- FDX or HDX data mode

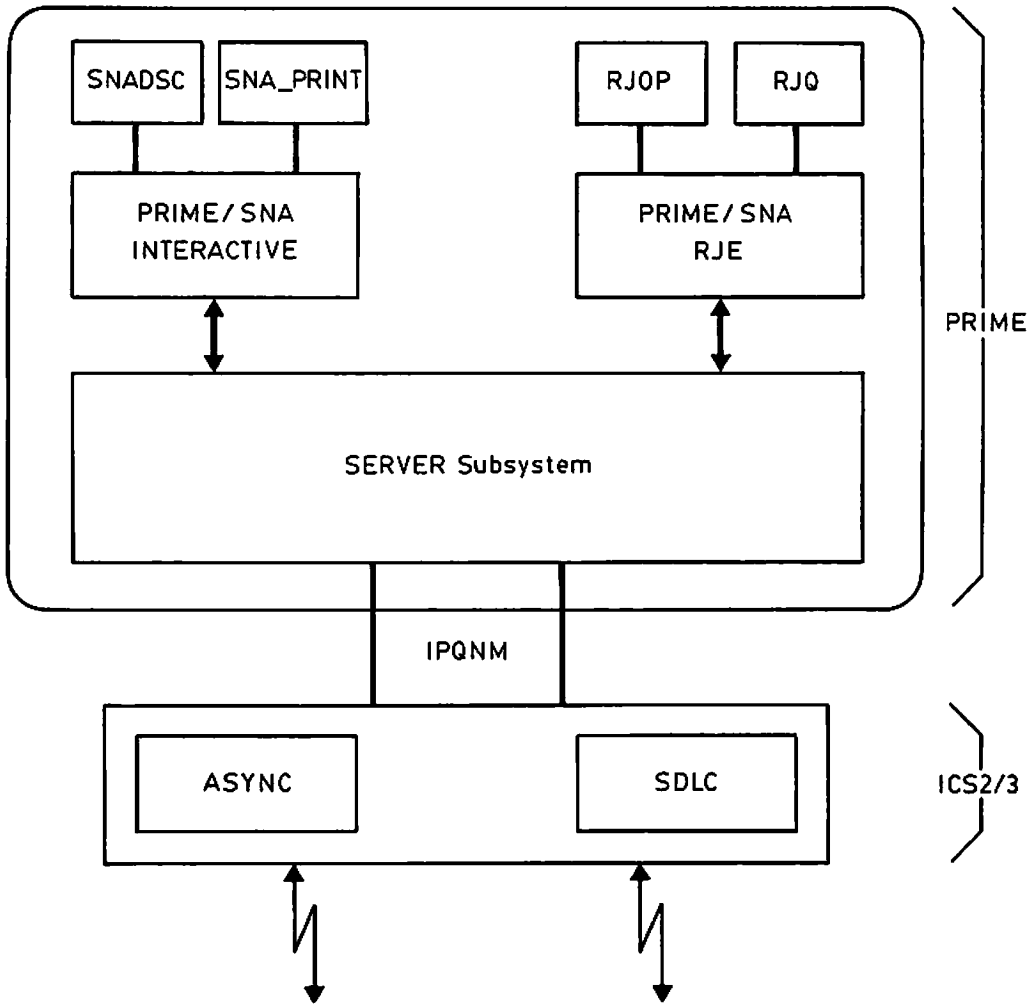
Other PRIME/SNA Components

The Interactive subsystem, using the basic support provided by the Server subsystem, provides support for emulation of IBM 3270-type devices. The SNADSC component of the Interactive subsystem emulates the IBM 3278 display station (terminal) using the Prime PT200 terminal. The SNA_PRINT component of the Interactive subsystem emulates IBM 3287 and 3289 printers.

Once logical connections are established, SNADSC has access to IBM applications such as TSO (Time Sharing Option) and CICS (Customer Information Control System), which allow Prime users to perform interactive tasks.

The RJE subsystem, functioning as an RJE Worker under RJE Phase II, emulates an IBM 3776 RJE workstation.

Configuring PRIME/SNA, from the ICS2/3 point of view, is summarized in Chapter 5, CONFIGURATION.



PRIME/SNA — PRIMOS Functions
Figure 3-3

ASYNCHRONOUS PROTOCOL IN THE PRIME HOST

Asynchronous protocol processing is mainly performed in PRIMOS. A Device Interface Module (DIM), called ASYNDM, handles interrupts and controls the input and output of data and status information. Figure 3-4 shows the components of asynchronous protocol processing in the Prime host. (In most aspects, this applies to the ICS1 also.)

Data Flow Between ICS2/3 and PRIMOS

The interface between the ICS2/3 and the ASYNDM process is described in Chapter 2, ICS2/3 SOFTWARE. Briefly, input data from the ICS2/3 is buffered in two DMQ input buffers (one buffer for the ICS1). Interrupts from the ICS2/3 then cause the ASYNDM process to read the DMQ input buffers. The ASYNDM process transfers the characters into the TFLIOB input ring buffer that belongs to the destination user. The user process then transfers the character from the TFLIOB buffer to the user buffer. For output from PRIMOS to the ICS2/3, the process is reversed. Transmitter enable commands initiate transfer of data from the DMQ output buffers into the ICS2/3 and then on to the user terminal, printer, or modem.

ASYNDM Process

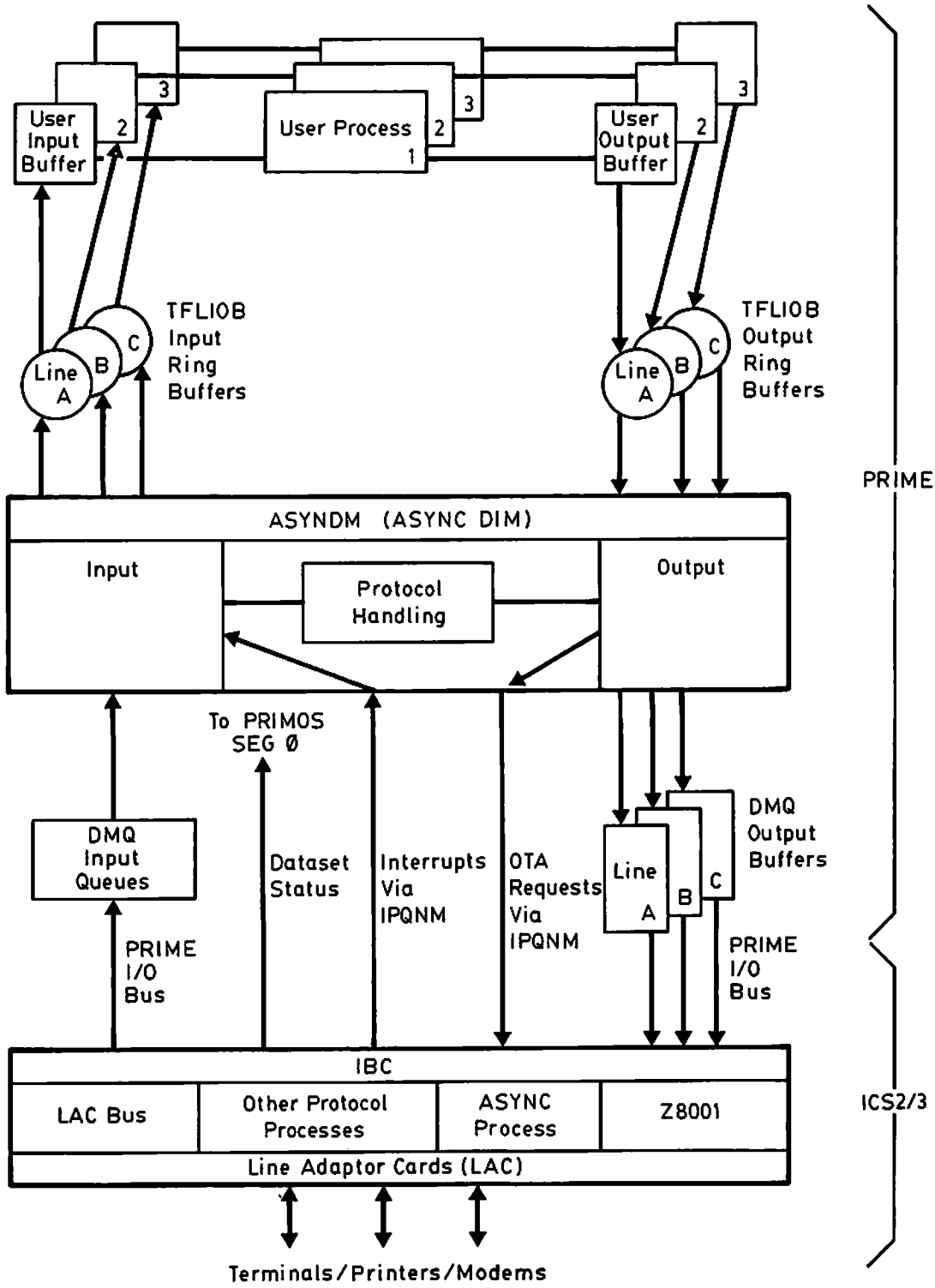
The ASYNDM can be divided into the input and output processes. The input process reads the sixteen-bit word from the DMQ input buffers. This read is triggered by ICS interrupts that are passed to ASYNDM through the IPQNM. The sixteen-bit word contains the character in its right half. The left half contains the line number, parity bit, and flags that indicate if the character in the right half of the word is a special control character.

ASYNDM then decodes the relevant line number from the left half of the sixteen-bit word, applies the rules of the protocol handler to each character, and transfers the character to the appropriate TFLIOB input buffer. Two characters are stored in each TFLIOB sixteen-bit word.

The protocol handlers referred to here are those specified in the AMLC command directive: TTY, TT8BIT, TRAN, TTYUPC, TTYNOP, and ASD. (See the System Administrator's Guide.) The protocol may cause the character to be simply passed along, echoed, changed, discarded, or to perform a special control function, depending on the protocol in use for that line.

The output process moves characters from the TFLIOB output buffers, applies the protocol handler rules, and then passes the character to the appropriate DMQ output buffer.

The TFLIOB buffers have two pointers; one indicates where the next received character will be stored, the other, where the next character will be read from. When the pointers are equal, the buffer is empty.



Prime Host Asynchronous Processing
Figure 3-4

The AMLBUF command directive is used to adjust TFLIOB input and output buffers and the DMQ output buffers for each specified line. The DMQ input buffer size (ICS2/3 to PRIMOS), is varied using the ICS INPQSZ command directive. (See Chapter 5, CONFIGURATION.)

ICS ASYNDM Code Flow

Figure 3-5 shows the basic ICS asynchronous control flow that operates as follows:

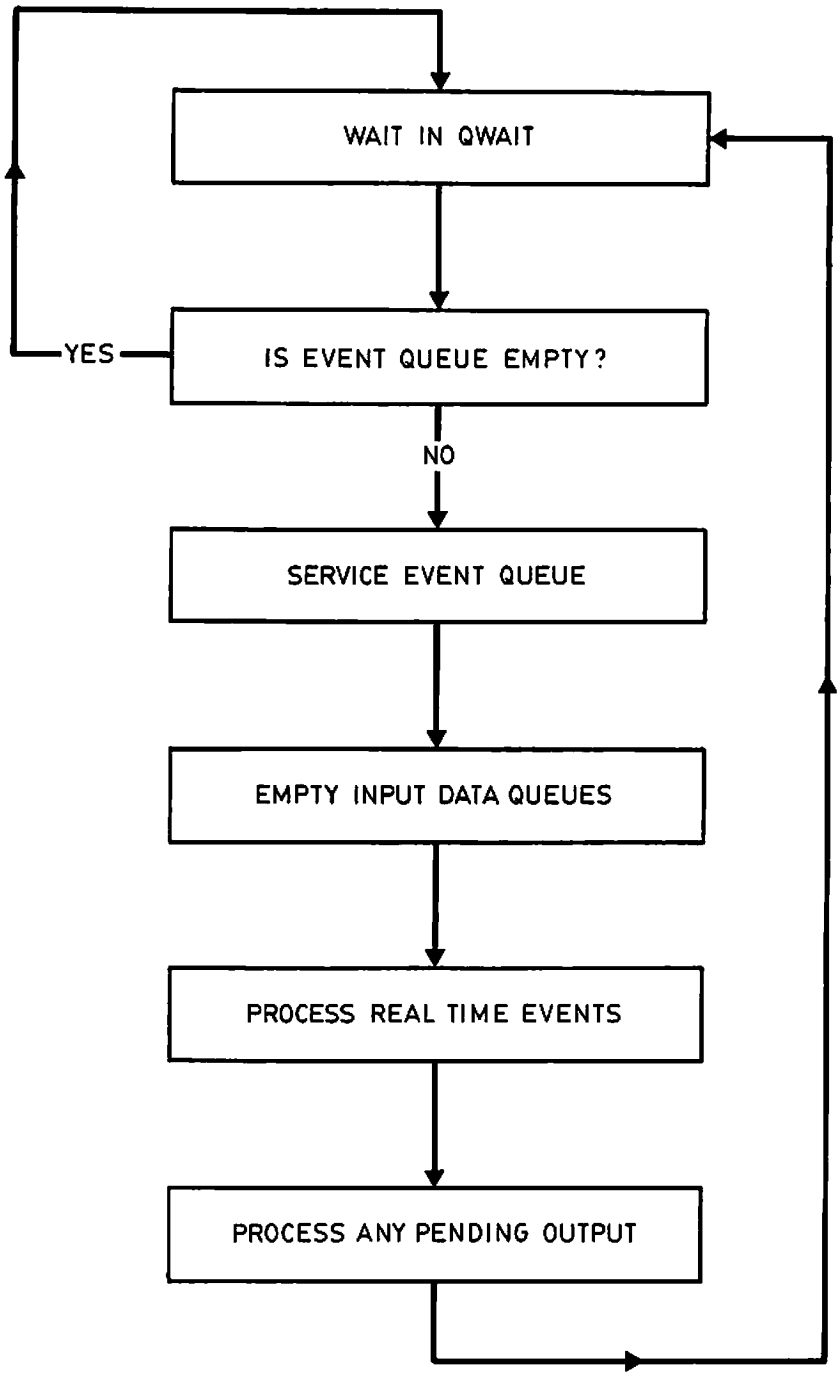
Wait in QWAIT: The DIM waits in the IPQNM routine QWAIT. When DIM returns from QWAIT, it will have the Logical Connection ID (LCID) code it removed from the event queue.

Service Event Queue: The LCID code points to the input queue that has data (characters or status). The queue is flushed.

Empty Input Data Queue: Data is read from input data queues (one queue per ICS1 and two queues per "virtual" ICS1 in ICS2/3s). Each queue is flushed in turn.

Process Real Time Events: The carrier (dataset status signal CD) is monitored such that, when it drops, the line is logically disconnected and, possibly, the process is force-logged out. (This is set up by the DISLOG and AMLTIM command directives.) The disable-transmitter facility disables the transmitter of lines whose output queues are empty. When data appears on the output queue or when characters are seen on the line's TFLIOB buffer, the transmitter is enabled. The disable and enable are done via IPQNM commands.

Process Any Pending Output: PRIMOS buffers for each line are read and data queued using the same servicing policy as for input.



ICS Asynchronous DIM Flow Control
Figure 3-5

4

ICS2/3 System Architecture

INTRODUCTION

ICS2/3 — Prime system architecture supports software communications for the following:

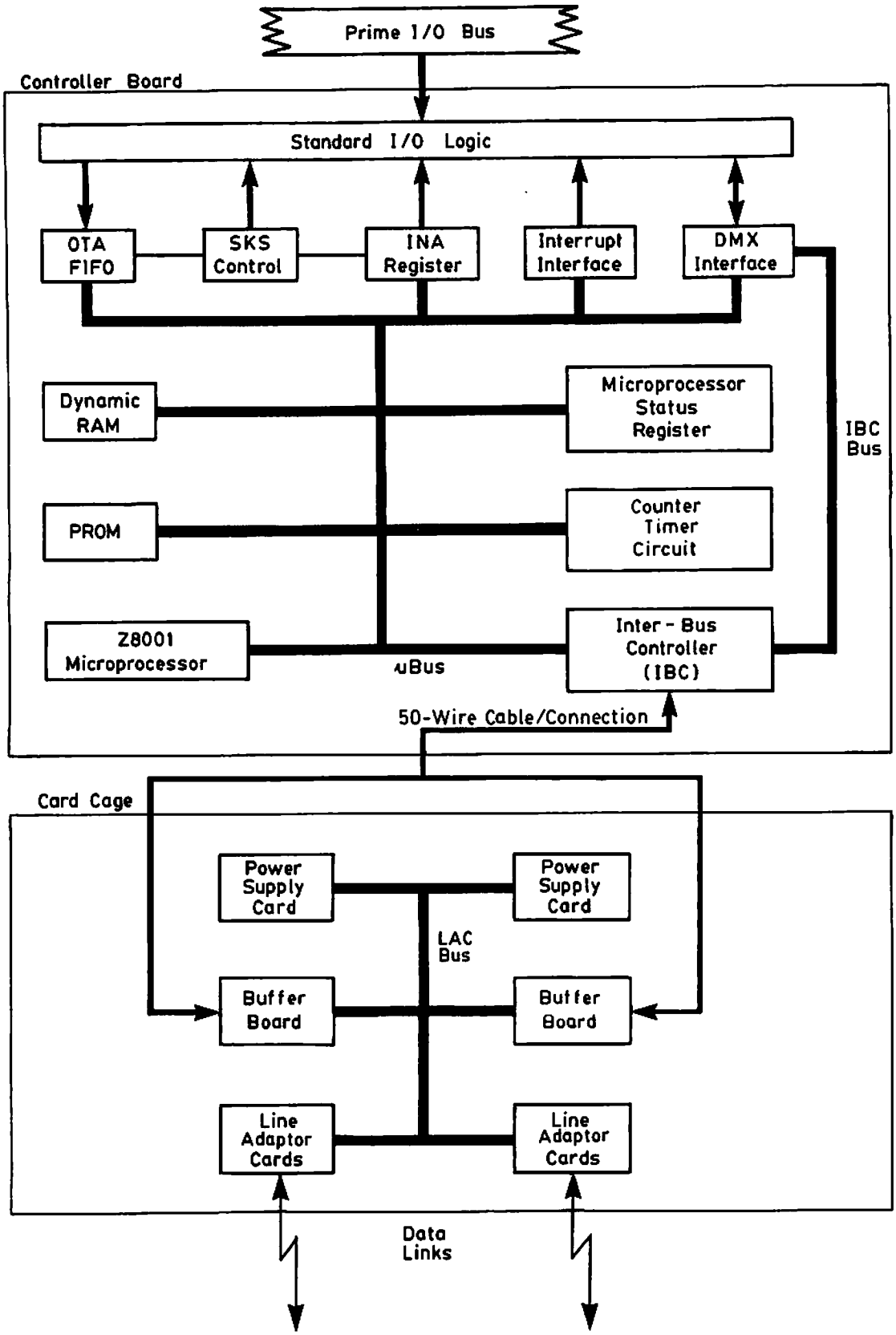
- PRIMENET/X.25, HDLC, and bisync-framed protocols
- Remote Job Entry (RJE) bisync-based protocol environments for IBM 2780/3780 and HASP
- SDLC support for SNA
- Asynchronous device support

This chapter describes the system architecture relevant to each protocol.

ICS2/3 HARDWARE ARCHITECTURE

The ICS2/3 hardware consists of a controller board, which is housed in the Prime host computer; a LAC card cage assembly housing the LACs, buffer board(s), and power supplies (where necessary); and a 50-wire cable that links the controller board to the LAC card cage assembly.

The ICS2/3 hardware architecture, that is, the data communications connection and control within the ICS2/3 boundaries, is shown in Figure 4-1, and is fully described in Chapter 1, HARDWARE COMPONENTS.



ICS2/3 Hardware Architecture
Figure 4-1

PRIMENET/X.25 SYSTEM ARCHITECTURE

PRIMENET/X.25 is constructed of several software layers. It is based on the International Standards Organization's Open Systems Interconnection (ISO OSI) model. This enables Prime 50 Series computers to communicate with other systems that support X.25 protocols. Read the following description of the PRIMENET/X.25 layers in conjunction with Figure 4-2, the data flow diagram.

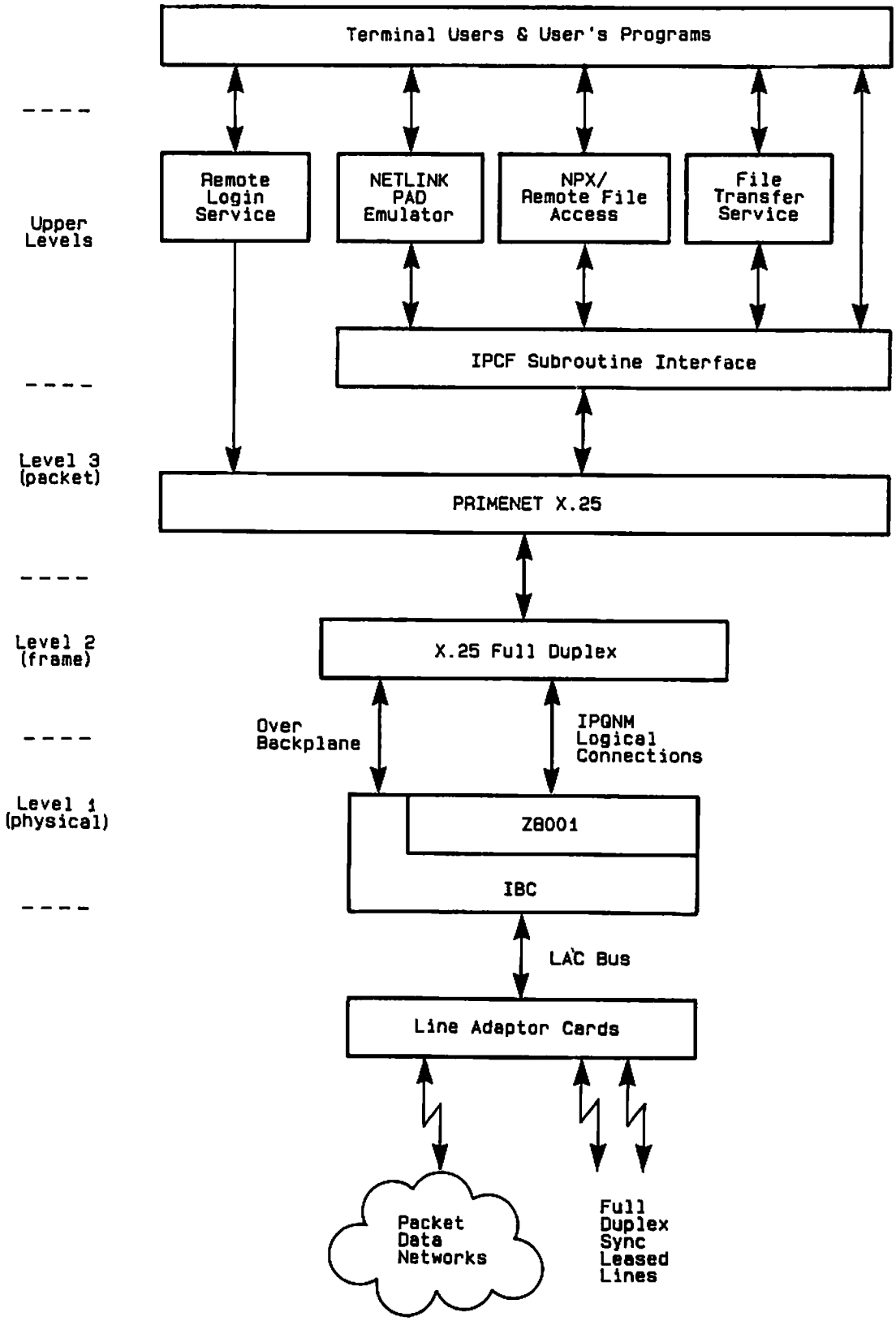
Level 3: The packet interface layer creates and controls the virtual circuits across the network, handles error recovery, and controls the flow of information. Level 3's X.25 support provides a standard interface to upper-level software, irrespective of what kinds of links comprise levels 1 and 2. This corresponds to the ISO OSI network layer.

Level 2: The link protocol layer, which corresponds to the ISO OSI's data link layer, defines a protocol to which two linked nodes must adhere when they transfer information from one to the other. The Level 2 layer handles node-to-node integrity and transmission of the Level 3 data by encapsulating the data in frames of a specified format.

Level 1: The physical layer is the hardware interface and also the layer that is contained in the ICS2/3. This layer acts as the intermediary between the physical transmission medium and the rest of PRIMENET and the system.

Upper Levels: The upper layers provide the user services (levels 4-7 in the ISO model). These user facilities utilize the lower layers to perform actions specified by user terminal inputs.

For further information see the PRIMENET Guide and any accompanying update packages.



PRIMENET/X.25 Architecture
Figure 4-2

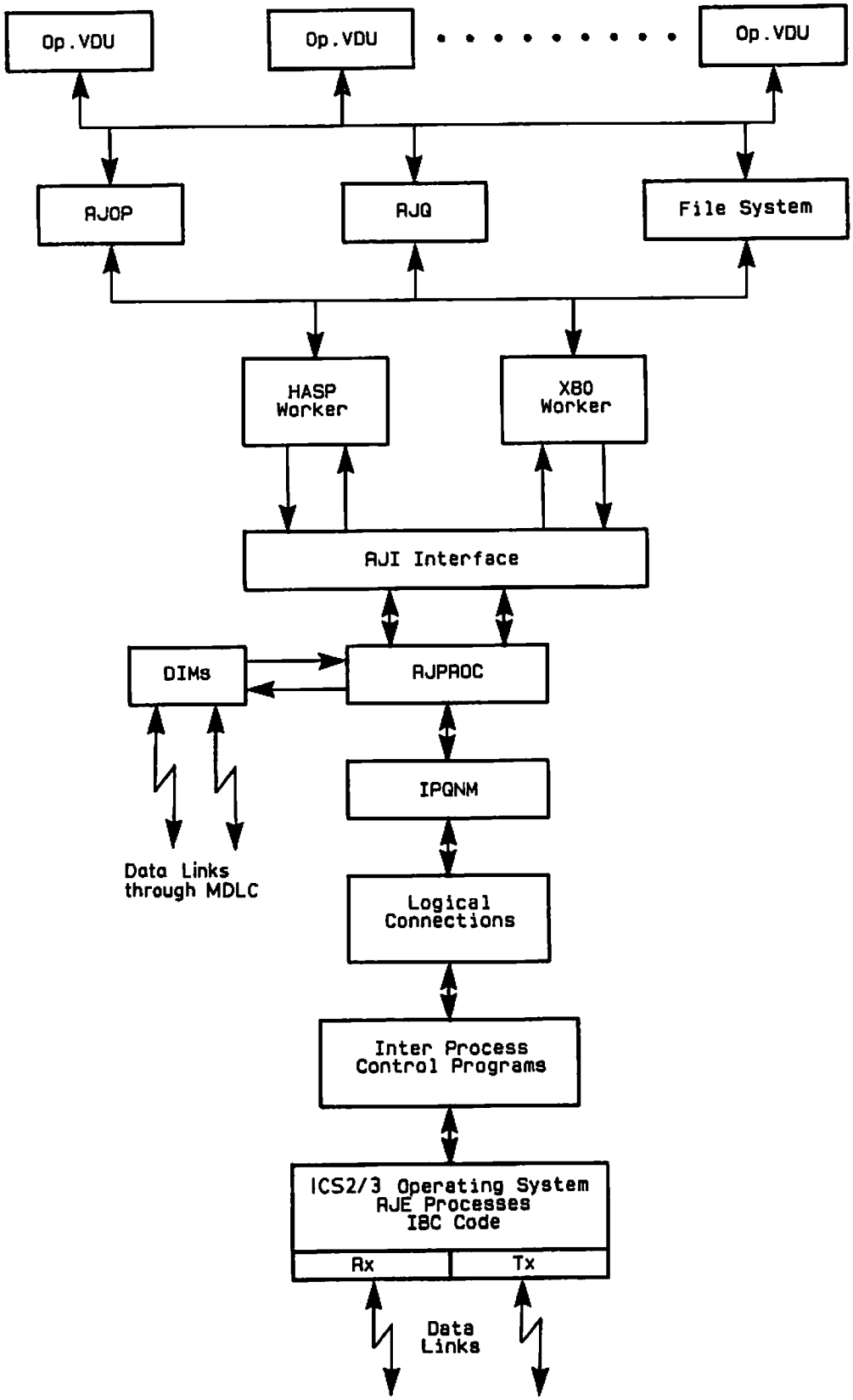
RJE SYSTEM ARCHITECTURE

When a Prime 50 Series computer emulating an RJE workstation (an RJE site) is to be linked to a central site or to another Prime 50 Series computer, a line is assigned and a Site Definition File (SDF) is created. There is one SDF for each RJE site, and this contains all the characteristics for that particular site. Typically, the SDF contains the following in addition to other site definition commands:

- Site name
- Line speed (BAUD command)
- Line translation code (CODE command)
- Data set types (DATASET command)
- Logical line number (CONNECT command)

The format of the commands and the details required to create the SDF can be found in the Remote Job Entry Phase II Guide.

Figure 4-3 shows the data path between the terminal operator of the Prime 50 Series computer and the links to the control RJE site.



ICS2/3-Prime RJE System Architecture
Figure 4-3

SDLC SYSTEM ARCHITECTURE

SDLC processing is enabled by the SYNC CNTRLR directive when its protocol-token includes SDLC. This causes the PRIMOS operating system to load into the ICS2/3 memory a downline load file containing SDLC protocol handling capability.

Within the ICS2/3, SDLC processing is performed mainly by the SDLMAN process, aided by the SDLMTR process acting as its interrupt processor and providing an interface to the terminal side of the ICS2/3.

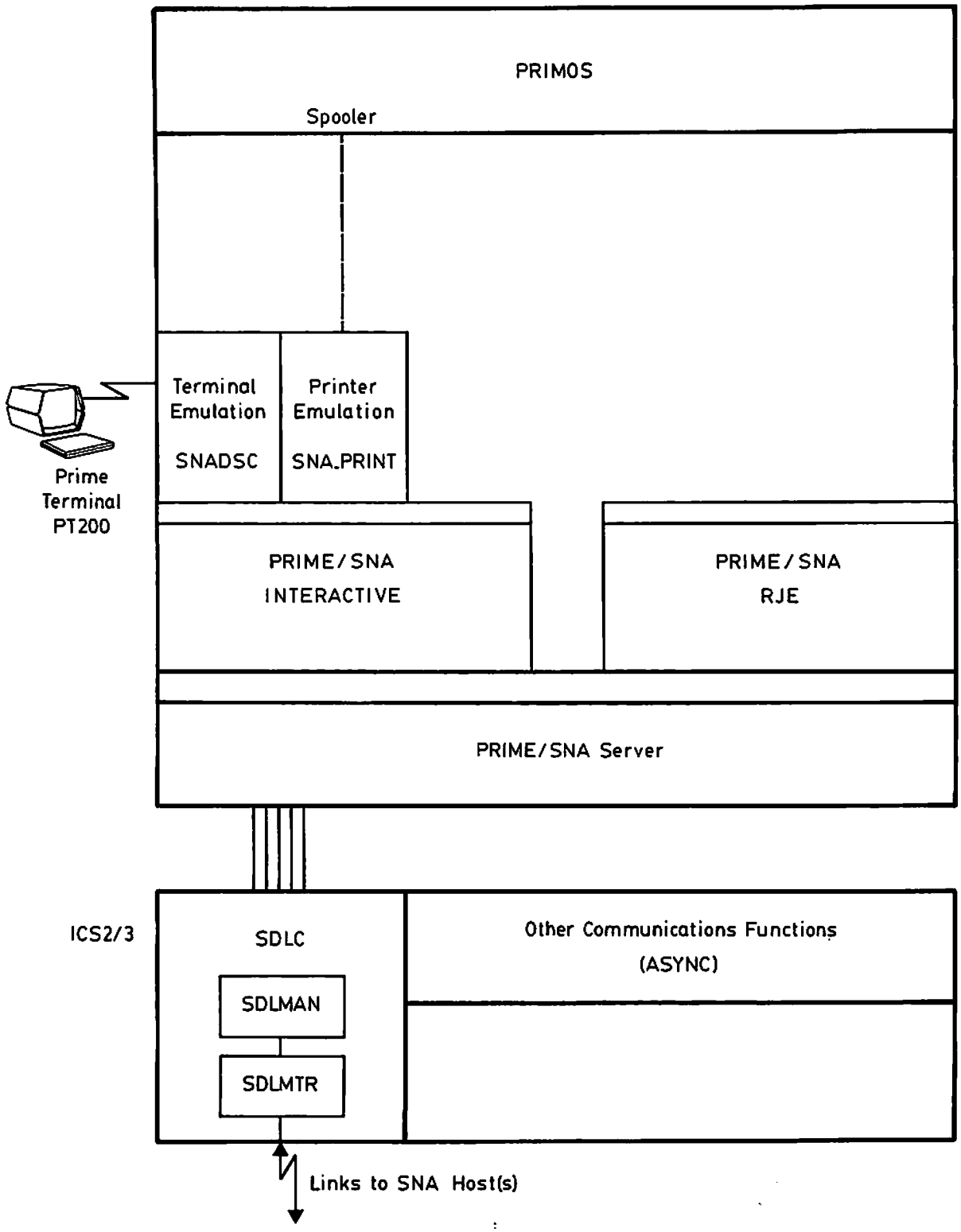
When the SNA_SERVER -START command is given, the Server subsystem takes information from the SNA configuration file and sends it to the ICS2/3. This information contains configuration details for each line and remote system in the Prime SNA network being started.

Similarly, the SNA_3270 -START command initializes the Interactive subsystem. The SNADSC process and microcode in the PT200 terminal perform terminal emulation, while SNA_PRINT performs printer emulation.

SNA RJE support is initialized when the RJE operator issues the -TOSITE RJOP subcommand and specifies a Site Definition File which has been defined for SNA RJE.

For a full description see the PRIME/SNA Administrator's Guide.

Figure 4-4 illustrates PRIME/SNA system architecture.



PRIME/SNA (SDLC) System Architecture
Figure 4-4

ASYNCHRONOUS SYSTEM ARCHITECTURE

The basic asynchronous system architecture is shown in Figure 4-5. Asynchronous processing is done mainly in PRIMOS, so the Prime host software diagram is used here.

At cold start, configuration information is taken from the CONFIG file and used to initialize all asynchronous lines. This includes setting up PRIMOS-to-ICS and ICS-to-PRIMOS data queues, Queue Control Blocks (QCBs), and addresses of the dataset status words.

The AMLC directive provides the configuration parameters for each asynchronous line by specifying such things as protocol, line number, dataset control, line looping, line speed, protocol, reverse flow control, number of stop bits, parity, and character length. The Lword within the AMLC directive specifies half or full-duplex, echo LINE FEED for RETURN, XON/XOFF flow control, XOFF seen, data set sense, error detection enable, and user number for that line.

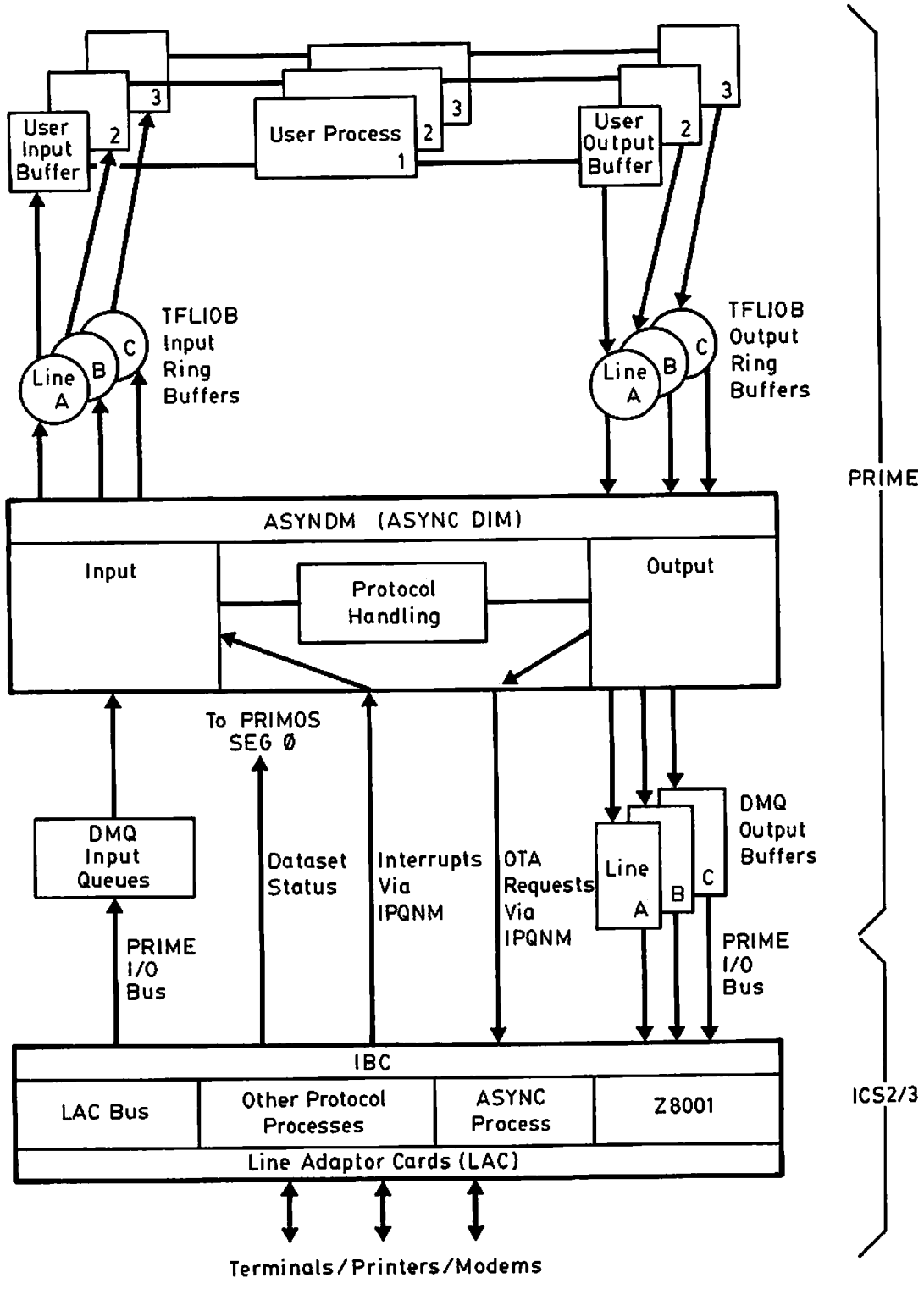
The protocol specified determines what the asynchronous DIM (ASYNDM) does to each character as it passes through, going to or from the ICS controller.

If reverse flow control is enabled, the ICS controller is able to send XOFF to a terminal that is about to overflow the DMQ input queues. The controller will send XON to the terminal when the queues are no longer full, allowing transmission to continue.

DMQ input queues use a pair of queues for each "virtual" ICS1, whereas DMQ output buffers are allocated for each line.

The ICS periodic interrupt drives the ASYNDM process, causing it to "wake up" and process both input and output data until all queues and buffers are flushed.

For more detail see either Chapter 2, ICS2/3 SOFTWARE, or Chapter 3, PRIME HOST SOFTWARE.



Asynchronous System Architecture
Figure 4-5

5
Configuration

INTRODUCTION

This chapter describes the possible configurations of the hardware and software components of the ICS2/3.

HARDWARE CONFIGURATION

The hardware configuration options for the ICS2/3 are applicable to the following units:

- Controller board
- Controller board to buffer board cabling
- Buffer board
- LAC bus backplane
- LAC bus power supply
- LAC card cage assembly

Controller Board

The controller board can be configured with different backplane addresses. The device address is the I/O bus address that the controller board responds to during programmed I/O (PIO) instructions from the host CPU. The device is also referenced in several error messages.

If there is only one ICS2/3 controller board fitted in the chassis, the device address of that controller board is usually configured to either '10 or '11. However, a system with two or more controller boards fitted must be configured with different device addresses for each. The address configuration may only be carried out by a Prime field engineer.

Buffer Board Cabling

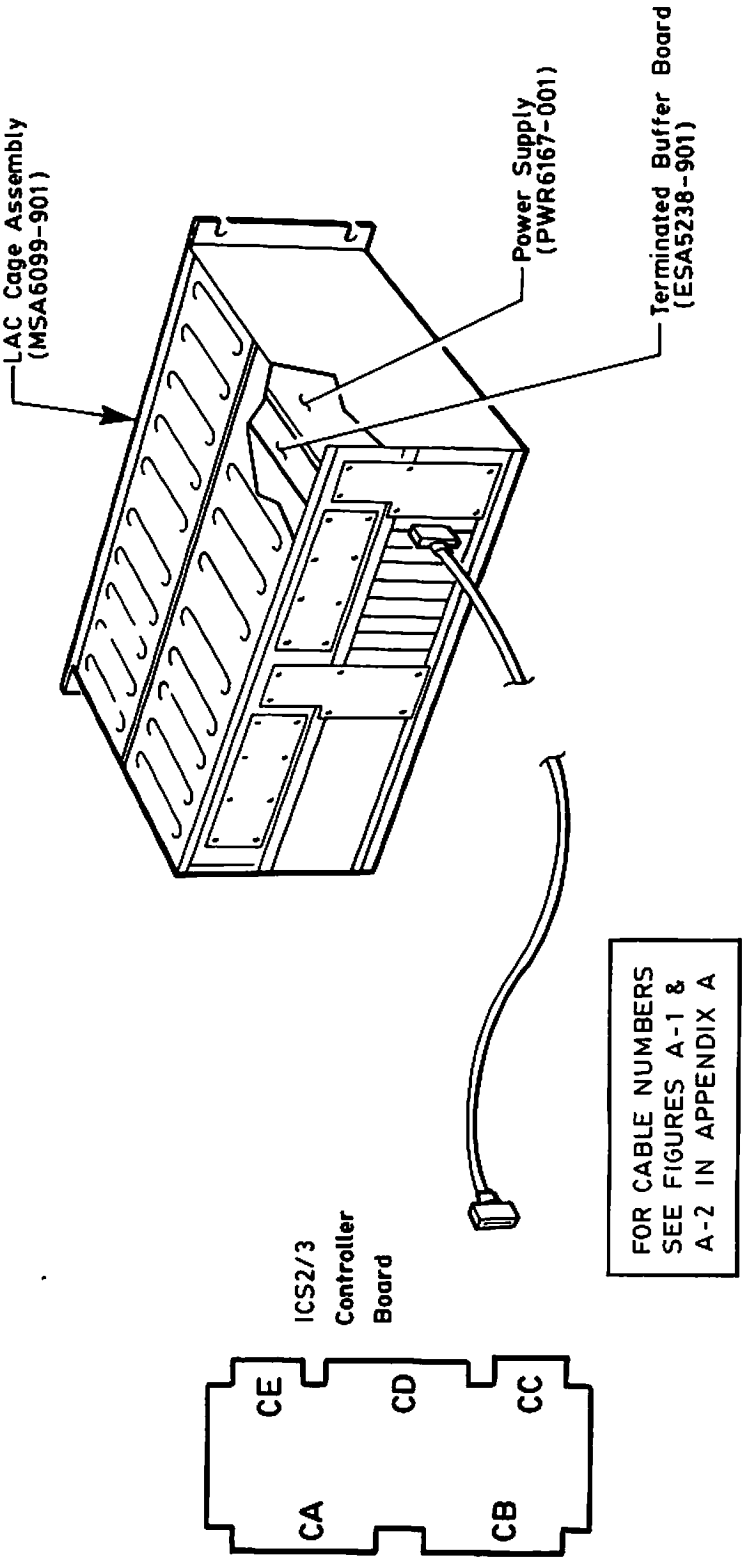
On the 8-LAC ICS3 card cage, the controller board to buffer board cable connection is inside the CPU cabinet. On larger systems this cabling may be configured with or without a cable bulkhead. Systems that do not require a cable bulkhead use a single 50-wire cable to make the connection. This 50-wire cable is available in either a single buffer board or double buffer board version.

The single buffer board cable has two connectors and connects one controller board with a single buffer board. The double buffer board cable is a daisy-chained cable with three connectors, used for connecting two buffer boards to a single controller board.

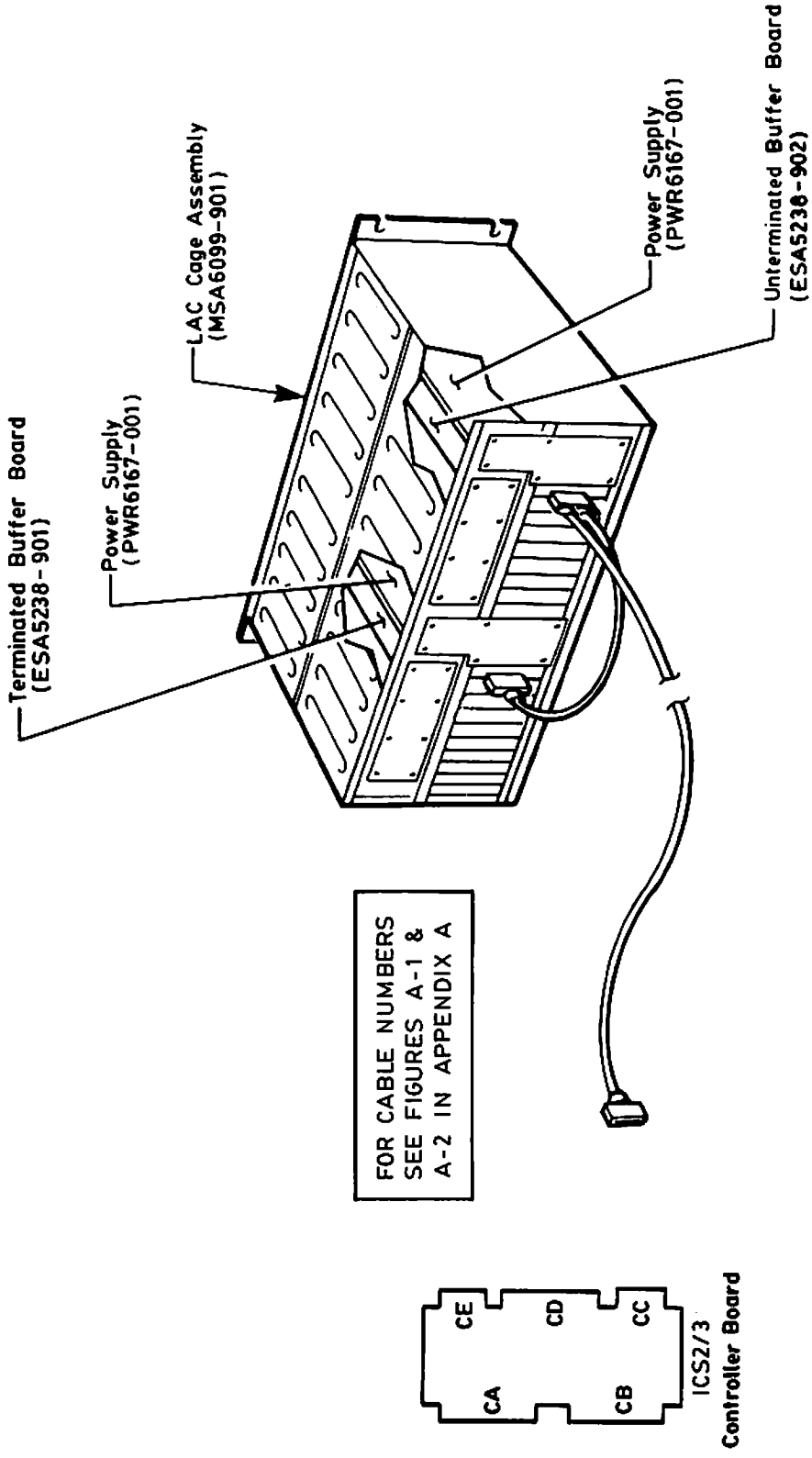
Connection of the 50-wire cable is as follows:

- Single buffer board: connector C1 of the cable is connected to edge connector CC of the controller board, and connector C2 of the cable is connected to edge connector CB of the buffer board (Figures 5-1, and A-1, A-2, and A-4 in Appendix A, ICS2/3 CABLES AND CONNECTORS).

- Double buffer board: connector C1 of the cable is connected to edge connector CC of the controller board, connector C2 of the cable is connected to edge connector CB of one buffer board, and connector C3 is connected to edge connector CB of the second buffer board (Figures 5-2, and A-1, A-2, and A-4 in Appendix A, ICS2/3 CABLES AND CONNECTORS).



ICS2 Card Cage, Single-complement Configuration
Figure 5-1



ICS2 Card Cage, Double-complement Configuration
Figure 5-2

Systems that use a cable bulkhead require three cables (see Figure A-2). Note that there is a choice between two cables to connect from the peripheral bulkhead to the LAC card cage assembly.

- Cable CBL6268-901 connects edge connector CC of the controller board to the system bulkhead.
- Cable CBL6796-XXX connects the system bulkhead to the peripheral bulkhead.
- Cable CBL6268-901 connects the peripheral bulkhead to a single buffer board in the LAC card cage assembly.
- Cable CBL6267-901 connects the peripheral bulkhead to two buffer boards in the LAC card cage assembly (double buffer board configuration).

A single buffer board cable is used for those configurations that have one buffer board connected to a single controller board. A double buffer board cable is used for configurations where two buffer boards are connected to one controller board. (See Figures A-1, A-2, and A-4 in Appendix A, ICS2/3 CABLES AND CONNECTORS.)

Prime field engineers ensure that the above cabling is correctly fitted during your ICS2/3 installation.

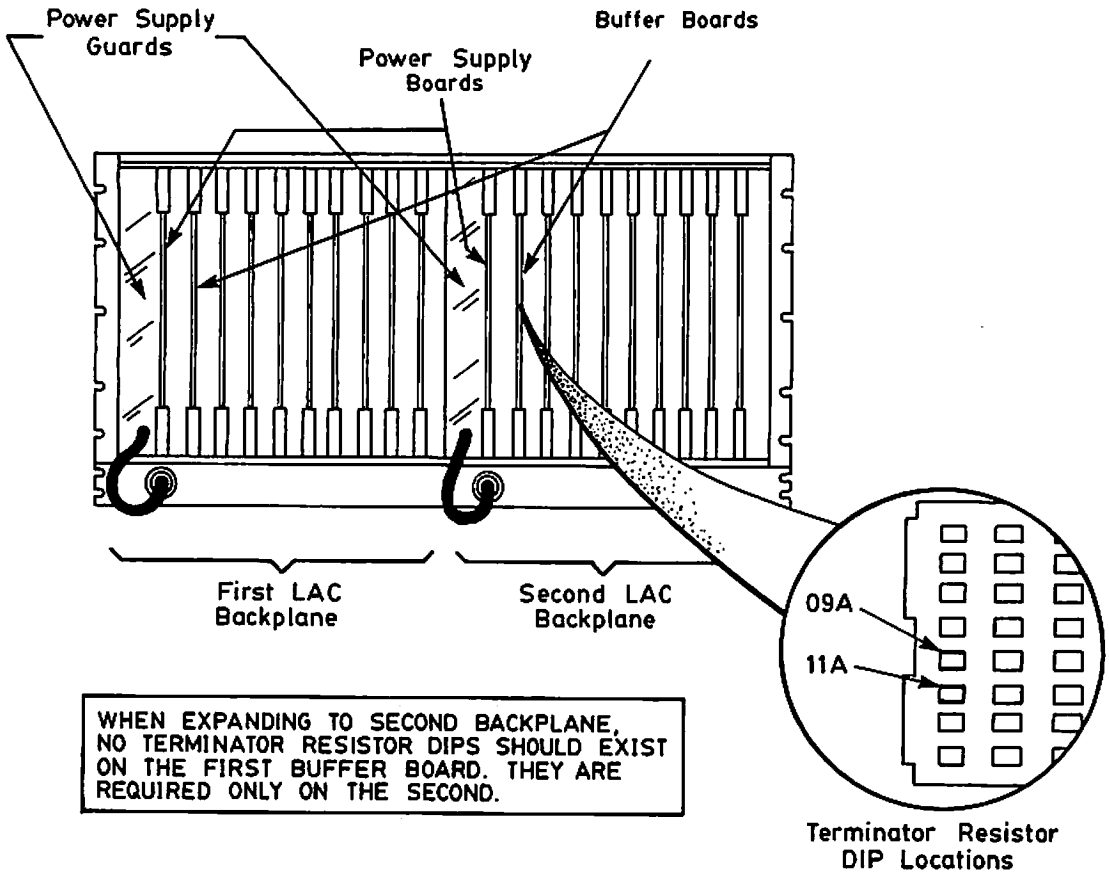
Buffer Board

The buffer board is available in two sizes: 10 by 8 inches for the 16-LAC ICS2 card cage, and 6.5 by 5 inches for the 8 and 16-LAC ICS3 card cage. The buffer board is available in two types:

- Terminated, which has two cable-terminating resister dips at board locations 09A and 11A. (See Figure 5-3.)
- Non-terminated, which is not fitted with resistors at dip locations 09A and 11A.

A terminated buffer board must be used if the configuration is for a single buffer board. However, a two buffer board configuration must have both a terminated and non-terminated buffer board, with the terminated buffer board being the second connected to the controller-to-buffer board cable (Figure 5-2). Prime field engineers will ensure that correct buffer boards are installed.

> **WARNING**
>
> Do not interchange buffer boards on your system; incompatible terminations may cause serious damage. Only Prime field engineers may perform hardware reconfigurations.



ICS2 Card Cage — Front View Showing Buffer Boards
Figure 5-3

LAC Bus Backplane

There are two possible configurations for the LAC Bus Backplane:

- Single-complement: with one buffer board, from one to eight LACs, and, for the ICS2 card cage only, one power supply.
- Double-complement: with two buffer boards and between nine and sixteen LACs (assuming the first half of the LAC card cage had eight LACs fitted). The ICS2 card cage has two power supplies, whereas the 16-LAC ICS3 card cage has one common power supply.

The single-complement configuration is not fitted with a jumper wire on the backplane, and the slot addresses for the eight LACs are 0-7.

The double-complement configuration is fitted with a jumper wire on the backplane. The jumper wire causes the state of an address signal on the backplane to be hardwired low. The LAC slot addresses in this configuration are 0-7 and 8-F (Hex notation).

LAC Bus Power Supply

The LAC bus power supplies contains internal straps used to configure for different input voltages appropriate to the country of use. These straps are not accessible to users.

The number of power supplies fitted depends on the configuration:

- One power supply board is fitted to power a ICS2 LAC card cage single-complement configuration.
- Two power supply boards are fitted to power an ICS2 LAC card cage double-complement configuration.
- One (different) power supply module is fitted to the 16-LAC ICS3 card cage to power up to sixteen LACs.

LAC Card Cage Assembly

The LAC card cage assembly is designed to enable flexibility in configuration. LAC card cages are available in three forms:

- 8-LAC ICS3 card cage (Figure 5-4)
- 16-LAC ICS3 card cage (Figure 5-5)
- 16-LAC ICS2 card cage (Figure 5-3)

The 8-LAC ICS3 card cage is designed for small Prime systems where it will fit inside the CPU cabinet and obtain its power from the CPU.

The 16-LAC ICS3 card cage is for larger Prime systems and contains up to sixteen LACs, one or two buffer boards, and a single power supply for all LACs. It is designed to fit in some Prime CPU cabinets or in a peripheral cabinet.

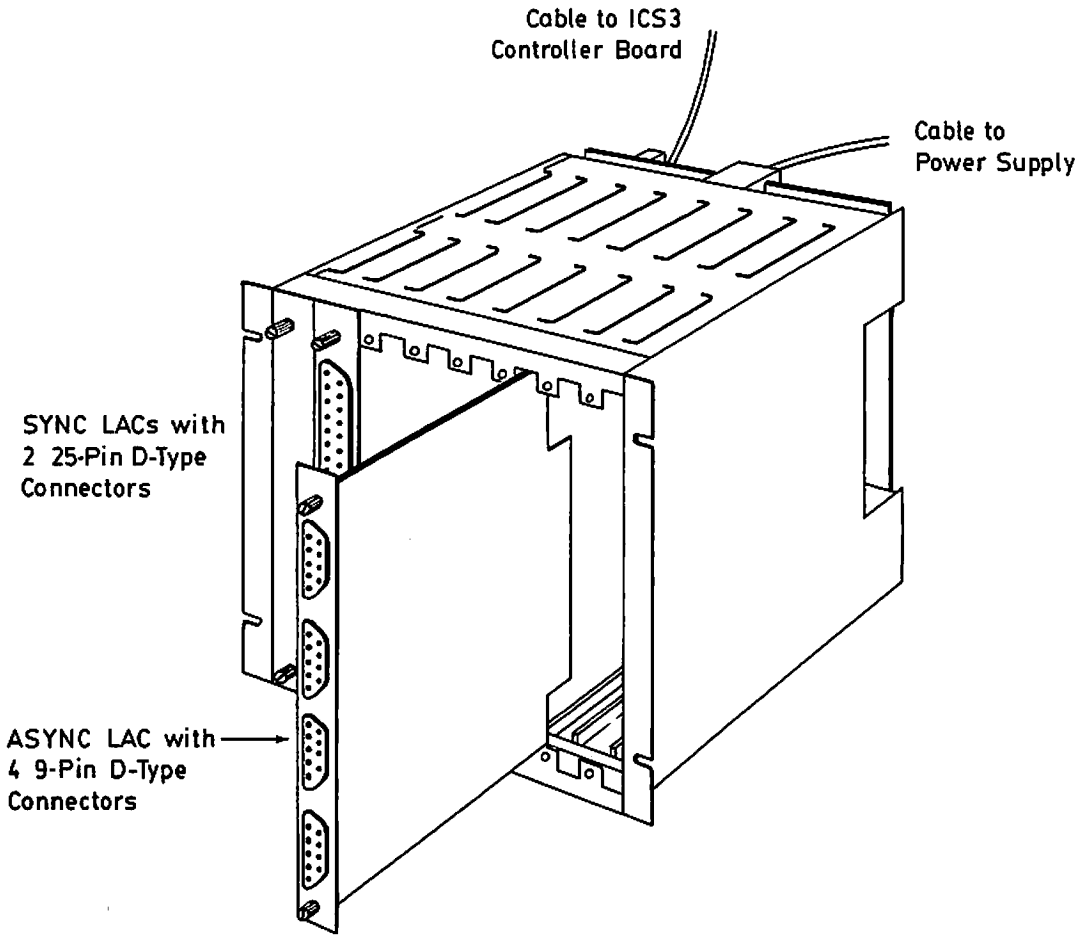
The ICS2 card cage is also for larger Prime systems and contains up to sixteen LACs, one or two buffer boards, and one or two power supply boards. It is designed for installation in a peripheral cabinet.

Both the 16-LAC card cages may be configured in three basic ways:

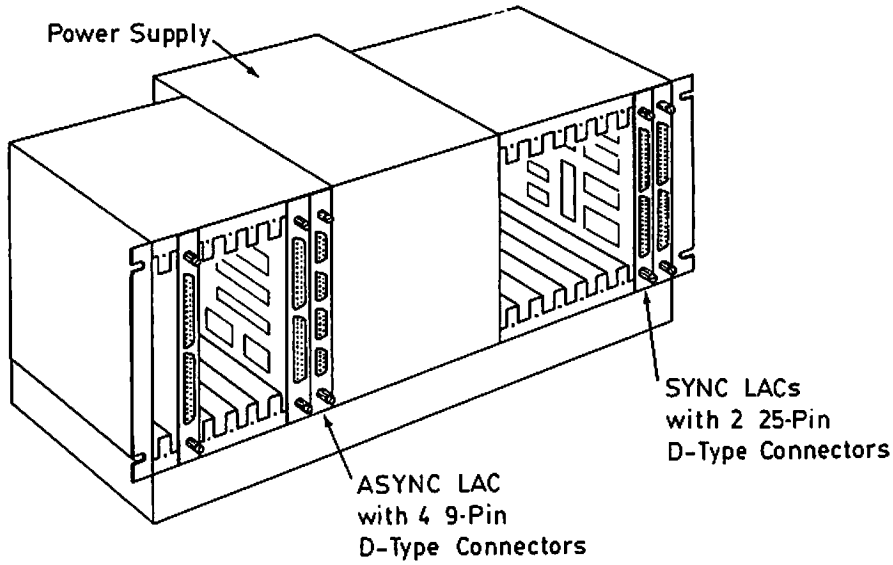
- Single-complement (small system) configuration: this is comprised of one LAC power supply, one buffer board, and from one to eight LACs. The single-complement configuration used a terminated type buffer board, and the single version cable connects the controller board to the buffer board. (See Figure 5-1 for ICS2, and Figure 5-6 for ICS3.)

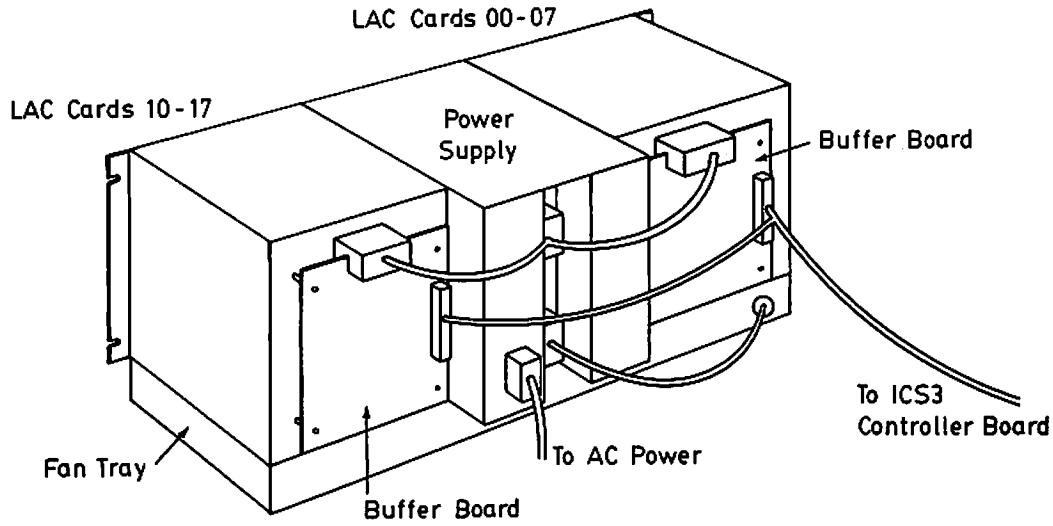
- Double-complement configuration: this configuration has two LAC power supply boards in the ICS2 and a single power supply module in the ICS3, two buffer boards, and between nine and sixteen LACs (assuming that the first half of the LAC card cage is full). Note that, in this configuration, the daisy-chained cable connects the buffer boards. Moreover, the first buffer board on the cable must be non-terminated, and the second buffer board must be terminated. It is not necessary for the first half of the LAC card cage to be full before LACs are added to the second half. (See Figure 5-2 for ICS2, and Figure 5-7 for ICS3.)

- Dual single-complement configuration: this consists of two single-complement configurations in one LAC card cage assembly. This configuration uses two, single version, controller board-to-buffer board cables. The two controller boards can be plugged into the same or different CPUs; in the latter case, both buffer boards are terminated. This particular configuration may not be available in all countries. Contact your local field service engineer for details. (See Figure 5-8 for ICS2, and Figure 5-9 for ICS3.)

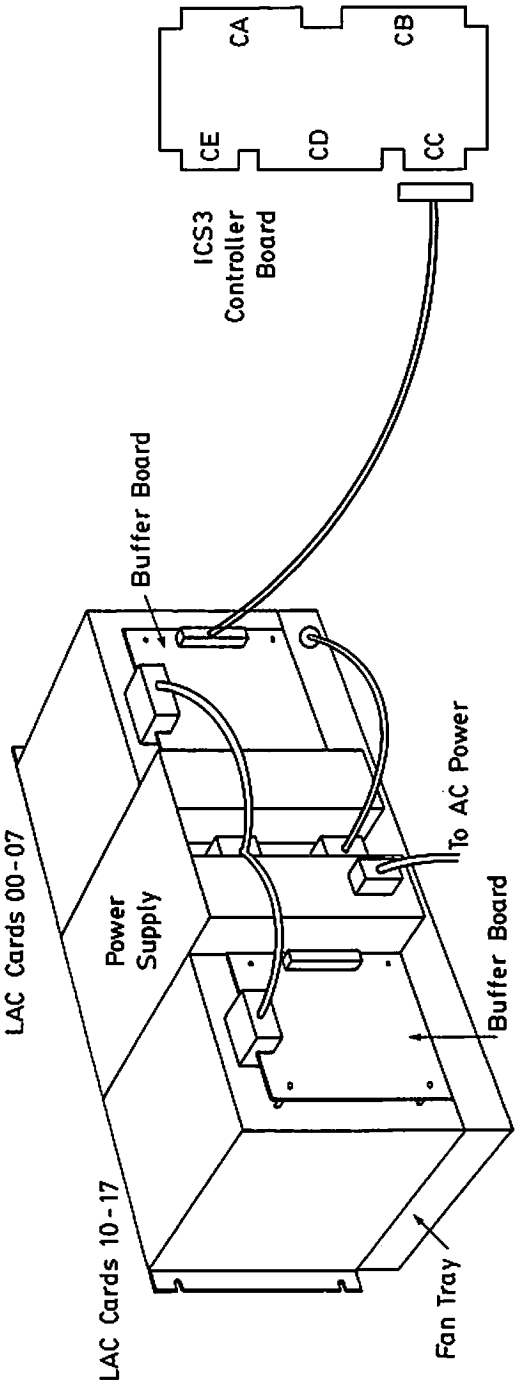


8-LAC ICS3 Card Cage
Figure 5-4

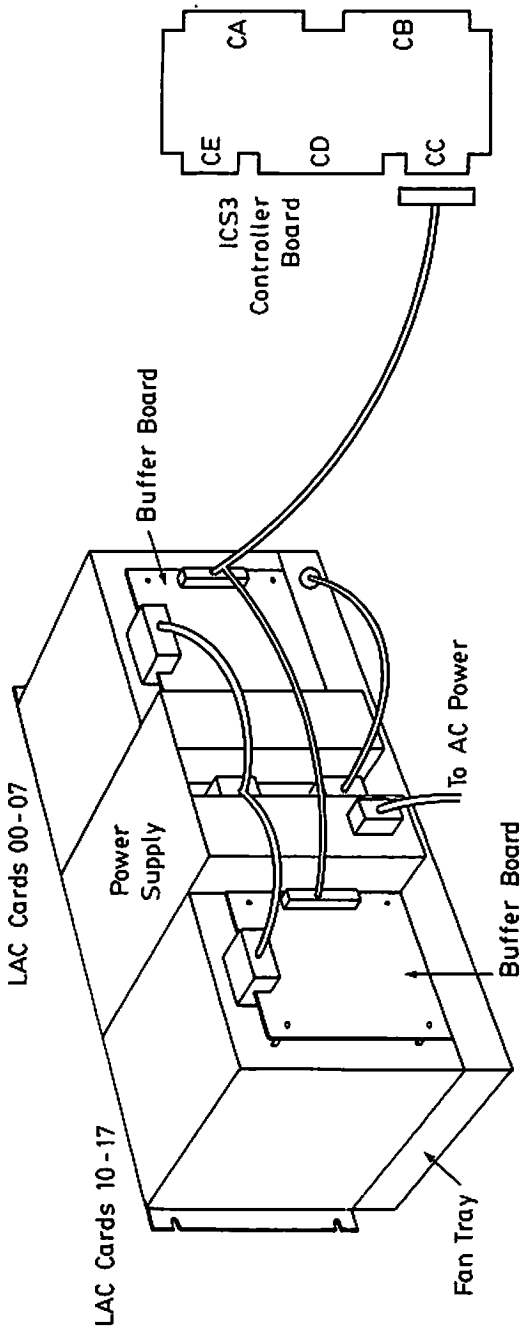




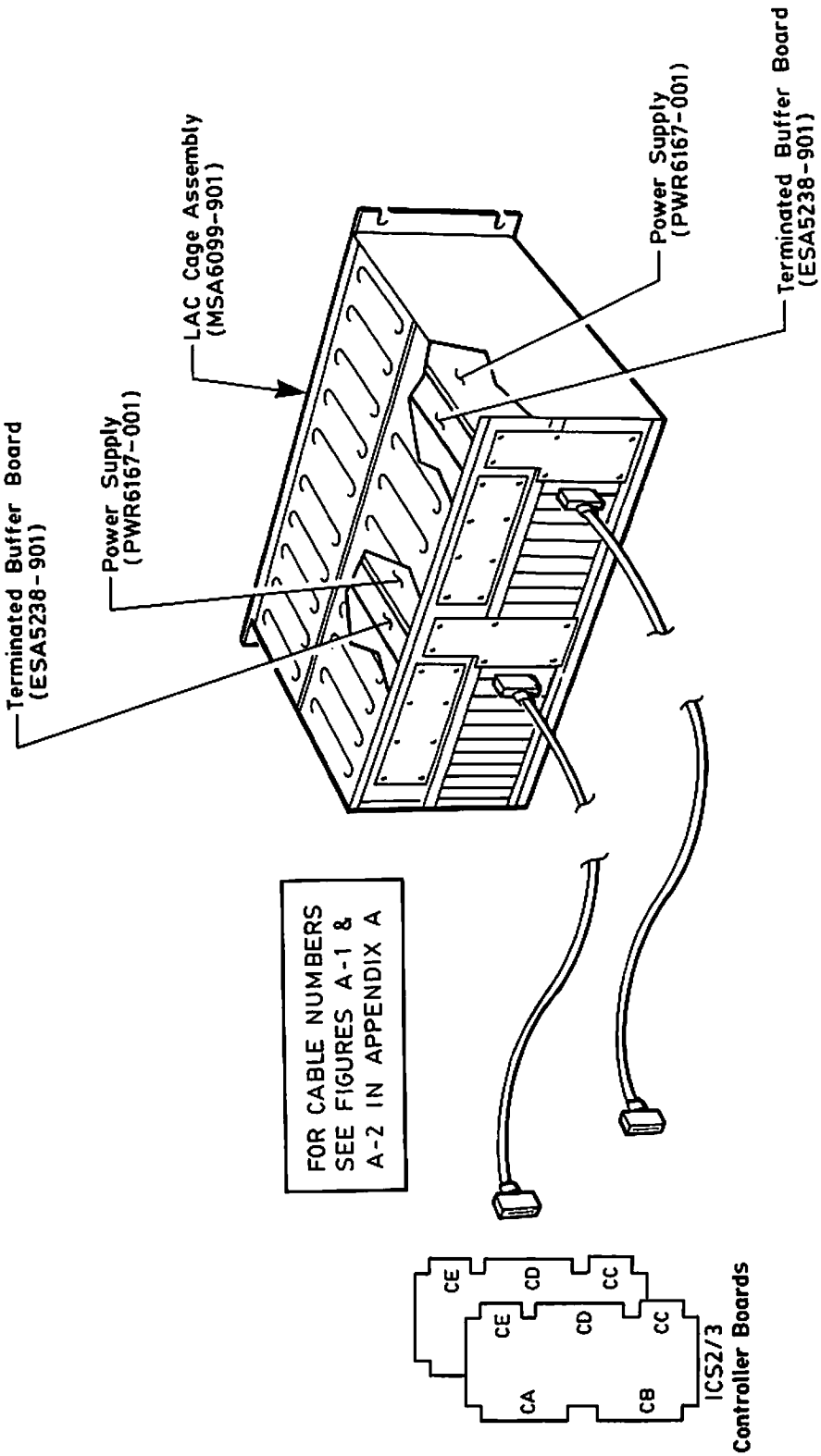
16-LAC ICS3 Card Cage
Figure 5-5



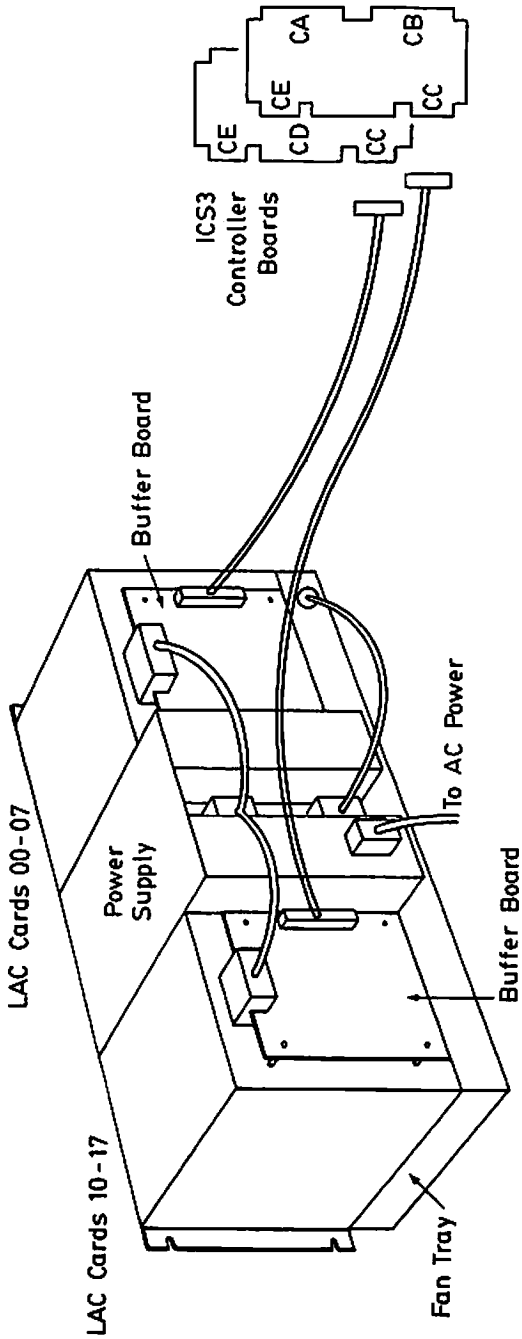
16-LAC ICS3 Card Cage, Single-complement Configuration
Figure 5-6



16-LAC ICS3 Card Cage, Double-complement Configuration
Figure 5-7



ICS2 Card Cage, Dual Single-complement Configuration
Figure 5-8



16-LAC ICS3 Card Cage, Dual Single-complement Configuration
Figure 5-9

LAC Slot Addressing

The slot/addressing schema are different for synchronous (non-SDLC), synchronous (SDLC), and asynchronous lines. Each is described below. If LACs are mixed, as they probably will be, simply use the appropriate scheme for the type of protocol being handled, as follows.

Synchronous (Non-SDLC) Lines: There may be one through eight, or one through sixteen, LACs installed in a card cage.

ICS2 Card Cage: The LACs plug into slots numbered (on the cage) 6 through 13 and 19 through 26, in two groups of eight, from left to right in the cage. The synchronous protocol software uses a LAC addressing scheme where the LACs are numbered 0 through 7 and 8 through 15 for the two groups. In general, this number is referred to as the "slot ID" (in this book).

Figure 5-10 shows line numbers for all synchronous LACs that are used for non-SDLC lines in a 16-LAC ICS2 card cage. The numbers at the top of the diagram, 1 through 26, are those printed on the LAC cage.

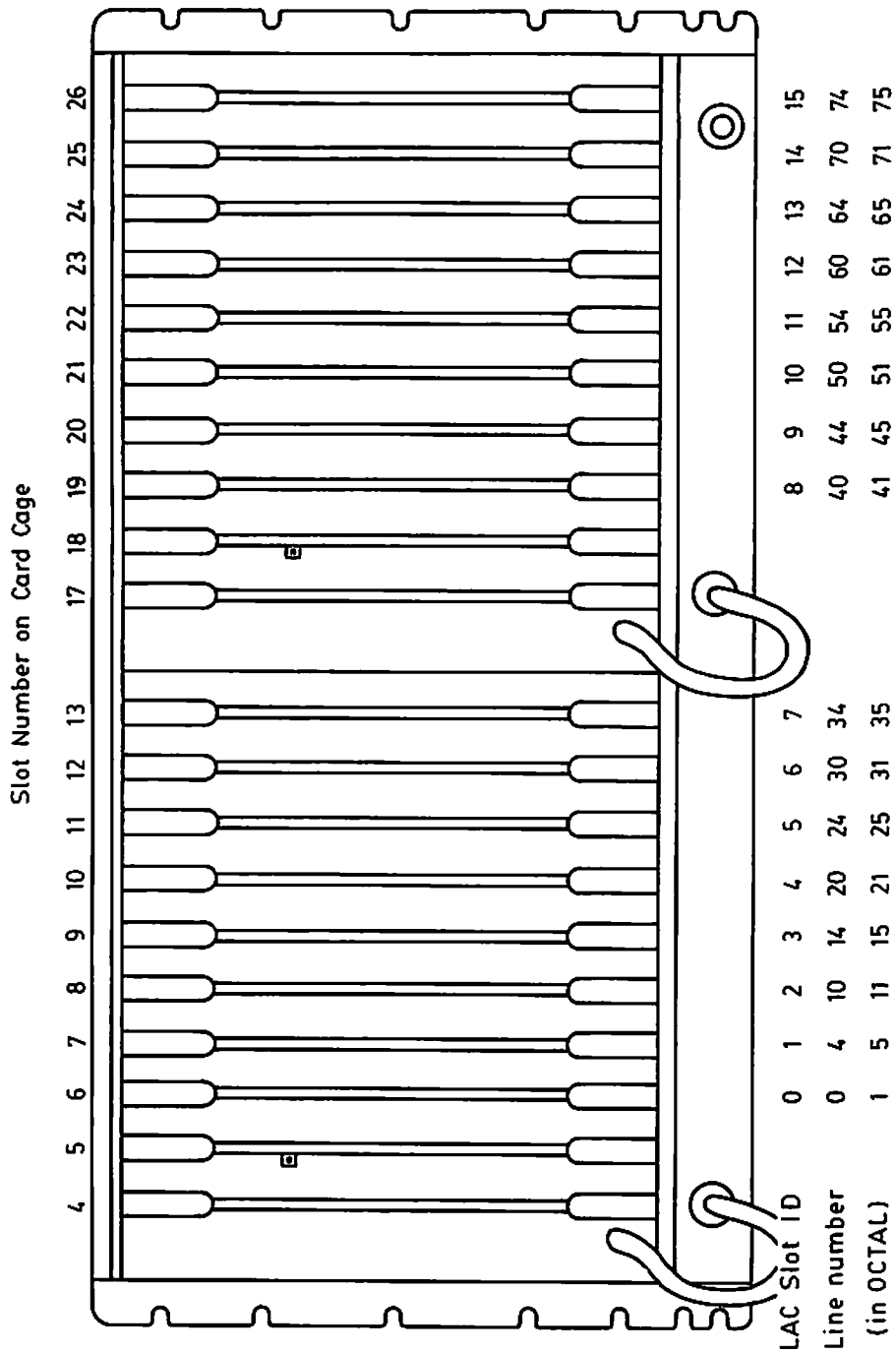
The physical line numbers for a synchronous (non-SDLC) LAC are slot ID multiplied by 4, for the first line; and slot ID multiplied by 4, plus 1, for the other line. Convert both line numbers to octal. The first line appears on J1 and the other on J2, of the peripheral bulkhead.

Each power supply takes up four slots (1 through 4, or 14 through 17), but plugs into only one connector, in slot 4 or in slot 17.

Each buffer board buffers signals for the adjacent eight LACs. The buffer board in slot 5 buffers all LACs in slots 6 through 13, and the buffer board in slot 18 buffers all LACs in slots 19 through 26.

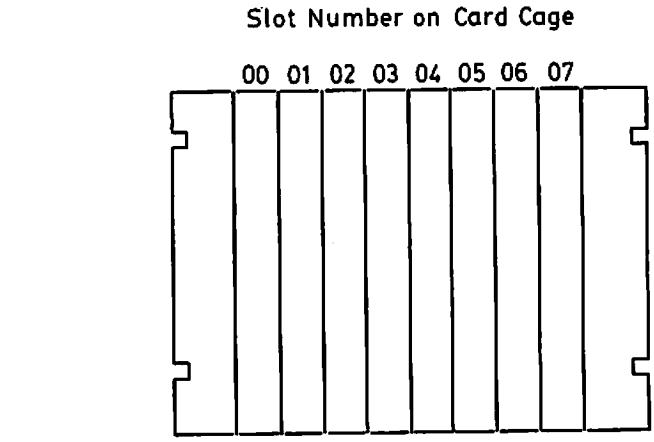
The slot IDs are shown for all slots; however, these IDs apply only where non-SDLC synchronous LACs are fitted.

Empty slots, asynchronous LACs, and SDLC synchronous LACs, make no difference to the non-SDLC synchronous LAC addressing/numbering scheme.



ICS2 Card Cage Addressing — Synchronous (non-SDLC)
Figure 5-10

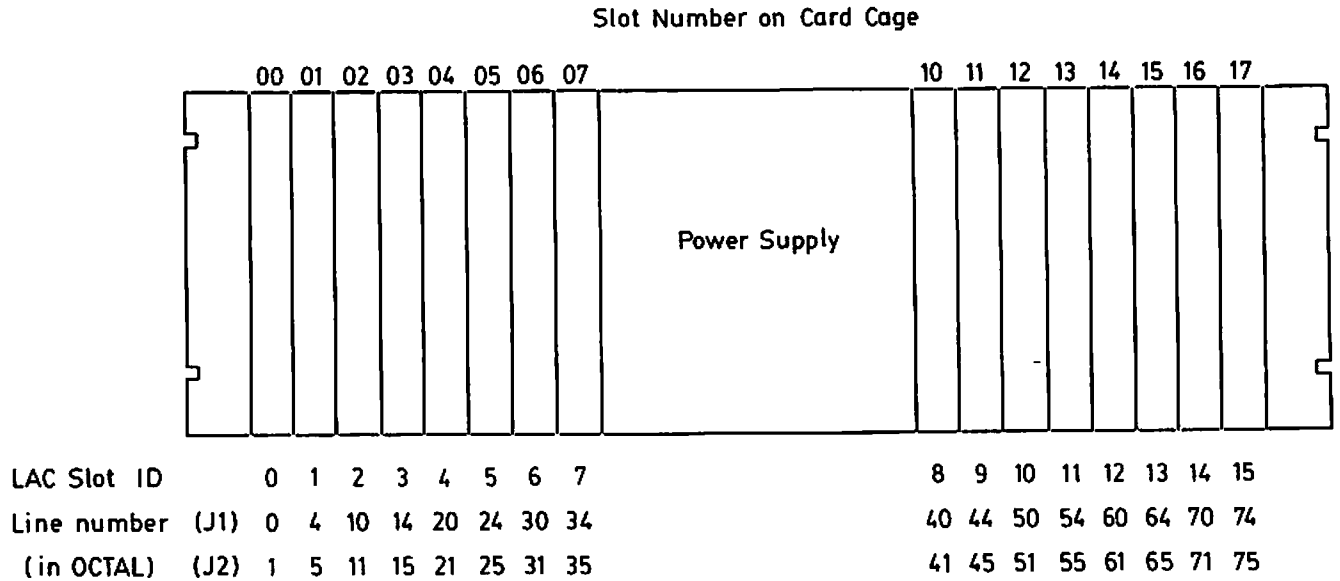
8-LAC ICS3 Card Cage: This card cage uses the same line numbering system as the ICS2 and ICS3 cages. It is a smaller version of half of the ICS2 cage and contains only eight LAC slots. Figure 5-11 shows the cage, from the front/integral bulkhead side, with the line numbers assigned to each slot. J1 and J2 are on the integral bulkhead.



LAC Slot ID	0	1	2	3	4	5	6	7
Line number (J1)	0	4	10	14	20	24	30	34
(in OCTAL) (J2)	1	5	11	15	21	25	31	35

8-LAC ICS3 Card Cage Addressing — Synchronous (non-SDLC)
Figure 5-11

16-LAC ICS3 Card Cage: This card cage is a smaller version of the ICS2 card cage. Use the method described above to calulate line numbers. Figure 5-12 shows the cage, viewed from the front/integral bulkhead side, with the line numbers assigned to each slot. J1 and J2 are on the integral bulkhead.

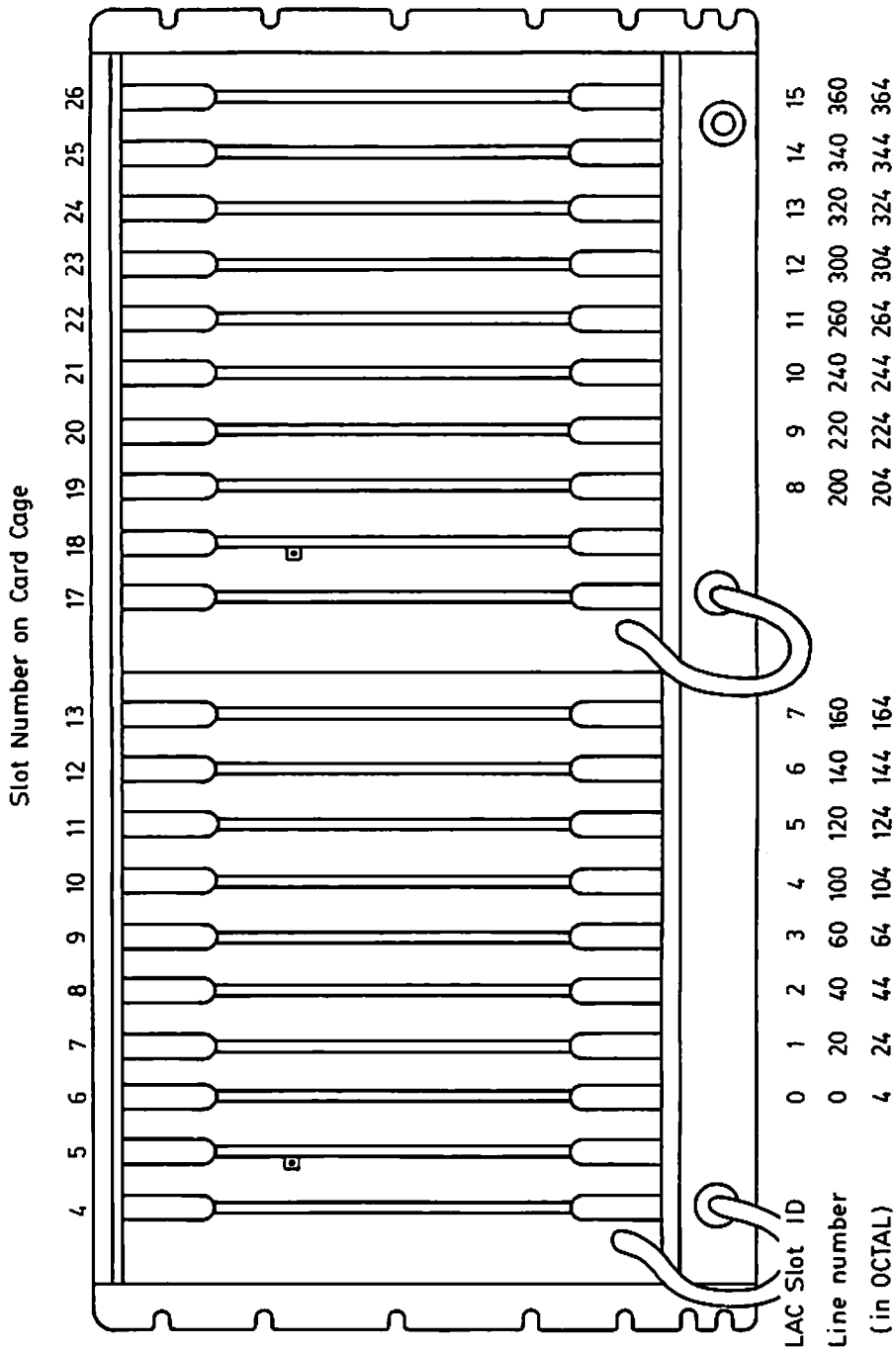


16-LAC ICS3 Card Cage Addressing — Synchronous (non-SDLC)
Figure 5-12

Synchronous (SDLC) Lines: This section describes the addressing schema for each type of LAC card cage.

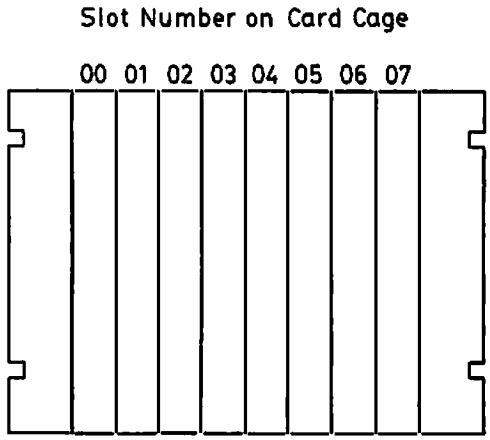
ICS2 Card Cage: Figure 5-13 shows line numbers for all or any SDLC LACs fitted to a ICS2 card cage. These line numbers can be calculated by multiplying the slot ID number by 16, for the first line on a LAC, and adding four for the other line on the same LAC. Convert both numbers to octal.

Figure 5-13 shows the numbers assigned to each slot. The lower-numbered line is physically on the top half of the LAC, and its signals will appear on J1 on the peripheral bulkhead. The higher-numbered line is on the lower half of the LAC, and its signals appear on J2 of the peripheral bulkhead (see Figures A-6 and A-7).



ICS2 Card Cage Addressing — Synchronous (SDLC)
Figure 5-13

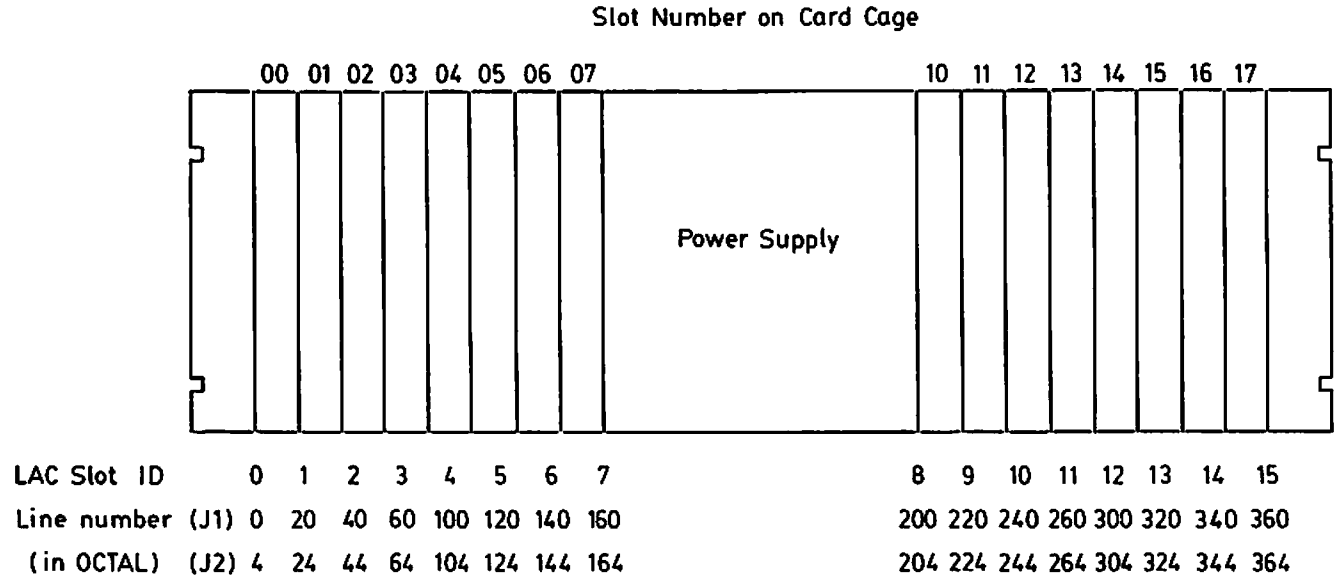
8-LAC ICS3 Card Cage: This cage uses the same system as the ICS2 card cage to determine line numbers. Connectors J1 and J2 are on the integral bulkhead on the edge of the LACs (see Figures A-9 and A10). Figure 5-14 shows the line numbers assigned to each slot.



LAC Slot ID	0	1	2	3	4	5	6	7
Line number (J1)	0	20	40	60	100	120	140	160
(in OCTAL) (J2)	4	24	44	64	104	124	144	164

8-LAC ICS3 Card Cage Addressing — Synchronous (SDLC)
Figure 5-14

16-LAC ICS3 Card Cage: This cage also uses the same system as the ICS2 card cage to determine line numbers. The connectors J1 and J2 are part of integral bulkhead on the edge of the LACs (see Figures A-9 and A-10). Figure 5-15 shows the line numbers assigned to each slot.



16-LAC ICS3 Card Cage Addressing — Synchronous (SDLC)
Figure 5-15

Asynchronous Lines: Asynchronous LACs have four lines per card. When installed in the LAC card cage, asynchronous LACs may be mixed with synchronous LACs at random. (See LAC Priority below.)

For asynchronous LACs, the asynchronous support software counts the LACs in the cage and allocates line numbers sequentially starting after the last AMLC line (if AMLCs are present). This ICS line number is always rounded up to the next sixteen-line boundary. Therefore, after an eight line AMLC, eight logical lines numbers are skipped, and the next sixteen line boundary number is allocated.

ICS1/2/3 asynchronous lines are always allocated after AMLC lines. ICS1/2/3 asynchronous line allocation order depends on the device address order, for those ICS1/2/3s, in the PRIMOS operating system. Normally the ICS2/3, with address '10 or '11 has lines allocated before the ICS1, with address of '36 or '37.

For example, assume that the AMLC has been allocated logical line numbers 0 through '17, and an ICS2/3 controller was connected with asynchronous LACs in various slots of the LAC card cage. Logical line numbers would be allocated as follows. The first asynchronous LAC (lowest slot number) would be allocated logical lines '20, '21, '22 and '23, the next async LAC (next lowest slot number), lines '24, '25, '26, and '27, and so on. It does not matter that the async LACs are not in adjacent slots, nor that sync LACs are interspersed; they will be allocated logical line numbers in the order in which they appear. The location of asynchronous LACs may be defined by the optional ICS CARDS directive. (See ASYNCHRONOUS CONFIGURATION later in this chapter.)

For ICS2 LACs, the four asynchronous circuits on the LAC (0, 1, 2 and 3) appear at the peripheral bulkhead on four 9-pin D-type connectors, J1, J2, J3, and J4, respectively (see Figure A-5 in Appendix A).

ICS3 LACs have integral bulkheading. The four asynchronous circuits (0, 1, 2, and 3) appear on four 9-pin D-type connectors (J1, J2, J3 and J4) respectively, on the edge of the LAC (see Figure 5-8 in Appendix A). The edge of the LAC is reinforced to form the integral bulkhead.

LAC Priority

The slots into which LACs plug are serviced in a particular order. Plugging certain LACs in the first-serviced, or higher-priority slots, improves operating efficiency. General rules for selecting LACs for higher priority servicing follow. These are general rules; each configuration must be assessed individually.

ICS2 Card Cage: The priority order for servicing LACs is the following; slot ID numbers 7 through 0 and then 15 through 8. Therefore, LACs requiring higher priority service should be placed in higher numbered slots in each group of eight. LACs requiring the highest priority should be placed in the first group of eight (7 through 0).

In general, it is recommended that LACs be priority ordered, from highest to lowest, as follows:

- Block mode asynchronous
- Bisynchronous-framed X.25 or RJE
- SDLC (PRIME/SNA) and HDLC
- Asynchronous

Higher speed lines in each group should be placed in higher priority slots.

On a synchronous LAC, the order of priority of its two lines (high to low) is 1, 0.

On an asynchronous LAC, the order of priority of its four lines (high to low), is 1, 0, 3, 2.

16-LAC ICS3 Card Cage: This card cage uses the same system of determining priority (or order of servicing) as the ICS2 cage. The order is, from high to low, slot 7 through 0 and then slot 17 through 10 (slot numbers in octal).

8-LAC ICS3 Card Cage: This cage uses the same system as the other cages. The order is, from high to low, slot 7 through 0.

SOFTWARE CONFIGURATION

This section of the chapter details network software configuration issues associated with integration of ICS2/3 support into PRIMOS for PRIMENET/X.25, RJE, PRIME/SNA (SDLC), and asynchronous protocols.

Configuration Directives

Physical line to logical line mapping is dictated by the configuration directives loaded into the CONFIG file. The configuration directives are loaded at cold start and consist of the following formats:

- SYNC ON
- SYNC CNTRLR controller-number device-address protocol
- SYNC SYNCnn controller-number line-number

System Administrators will be familiar with the configuration directives:

- SMLC ON
- SMLC CNTRLR controller-number device-address protocol
- SMLC SMLCnn controller-number line-number

SMLC is a synonym for SYNC; consequently, their formats are identical to the SYNC formats.

▶ SYNC ON

SYNC ON configures synchronous communication drivers for all synchronous communication controllers, including the ICS2/3, and must be specified if synchronous lines will be used for products other than PRIMENET.

Default configuration is given below. (Note that 'xx is octal notation.)

Logical Line Number	Logical Controller	Controller Address	Controller Physical Line Number
00	0	'50	0
01	0	'50	1
02	0	'50	2
03	0	'50	3
04	1	'100000	0
05	1	'100000	1
06	1	'100000	2
07	1	'100000	3

Logical lines 00 through 03 are mapped to logical controller 0. The physical device address is '50, with physical line numbers 0 through 3. (Note that the MDLC device address is usually '50 or '51.)

Logical lines 04 through 07 are mapped to physical lines 0 through 3 on logical controller 1. Controller 1's physical address is '100000, indicating the controller is disabled. To enable controller 1, set its address to a valid device address ('50 or '51) with the SYNC CNTRLR directive.

The default configuration can be changed with the SYNC CNTRLR and SYNC SYNCnn directives, explained below. SYNC CNTRLR changes the mapping of logical controller to physical address. SYNC SYNCnn changes the mapping of a single logical line number.

Either directive may be used separately. If both are needed, then the SYNC CNTRLR directive(s) must be given first. In other words, the controller must have been assigned its correct physical address before any new SYNC lines are assigned to it.

Synchronous line configuration should always be specified explicitly. Do not rely on default configuration for lines to be used or those to be left unused.

Take care to override the default configuration when synchronous lines are configured on two controllers, one of which is not device address '50.

▶ SYNC CNTRLR <controller-number> <device address> [protocol]

The SYNC CNTRLR assigns a physical controller address to a logical controller-number with a particular protocol. This directive must be given before any SYNC SYNCnn directive. It enables an ICS2/3 to handle synchronous communications.

If the SYNC CNTROLR directive is absent, the default configuration is used. The default DDL files are loaded, ICS2_01.DL (ASYNC only) for the ICS2, and ICS3_09.DL (ASYNC_HDLC_BSCRJE_BCSX25) for the ICS3.

The variables are

<controller-number> The logical controller, 0, 1, or 7. Any other number produces the error message, BAD SYNC CONTROLLER MAPPING COMMAND. The 7 is for SDLC or ASYNC_SDLC (only) and allows multiple controllers (that only support SDLC) to be configured. You can have more than one controller-number 7.

<device-address> The physical device address of the specified controller given in octal: typically '10 or '11 for an ICS2/3, and '36 or '37 for an ICS1. The default values are address '50 for controller 0 and '100000 (disabled address) for controller 1. Note that device address '50 is usually an MDLC. If controller 1 is used, its address should not conflict with the address of any other peripheral controller.

[protocol] This field can be used only with ICS2/3 controllers and should not be used with any other synchronous controller. (See Note below.) See Tables 5-1 and 5-2 for valid protocol combinations. This field selects the correct downline load file for the specified ICS2/3 and protocols.

Notes

If a protocol token is entered for a non-ICS2/3 controller, the following error message results:

```
Error:  Controller xx does not support sync protocols
          (COMINI)
```

where xx is the device-address of the controller.

If the protocol token is omitted, the following message appears:

```
Error: protocol combination not supported on ICS2
       device address dd (BTPCC).
```

If the SYNC CNRTLR directive is omitted entirely, the default protocol (ASYNC) is loaded.

If you map one logical controller to an address already mapped, SYNC automatically disables the previously mapped controller (without warning), setting its address to '100000. The disabled controller may be enabled again by a new mapping directive.

The following directives map controller 1 to address '10 and controller 0 to address '11. Appropriate downline load files to support the specified protocol combination are selected with the protocol field.

```
SYNC CNTRLR 1 10 ASYNC_HDLC
SYNC CNTRLR 0 11 BSCRJE_BSCX25
```

Note that the second example above disables the default configuration for '50.

The operator may also disable a logical controller by setting its address to blank or to '100000. For example, controller 0 could be disabled by either of these directives:

```
SYNC CNTRLR 0 100000
SYNC CNTRLR 0
```

Valid Protocol Token Combinations

Valid protocol tokens are ASYNC, SDLC, HDLC, BSCX25, and BSCRJE. A token combination need not be entered in the order shown in Tables 5-1 and 5-2; any order will work, providing an underscore separates the parts.

ICS2: Table 5-1 lists valid protocol token combinations, microcode image names, and downline load file numbers used for the ICS2. The ICS2 cannot support all protocols concurrently because of memory size limitations.

When BSCRJE (or BSCX25), and ASYNC are used together, there is no asynchronous reverse flow control (RFC). This lack of RFC is due to shortage of Writable Control Storage (WCS) memory.

Table 5-1
ICS2 Valid Protocol Token Combinations

Protocol Token	Microcode Image*	DLL-File Number
ASYNC	HSAXX	ICS2_01.DL
SDLC	HSAXX	ICS2_02.DL
HDLC	HSAXX	ICS2_03.DL
BSCRJE_BSCX25	HSBXX	ICS2_04.DL
BSCX25	HSBXX	ICS2_04.DL
ASYNC_SDLC	HSAXX	ICS2_05.DL
ASYNC_HDLC	HSAXX	ICS2_06.DL
HDLC_SDLC	HSAXX	ICS2_07.DL
BSCRJE_BSCX25_ASYNC (Note 1)	ABXX	ICS2_08.DL
ASYNC_BSCX25 (Note 1)	ABXX	ICS2_08.DL
BSCRJE_BSCX25_SDLC	HSBXX	ICS2_09.DL
BSCRJE_SDLC	HSBXX	ICS2_09.DL
BSCX25_SDLC	HSBXX	ICS2_09.DL
BSCRJE_BSCX25_HDLC	HSBXX	ICS2_10.DL
BSCRJE_HDLC	HSBXX	ICS2_10.DL
ASYNC_HDLC_SDLC	HSAXX	ICS2_11.DL
BSCRJE	HSBXX	ICS2_12.DL
BSCRJE_ASYNC (Note 1)	ABXX	ICS2_13.DL
BSCX25_HDLC	HSBXX	ICS2_14.DL
BSCRJE_BSCX25_HDLC_SDLC	HSBXX	ICS2_15.DL
BSCX25_HDLC_SDLC	HSBXX	ICS2_15.DL
BSCRJE_HDLC_SDLC	HSBXX	ICS2_15.DL

* XX is the version number.

Notes

In an ICS2, when a combination of BSC and ASYNC protocols is selected (as in BSCRJE_BSCX25_ASYNC, BSCRJE_ASYNC, and ASYNC_BSCX25), there is no asynchronous, reverse flow control (RFC) facility.

Asynchronous lines, on an ICS2 using any of the above protocol combinations, can still be configured for RFC. (Configuration is done by using the AMLC command, in which RFC is enabled by setting bit 11 of the configuration word.) The ICS2, however, will ignore the RFC-enable for the above protocol combinations only. PRIMOS will issue the following message indicating that RFC is not operating:

> Asynchronous Reverse Flow Control is not an option for the protocol combination on the controller supporting this line. (AMLC)

ICS3: The ICS3 is available with 256K, 512K, and 1024K bytes of RAM. In the 512KB ICS3, any combination of protocol tokens is allowed. When all protocols are required, the downline file loaded is ICS3_08.DL. The 256KB ICS3 will handle everything except the two largest protocol combinations, ASYNC_SDLC_HDLC_BSCX25_BSCRJE and SDLC_BSCX25_BSCRJE.

Any combination not containing SDLC causes downline load file ICS3_09.DL to be loaded. This file contains all protocols except SDLC. For any combination containing SDLC, the loaded file contains only SDLC and the other protocol(s), resulting in more economical running than when loading the "all protocols" file, ICS3_08.DL.

Table 5-2 lists the protocol tokens, the microcode image names and the downline load file numbers used for the ICS3.

To conserve memory space, the system administrator may choose to delete unused downline load files and maps from the DOWN_LINE_LOAD* directory.

Table 5-2
ICS3 Valid Protocol Token Combinations

Protocol Token	Microcode Image*	DLL-File Number
SDLC	ABHSXX	ICS3_01.DL
ASYNC_SDLC	ABHSXX	ICS3_02.DL
SDLC_HDLC	ABHSXX	ICS3_03.DL
SDLC_BSCX25	ABHSXX	ICS3_03.DL
SDLC_HDLC_BSCX25	ABHSXX	ICS3_03.DL
ASYNC_SDLC_HDLC	ABHSXX	ICS3_04.DL
ASYNC_SDLC_BSCX25	ABHSXX	ICS3_04.DL
ASYNC_SDLC_BSCX25_HDLC	ABHSXX	ICS3_04.DL
SDLC_BSCRJE	ABHSXX	ICS3_05.DL
ASYNC_SDLC_BSCRJE	ABHSXX	ICS3_06.DL
SDLC_HDLC_BSCRJE	ABHSXX	ICS3_07.DL
SDLC_BSCX25_BSCRJE	ABHSXX	ICS3_07.DL
SDLC_HDLC_BSCX25_BSCRJE	ABHSXX	ICS3_07.DL
ASYNC_SDLC_HDLC_BSCRJE_BSCX25	ABHSXX	ICS3_08.DL
ASYNC_SDLC_BSCRJE_BSCX25	ABHSXX	ICS3_08.DL
ASYNC_SDLC_HDLC_BSCRJE	ABHSXX	ICS3_08.DL
ASYNC_HDLC_BSCX25_BSCRJE	ABHSXX	ICS3_09.DL

* XX is the version number.

▶ SYNC SYNCnn <controller-number> <line-number>

SYNC SYNCnn maps logical line-numbers to physical line-numbers on a specified logical controller (used for X.25/RJE, but not for SDLC).

The variables are

nn	The logical line-number; values range from 00 through 07.
<controller-number>	The logical controller set by a SYNC CNTRLR directive, either 0, 1, or '100000. Use 0 or 1 to identify a controller; use '100000 if the specified line is not to be configured nor allocated memory. The default is '100000. (7 is not accepted.)
<line-number>	The physical line-number of the specified controller onto which the logical line-number is mapped. (For ICS2/3 sync LAC physical line numbering, see LAC Slot Addressing earlier in this chapter.) If the controller is an ICS1, line-number must be 0. The default values map SYNC00 through SYNC03 to physical lines 0 through 3 on the first controller, and SYNC04 through SYNC07 to physical lines 0 through 3 on the second controller. This value must be specified unless controller-number is '100000 or unspecified.

For example, the directive

```
SYNC SYNC04 1 3
```

assigns logical line 04 to physical line 3 on controller 1.

Setting the controller-number to a blank or to '100000 disables a logical line-number. For example, logical line 07 can be disabled by either of these directives:

```
SYNC SYNC07
SYNC SYNC07 100000
```

Giving any value for controller-number other than 0, 1, blank, or 100000, results in the following error message:

```
BAD SMLC LINE MAPPING COMMAND
```

Note

This error message will probably change when more SMLCs are replaced by ICS2/3s. The SMLC will probably become SYNC.

PRIMENET/X.25

A PRIMENET network is configured by the Systems Operator/Administrator issuing the CONFIG_NET command. This command guides the operator through the specifications needed to fully describe the network. CONFIG_NET validates the entered information and, when complete, creates a binary file. This file (PRIMENET.CONFIG), is read during network startup. The CONFIG_NET command is fully described in the Network Planning and Administration Guide.

The network is started by issuing the START_NET command. The command may be given automatically either by inclusion of the command in system cold start processing [C_PRMD or PRIMDS.COMI] or by the operator typing START_NET at the system console. The network configuration file is read by the associated START_NET routines and the network data structures are created. When the data structures are built, the Network Server Process (NETMAN) is spawned. All FDX lines specified in the network configuration file are assigned (if possible). No mechanism exists for the assignment/reassignment of FDX lines after network startup other than to reconfigure and restart the network. However, HDX lines may be assigned after startup, by use of the NET [-ASSIGN Line] command, as long as the HDX line is available and specified in Network Configuration. Note, however, that HDX PRIMENET is not supported on the ICS2/3 at Rev. 20.1.

The PRIMENET/X.25 initialization process that is responsible for assigning synchronous logical lines, for either FDX or HDX modes, passes appropriate configuration information to the ICS2/3 to mark data structures indicating line assignment and appropriate protocol information.

Currently, it is possible to assign HDX-configured lines after network startup. However, it is not possible to change the protocol associated with the configured line, as defined in the Network Configuration file, unless a new Network Configuration file is created and the network is restarted.

FDX configured lines may be re-assigned (for example, for RJE) once the network is shut down with the STOP_NET command.

RJE

The Prime emulators are informed of the characteristics of an RJE site through a Site Definition File (SDF) for each site serviced by an ICS2/3. The SDF is described in the Remote Job Entry Phase II Manual.

Each SDF contains a set of commands processed by the RJOP to specify the attributes of the remote site, including the name. When a TOSITE command, with a site name, is given by the RJE operator, the corresponding SDF is processed and the site characteristics are forwarded to the appropriate worker process. The line is enabled through the ENABLE command. Once the line has been enabled, reception and transmission over that line are conducted by the corresponding worker and protocol handler. In a similar way, other sites can be defined and enabled with the RJOP interface.

The CONNECT command is specified in the SDF to allow the RJOP to associate a synchronous logical line-number with a specified site name. All further commands refer to the site name instead of logical line-number. The logical line-number ranges 0 through 7. The RJOP is capable of handling a maximum of eight synchronous lines, each of which corresponds to a discrete site name.

Whenever an (non-SNA) RJE site is to be connected to a Prime CPU, a line is assigned by the RJE protocol handler because the SDF is processed by the RJOP command.

Successful processing of the SDF results in the appropriate configuration information being passed on to the ICS2/3 software. The ICS2/3 software marks internal data structures to indicate that the line has been assigned to an RJE process and to note protocol-dependent information relevant to that particular line.

PRIME/SNA (SDLC) CONFIGURATION

Some of the PRIME/SNA configuration directives are mentioned here. This description is a very brief introduction to the most commonly used directives only. Full details of these and additional directives may be found in the PRIME/SNA Administrator's Guide.

After installation of PRIME/SNA, CMDNC0 contains run files and configurator command programs for the Server Subsystem.

The configurator programs enable definition of SNA lines and remote systems which must be symmetric with definitions at the SNA host(s) connected to. The configurator programs result in an SNA configuration file that is used when the Server or Interactive subsystem is started, to establish or change configurations.

Server Subsystem Configuration

This is invoked by the command

```
SNA_SERVER_CONFIG [config-pathname] [options]
```

The config-pathname is the new or existing name of the SNA configuration file. Usually it is called PRIME/SNA*>SNA.CONFIG and this is the default file used if none is specified in this command and SNA_SERVER commands.

This configurator enables definition of names, numbers, and characteristics of each line, remote system, and LU ports being configured.

Invoking/Starting the Server

SNA_SERVER Command: The command

```
SNA_SERVER -START
```

starts the Server Subsystem as a phantom process and activates the necessary local support for lines and remote systems. You can specify the line or group of lines to be started by using this option:

```
SNA_SERVER -START -LINE LINENAME
                  -LINE_GROUP LINEGROUPNAME
```

For remote systems, use the directive

```
SNA_SERVER -START -REMOTE_SYSTEM REMOTESYSTEMNAME
                  -REMOTE_SYSTEM_GROUP REMOTESYSTEMGROUPNAME
```

This extracts, from the SNA configuration file, the information required to set up the environment for the remote systems named, and reports back to the user when the systems are started.

Once started, you can request server status with the command

```
SNA_SERVER -STATUS
```

or specify the statistics file by using the command

```
SNA_SERVER -STATISTICS [stats-file]
```

You can also set the amount of detail logged by using the command

```
SNA_SERVER -MESSAGE_LEVEL BRIEF/MEDIUM/DETAILED
```

PRIMOS Directives and Commands

The following PRIMOS directives may require some adjustment when using PRIME/SNA. For more detail, see the PRIME/SNA Administrator's Guide and the System Administrator's Guide.

SYNC CNTRLR: The logical controller-number must be 0, 1, or 7. The device-address is the ICS2/3 to be used for SDLC. The protocol entered must contain SDLC.

NPUSR: Add phantoms for SNA_SERVER, SNA_3270, and SNA_PRINT (where run as phantom).

AMLBUF: Increase the input buffer size to '2000 for 24 X 80 character screen, and '4000 for 27 X 132 character screen.

Set output buffer size according to line speed. As a guide use line speed divided by 40, and then converted to octal. For example, a 9600bps line would require an output buffer size of 240 (octal 360).

Set DMQ size according to line speed and interrupt rate. As a guide use: Line speed divided by ten, all divided by the interrupt rate. Round up to a power of two, and convert to octal. The interrupt rate is that set by the ICS INTRPT directive or the line speed of the last AMLC line, divided by ten.

Buffer sizes must be increased for any terminal (local or remote) that will run SNADSC.

AMLIBL: Sets the tumble-table sizes for the AMLC controller. Must be increased for any terminal (local or remote) that will run SNADSC.

ICS INPQSZ: Sets input queue size (recommended values '1777 or '777), depending on the memory constraints of the Prime system.

ICS INTRPT: Sets ICS2/3 interrupt rate. Recommended value '36, which is equivalent to 300bps on a last AMLC line.

REMBUF: Sets size of input and output buffers for remote users. (Recommended value '1000 each on local system when remote users will use PRIME/SNA.)

Also increase same buffers on any intermediate Prime system through which a user will NETLINK or REMOTE-LOGIN.

AMLC: You must enable Reverse Flow Control by setting bit 11 of the configuration word.

ASYNCHRONOUS CONFIGURATION

Asynchronous lines may be connected to an Asynchronous Multi-line Controller (AMLC) ICS1, ICS2, or ICS3. This description generally applies to the ICS2 and ICS3.

This section describes the following aspects of asynchronous protocol configuration:

- Asynchronous lines generally
- Difference between ICS2 and ICS3 asynchronous LACs
- Asynchronous LAC Slot Addressing
- Asynchronous directives for ICS2/3

Asynchronous Lines

Asynchronous lines generally connect low to medium speed serial communications devices or services to a Prime system. Speeds range from 75 bps to 19200 bps. An ICS2/3, with 16-LAC slot card cage, can support up to 64 asynchronous lines (each of the 16 LACs has four lines). An 8-LAC card cage can support up to 32 asynchronous lines. Asynchronous lines typically connect local terminals, printers, plotters, and modems to more remote terminals or other Prime (or other brand) systems.

Difference Between ICS2 and ICS3 LACs

The ICS2 asynchronous LAC uses Zilog's Serial Input/Output (SIO) device, while the ICS3 uses Zilog's Serial Communications Controller (SCC) device. The ICS3 controller can handle both the SIO and the SCC, but the ICS2 can only support the SIO devices. The controller knows which type of LAC is installed by reading an 8-bit ID register on the LAC. This ID is common for all lines on the LAC; the individual lines are selected by a unique line address number being placed in the IBC line address register just prior to requesting a LAC bus read or write cycle.

The primary function of both the SIO and the SCC is to provide serial to parallel and parallel to serial conversion for each of the four lines on the LAC. There are two dual-channel SIOs or SCCs on a LAC. Other functions performed on the LAC include baud rate generation and loopback control.

Asynchronous LAC Slot Addressing

For asynchronous LAC slot addressing see LAC Slot Addressing, Asynchronous Lines earlier in this chapter, which describes all LAC slot addressing schema. The asynchronous scheme is common to ICS1, ICS2 and ICS3.

The AMLC Command

The AMLC command is used to configure asynchronous lines for the ICS1, ICS2, and ICS3, just as it is for the Asynchronous Multi-Line Controller (AMLC).

The AMLC command allows specification of numerous options and parameters, and is fully described in Chapter 10 of the System Administrator's Guide.

Asynchronous Directives for the ICS2/3

The following directives are used for configuring a ICS2 or ICS3 for asynchronous operation. Additional explanation of the directive is given where warranted; however, for a full description, see Chapter 10 of the System Administrator's Guide.

AMLBUF: Specifies the size of input, output and DMQ buffers.

AMLCLK: Sets the programmable clock in an AMLC. If used for a ICS1, ICS2, or ICS3, the baud rate (speed) selected must be one of those legal for the ICS JUMPER directive. (The ICS JUMPER directive is being replaced by the ASYNC JUMPER directive. Both are currently supported.)

AMLTIM: Sets three timers: ticks, disctime, and gracetime. On an ICS1/2/3, the disctime must be at least twice the value of ticks. A disctime value between 10 and 20 seconds is recommended to allow the line to return to speed-detect mode if the caller hangs up without logging in.

ICS INPQSZ: Specifies the size of the input queue, for all lines, for data going from the ICS1/2/3 to the asynchronous DIM.

ICS INTRPT: Specifies the CPU interrupt rate for asynchronous services on the ICS1/2/3. (See table of values in Chapter 9 of the System Administrator's Guide.

ICS JUMPER: This is being replaced by ASYNC JUMPER; both are currently supported. These directives set three line speeds for the ICS1/2/3, from a list of allowable speeds, and in conjunction with the configuration word in the AMLC directive.

NAMLC: Specifies the number of assigned asynchronous lines for AMLC and ICS1/2/3.

NTUSR: Specifies the number of local terminal users, including the supervisor terminal for AMLC and ICS1/2/3.

ICS CARDS Directive

This optional directive specifies, for a given ICS2/3 controller, which slots are occupied (or not) by asynchronous LACs. The purpose of this directive is to ensure that the same logical line number to physical line number mapping is maintained, even if a particular asynchronous LAC is removed or becomes inoperative. The format is as follows:

▶ ICS CARDS <device-address> <config-word>

<device-address> The controller address in octal; legal values are currently, '10, '11, '36, and '37.

<config-word> A 16-bit word in which each bit represents a slot in the LAC Card Cage. If the bit is a 1, it represents the presence of an async LAC. If the bit is a 0, it represents a synchronous LAC or an empty slot.

For example, consider the following directive and its breakdown.

ICS CARDS 10 114764

This directive defines the asynchronous card configuration for the controller, device-address '10.

The config-word, 114764, is derived as follows:

config-word bit number	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
slot number ICS2 card cage	6	7	8	9	10	11	12	13	19	20	21	22	23	24	25	26
slot number ICS3 card cage	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17
async card in slot = 1	1	0	0	1	1	0	0	1	1	1	1	1	0	1	0	0
config-word	1			1			4			7			6			4

The ICS CARDS directive thereby tells the system how the ICS2/3 is configured with async LACs. This configuration is checked at time of cold start and warm start as follows.

Cold Start: At cold start, the actual ASYNC configuration is compared with that defined by the ICS CARDS directive. A check is made for the following:

- An asynchronous LAC where it is not expected
- A faulty LAC where an async LAC was expected
- A sync LAC where an async LAC was expected
- An empty slot where an async LAC was expected

If a difference is found, one of the following four messages is displayed on the system console:

- Inconsistent ASYNC cold start configuration for ICS2 device dd: an async line card has been found where not expected in slot ss.

- Inconsistent ASYNC cold start configuration for ICS2 device dd: faulty line card in slot ss where an async line card was expected.

- Inconsistent ASYNC cold start configuration for ICS2 device dd: sync line card in slot ss where an async line card was expected.

- Inconsistent ASYNC cold start configuration for ICS2 device dd: slot ss is empty where an async line card was expected.

A further error message may be displayed at cold start time if the number of status words returned by a ICS2 controller is incorrect. The message is as follows:

> ICS2 device dd has returned the wrong number (m) of status words.

Warm Start: At warm start, the actual ASYNC configuration is also compared with that defined by the ICS CARDS directive. LAC characterization is saved on the controller (from cold start) across a warm start. The ICS2/3 uses this cold start characterization to reprogram the LACs to their former state. Missing, faulty, or additional LAC cards are checked for, and one of the following messages is output to the system console where appropriate:

- Inconsistent ASYNC warm start configuration for ICS2 device dd: an async line card has been inserted into slot ss.

- Inconsistent ASYNC warm start configuration for ICS2 device dd: the async line card in slot ss in now inoperable.

- Inconsistent ASYNC warm start configuration for ICS2 device dd: the async line card in slot ss has been removed or is now inoperable.

- Inconsistent ASYNC warm start configuration for ICS2 device dd: a sync line card has been inserted into slot ss.

Warm or Cold Start: An additional error message may be displayed at either warm or cold start if the key value in call ICSCFG is incorrect. The message is

Bad key value (n) in call (ICSCFG).

Notes

In all the above messages, the variables are as follows:

The symbol dd is the octal device address, such as, 10, 11, 36, and 37.

The symbol ss is the slot number; range equals 1 through 16.

The symbol m is the actual number of status words returned by the ICS2.

The symbol n is the key value used to call the routine ICSCFG.

6
Performance and Restrictions

INTRODUCTION

This chapter provides information on the performance of the ICS2/3 at Rev. 20.1 and also lists some configuration restrictions. The figures and information provided here may not apply to pre-Rev. 20.1 software.

Generally, to accurately predict performance, we must consider the actual combination of mainframe; type and number of terminals; their workload; protocols used; and other peripherals. The following information is a guideline only, to be used for clearer understanding of configuration and performance issues. Customers should always consult with local Prime support staff on configuration and performance issues.

In general, a detailed discussion of the PRIMOS operating system and other Prime host software performance is outside the scope of this book. Where possible, however, guidelines are given for total system configuration limits and throughput. In some cases other influences, such as Prime host software, may modify speeds or number of lines supported to figures below those calculated for the controller alone.

The important factors in determining the communications performance of a Prime system equipped with an ICS2/3 include the following:

- Prime Host CPU power
- ICS2/3 — Prime software/hardware interface
- Z8001 software performance for each protocol
- IBC software performance
- Line speeds

This chapter provides information about each supported protocol and protocol behavior under conditions of stress. Following the specific protocol information are data on overall performance/configuration and an example.

PROTOCOL PERFORMANCE

ICS2/3 — Prime Interface

Synchronous throughput across the backplane depends mainly on the number of individual frames of data, rather than the quantity of data these frames contain. The ICS2/3 can currently handle approximately 300/600 frames per second (X.25) respectively. This speed is determined by the management of buffers between the ICS2/3 and the Prime CPU rather than by the speed of backplane hardware or DMQ. Lower-end Prime machines, such as the 2250, can handle approximately 240 frames per second, while high-end machines can handle much higher rates (in proportion to their approximate CPU performance). A machine with twice the power of a 2250 can handle approximately 500 frames per second. Thus, for low-end machines, the Prime CPU limits the speed across the backplane, and for higher-powered machines, the ICS2/3 becomes the limiting factor. The higher-powered Prime CPU, therefore, supports greater throughput if communications are spread over multiple ICS2/3s.

Asynchronous data is transmitted directly across the backplane rather than blocked into buffers. The speed of this transfer is very high and thus does not have any significant effect on asynchronous performance.

Z8001 Software Performance

Z8001 software performance can be quantified in terms of data transit time through the Z8001 software as well as Z8001 CPU time (on a per-character or data-block basis). Synchronous traffic through the ICS2/3 experiences insignificantly short transit times of approximately several hundred microseconds per frame or block. Z8001 asynchronous software handles dataset, reverse flow control, and polling of LACs (via the ICS INTRPT directive); these provide a constant load of about 5 or 3 percent (of ICS2/3 cpu time, per ICS2/3 respectively) for most ICS2/3 asynchronous configurations. Asynchronous characters do not, in general, pass through Z8001 software; this traffic is handled by the IBC.

Overheads for synchronous traffic are on a milliseconds per-frame or per-block basis and are approximately as follows:

	ICS2	ICS3
• PRIMENET/X.25 HDLC Frames	4	2
• PRIMENET/X.25 BSC Frames	4.2	2.1
• SDLC Frames	7	3.7
• RJE Data (blocks)	14	7

IBC Software Performance

The IBC is a very high performance unit capable of several simultaneous operations in one 250-nanosecond-cycle. Its throughput performance tends to be greater than that of other parts of the ICS2 or host software. Average (in microseconds per character) overheads are approximately as follows:

	ICS2	ICS3
• PRIMENET/X.25 HDLC or PRIME/SNA SDLC Data	20	10
• BISYNCHRONOUS PRIMENET/X.25 or RJE Data	28	15
• ASYNCHRONOUS Data	14	9

Line Speeds

Line speed determines the maximum rate at which data can enter or exit the ICS2/3 and also affects queuing delays. The ICS2/3 supports synchronous lines to a maximum of 64Kbps (using V.35) and asynchronous lines to a maximum of 19.2Kbps.

Frame/block queuing for synchronous protocols is provided in the ICS2/3 Z8001 memory as well as in PRIMOS. The maximum number of high speed lines (higher than 19.2Kbps) supportable on an ICS2/3 is a function of the ICS2/3 throughput and also of the ability of the IBC to handle per-character (async) interrupts without data loss.

Protocol mixtures may occur on high speed lines; for instance, one full-duplex BSC line could be used with one full-duplex HDLC or SDLC line. Host software is often the limiting factor in predicting performance of the many permutations of this type of configuration.

Adding asynchronous lines reduces the number of supported synchronous lines. (See CALCULATING PERFORMANCE later in this chapter.)

ICS2: For an ICS2, this latter consideration limits the number of 64Kbps lines on a single ICS2/3 to two for full-duplex HDLC or SDLC (four for half-duplex SDLC), two for full-duplex BSC, or four for half-duplex BSC (RJE). The Z8001 SDLC software further reduces the number of supported high speed lines to two full-duplex or half-duplex, even though the IBC can handle more lines.

ICS3: For an ICS3, the above consideration limits the number of 64Kbps lines, on a single ICS3, to three for FDX, HDLC, or BSC-framed PRIMENET/X.25 (the maximum that may be supported on a single Prime system), to four for FDX or HDX SDLC (the maximum), and to four FDX or eight HDX RJE lines.

On a single ICS2/3, slot priorities are in the order (high to low) 7 through 0, then 15 through 8 (7 through 0, then 17 through 10 on the 16-LAC ICS3 card cage). See LAC Priority in Chapter 5. Therefore, in general, higher speed synchronous lines should be placed in higher numbered LAC card cage slots within an individual LAC card cage. (Refer to LAC Slot Addressing and Figures 5-10 through 5-15 in Chapter 5.) Sync line priority order (high to low) on a given LAC is 1, 0. Asynchronous lines should also be ordered with higher speed lines in higher numbered slots. Async lines have the priority order, on a given LAC (higher to lower) of 1, 0, 3, 2. High speed bisynchronous-framed lines should be placed in slots of higher priority than those of equivalent speed HDLC or SDLC lines. Block mode asynchronous lines should be placed in slots of higher priority than those of synchronous lines, and character mode lines should be placed in slots of lower priority than those of synchronous lines.

Effects of ICS2/3 Overload

HDLC and SDLC are the most "forgiving" of the protocols. The reduction in line utilization (caused by limitation of processing power in the Z8001 code) does not otherwise affect PRIMENET/X.25 or PRIME/SNA communications. The same comments are true for bisynchronous-framed PRIMENET/X.25. Bisynchronous-framed RJE is much more sensitive to load than are other protocols, because of short timeout values associated with the protocol.

It is very unlikely that lack of processing power in the ICS2 will disable an RJE transmission. If overload of an ICS2 running RJE becomes the "final straw" for an already overloaded low-end Prime system, an acknowledgement onto the line within the stringent timeout interval could fail. In this case, the RJE line is disabled, and an error message is logged (providing RJE logging is enabled).

As described earlier, asynchronous protocols use the IBC almost entirely, and synchronous protocols mainly use the Z8001. The result of this is that mixtures of asynchronous and synchronous communications provide a good cost-effective load on the ICS2. Excessive load on the IBC could cause loss of asynchronous data and result in underrun on asynchronous receive. Note that it is extremely unlikely that the IBC will ever run out of power even with a full load (64) of asynchronous terminals at 9.6Kbps. (The IBC gives precedence to asynchronous data input; so the IBC can theoretically support an input data rate of over 65K characters per second.) Asynchronous performance is usually determined in practice by Prime host software, which is somewhat slower than the IBC.

CALCULATING PERFORMANCE

Assumptions to be used follow:

- Assume 80 percent availability for Z8001 when an asynchronous protocol is downline-loaded. Use 90 percent if asynchronous protocol is not present. Calculate on average load figures rather than peaks.
- Assume 90 percent availability for IBC. Calculate on peak load figures, not averages.
- Assume a limit of 300 or 600 (for ICS2 or ICS3) frames per second across the backplane for PRIMENET/X.25 applications per ICS2/3. Only data from I (information) frames travel across the backplane when using PRIME/SNA. A Prime 2550 machine supports approximately 600 frames per second; a Prime 2250 machine supports approximately 240 frames per second.

Example

Assume that a Prime 2250 has one ICS2, 32 (9.6Kbps) asynchronous terminals, and one 9.6Kbps connection to TELENET or another Prime CPU. Eight terminals might be doing block mode input at any one time; 80 percent of the remainder might perform keyboard input/echoing (as in EMACS) with a screen refresh or display rate of one per minute.

Calculations

RJE: These calculations are based on average figures. A 4800bps line at 80 percent utilization with 8-bit characters gives 480 characters per sec (cps). The line is half duplex. IBC load equals 480cps x 0.000028 secs/char x 100 = 1.3 percent. As RJE uses a polling protocol, there must be an ACK or POLL for each data block. Assume that an ACK or POLL contains about 12 characters and that each data block contains 90 characters, including 80 characters of data. The resultant backplane traffic will be approximately 480/(90+12) pairs of blocks per second (or nine blocks per second). The Z8001 overhead, under these conditions, will be 9 blocks/sec x 0.014 sec x 100 = 13 percent.

PRIMENET/X.25 — HDLC: These calculations are based on average figures. A full-duplex 9600bps line at 80 percent utilization gives a total of 1920 characters per second (960 characters in each direction). IBC load equals 1920cps x 0.00002 sec x 100 = 3.8 percent.

As the line becomes heavily loaded, assume three full I-frames (140 characters long including, 128 characters of data) to each control frame (approximately 10 characters long). The resultant backplane traffic is approximately 1920/((3 x 140) + 10) quads of frames per second (or 18 frames per second). The Z8001 overhead, under these conditions, is 18 frames/sec x 0.004 sec x 100 = 7.2 percent. The frame length is configurable and, the commonly used length of 128 characters has been used in this calculation.

ASYNC: These calculations are based on worst-case figures. Block mode terminals, if all inputting at once, would contribute 8 x 960 = 7680 characters per second. If the 80 percent operational terminals each did a simultaneous screen refresh, this would generate 20 x 960 = 19,200 characters per second. (Note that it is extremely unlikely that host software could actually refresh 20 screens within the one second required to generate these loading figures.) Total IBC load is thus 26,880 x 0.000014 x 100 = 37.6 percent.

Totals:

IBC = 37.6 + 3.8 + 1.3 = 42.7 percent (allowed 90 percent)

Z8001 = 7.2 + 13 = 20.2 percent (allowed 80 percent)

ICS2-Prime Interface = 18 + 9
= 27 frames/sec (allowed 300 frames/sec)

These figures show that this ICS2 should operate comfortably and could still accept further loading, and that no real overhead reduction or increase in processing speed would be derived from adding another ICS2 controller board. At the line speeds used in this example, there is no particular advantage in prioritizing LAC positioning in the card cage other than LACs associated with block mode terminals.

If the RJE line or X.25 line runs at 48Kbps or faster, then their LACs should be placed in one of the higher-numbered slots.

This example does not take account of the performance of Prime host software. In general, PRIMENET/X.25, RJE, and SNA software function satisfactorily with any allowable ICS2 configuration, though full synchronous line utilization may not be achievable in every situation.

Asynchronous reverse flow control should be used when there is a medium or high asynchronous traffic load on the ICS2, that is, many terminals and/or block mode applications. Asynchronous reverse flow control is not available where BSC and ASYNC protocols are selected together. (See Table 5-1 in Chapter 5 for details.)

Calculation Summary

The calculations shown above may be abbreviated to the following algorithms for synchronous protocols. All line speeds are in bits per second (bps).

ICS2

RJE (HDX)

IBC load = (line speed x 0.1) x 0.0028 percent

Z8001 overhead = ((line speed x 0.1) / 51) x 1.4 percent

HDLC — X.25 (FDX at 80 percent utilization)

IBC load = (line speed x 0.2) x 0.002 percent

Z8001 overhead = (line speed x 0.32) / ((3 x frame length) + 46) percent

BSC — X.25 (FDX at 80 percent utilization)

IBC load = (line speed x 0.1) x 0.0028 percent

Z8001 overhead = (line speed x 0.336) / ((3 x frame length) + 46) percent

SDLC (FDX at 80 percent utilization)

IBC load = (line speed x 0.2) x 0.002 percent

Z8001 overhead = (line speed x 0.56) / ((3 x frame length) + 46) percent

Note

Line speed refers to the speed in bits per second. For example, 1200bps FDX and HDX lines both have a line speed of 1200bps.

ICS3

RJE (HDX)

IBC load = (line speed x 0.1) x 0.0015 percent

Z8001 overhead = ((line speed x 0.1) / 51) x 0.7 percent

HDLC — X.25 (FDX at 80 percent utilization)

IBC load = (line speed x 0.2) x 0.002 percent

Z8001 overhead = (line speed x 0.16) / ((3 x frame length) + 46) percent

BSC — X.25 (FDX at 80 percent utilization)

IBC load = (line speed x 0.1) x 0.0015 percent

Z8001 overhead = (line speed x 0.168) / ((3 x frame length) + 46) percent

SDLC (FDX at 80 percent utilization)

IBC load = (line speed x 0.2) x 0.001 percent

Z8001 overhead = (line speed x 0.296) / ((3 x frame length) + 46) percent

RESTRICTIONS

This section lists the current restrictions in PRIMENET/X.25, RJE, SDLC, and asynchronous operations.

PRIMENET/X.25 Restrictions

The following is a list of the current restrictions on ICS2/3 PRIMENET/X.25 operations.

- Two synchronous controllers can be configured with PRIMOS. These controllers can be any combination of ICS1/2/3, or MDLC. The synchronous controllers and lines are configured and defined by SYNC directives in the CONFIG file. (See Chapter 5.) More controllers may be present, supplying ASYNC or SDLC lines.
- PRIMOS can support three PRIMENET/X.25 synchronous lines.
- At Rev. 20.1, ICS2/3 PRIMENET support includes full-duplex (FDX), but not half-duplex (HDX) PRIMENET support.
- Transparent ASCII is NOT supported (same as for the MDLC and ICS1).
- At least four SYN characters must be received and recognized between BSC frames. This is not a problem with Prime machines using ICS1, ICS2, or MDLC controllers. However, you should check the number of SYNs when you make connections to non-Prime equipment.

RJE Restrictions

The following is a list of the current restrictions on ICS2/3 RJE operations.

- Two synchronous controllers can be configured with PRIMOS. These controllers can be any combination of ICS1/2/3, or MDLC. The synchronous controllers and lines are configured and defined by SYNC directives in the CONFIG file. (See Chapter 5.)

- PRIMOS can support eight synchronous assignable lines (logical lines 0-7). (A maximum of eight of these may be used for RJE and a maximum of three for PRIMENET/X.25)

- Software loopback (SDF DATASET option) is an MDLC hardware feature not implemented on the ICS2/3.

- Transparent ASCII is not supported (same as for the MDLC and ICS1).

- At least four SYN characters must be received between messages. (This is unlikely to be a problem with IBM, Prime, and most common IBM-like machines running 2780/3780/HASP emulators, because these use more than four SYNs.)

- At least two SYN characters must be received immediately following an ITB.CRC sequence.

- When you transmit ASCII, there are nine trailing PAD bits. (IBM specifies at least eight.) This difference can confuse line monitoring equipment if an out-of-sync spec of 16 or more PAD bits is specified.

SDLC Restrictions

The following are restrictions on a Prime system running PRIME/SNA (SDLC).

- There can be a maximum of one ICS2/3 running SDLC on a system.
- There can be a maximum of four SDLC lines configured and/or active, per ICS2/3. The SNA Configuration file will allow definition of up to four lines; use multiple configuration files to cater for different combinations of four connections.
- There can be a maximum of eight Remote Systems configured and concurently active.
- There can be up to 48 active LUs on Prime 9950 and 9955 systems.
- There can be up to 32 active LUs on Prime 9750 through 2550 systems.
- There can be up to 16 active LUs on a Prime 2250.
- There can be a maximum of two lines running at 64K bps.
- The sum of all SDLC line speeds must be no larger than 147.2K bps.
- Avoid having many busy asynchronous lines and SDLC lines on the same ICS2/3. If this is a problem, share the asynchronous lines among other ICS1/2/3s.

Asynchronous Restrictions

ICS2: The ICS2 is only available with the 16-slot LAC card cage. It can, therefore, support up to a maximum of 64 asynchronous lines if no other protocol is supported.

ICS3: The ICS3 is available with two sizes of LAC card cage, 8-slot and 16-slot. The 8-slot card cage can support up to 32 asynchronous lines, and the 16-slot card cage can support up to 64 asychronous lines.

Restrictions on the Protocol Combinations Supported

Full details of the valid protocol combinations are given in Chapter 5, Tables 5-1 and 5-2, for the ICS2 and ICS3 respectively.

ICS2: The following combinations of protocols are supported on a single ICS2 at Revision 20.1 of PRIMOS.

- Asynchronous
- SNA (SDLC)
- HDLC-framed X.25
- 2780/3780/HASP
- BSC-framed X.25
- Asynchronous and HDLC-framed X.25 and SNA (SDLC)
- 2780/3780/HASP and BSC-framed X.25 and Asynchronous
- 2780/3780/HASP and BSC-framed X.25 and HDLC-framed X.25 and SNA (SDLC)

When BISYNC (BSCX25 or BSCRJE) and ASYNC protocols are used together, there is no asynchronous Reverse Flow Control. See Notes under Table 5-1 for more detail.

ICS3: The ICS3 controller is available with 256K, 512K, or 1024K byte RAM. The 256K byte version can support all protocol combinations except the two largest ones, ASYNC_SDLC_HDLC_BSCX25_BSCRJE and SDLC_BSCX25_BSCRJE. The 512K byte and 1024 byte RAM versions can support any combination of the allowable protocols, including all protocols together. The allowable protocols are as shown above under ICS2.

7
Diagnostics

INTRODUCTION

This chapter details the built-in diagnostics that give visual indication (on the system console) of any faults found during the startup of the ICS2. The built-in diagnostics are active during two stages of the startup procedure.

- Stage 1 is from the PROM code, and it checks the memory and memory size. (256K memory, or larger, is required for SDLC/HDLC or BSC-related protocols. Asynchronous-only controllers may have 128K or 256K memory.)
- Stage 2 is a runtime, self-verify test, which is performed at the completion of the downline load.

STAGE 1 DIAGNOSTIC

Stage 1 of the built-in diagnostics checks the memory and memory size prior to booting the Z8001 memory. If the memory is good, and of the correct size, a message (showing the verified memory size, 256/512/1024K bytes) is displayed on the system console. For example:

```
...Good status returned to host

...Memory verified, size = 256K bytes
```

Failure of Stage 1 diagnostic produces a message containing two hexadecimal status words on the system console. The first word is 8001, the second is broken down into 16 bits, as shown in Table 7-1 (ICS2) and Table 7-2 (ICS3). Bit 0 is the right end bit, and bit 15 is the left end bit.

If these status words indicate a faulty ICS2/3, contact your local Customer Service Engineer.

Table 7-1
ICS2 Stage 1 Diagnostic — Second Status Word

Bit(s)	Description
15	0 = memory test passed 1 = memory test failed
14-13	Microprocessor memory size 10 = 128K bytes 11 = 256K bytes
12	Zero
11-8	Error codes, valid if bit 15 = 1: 0 = no good memory found during memory sizing 1 = data error during offset memory test 2 = data error during complemented offset memory test 3 = data error during byte memory test 4 = parity error during memory test 5 = bad parity RAM or parity generator detected 6 = bad parity checker or parity flop detected 7 = reset cycle failing
7-6	ICS type, 01 = ICS2 10 = ICS3
5-0	ICS device ID in octal

Table 7-2
ICS3 Stage 1 Diagnostic — Second Status Word

Bit(s)	Description
15	0 = memory test passed 1 = memory test failed
14-13	Microprocessor memory size: 10 = 512K bytes 11 = 1024K bytes
12	Zero
11-8	Error codes, valid if bit 15 = 1: 0 = data error during data checkerboard test 1 = data error during refresh test 2 = checkbit error during checkbit checkerboard 3 = single bit error detected during checkbit 4 = data error detected during byte write test 5 = address line test failed 6 = data error during bit shift test 7 = data error during address offset test 8 = EDAC failed to correct a single bit error 9 = single bit error flop should have been set A = syndrome bits do not indicate correct bit B = single bit error interrupt test failed C = EDAC multiple bit error test failed
7-6	ICS type, 01 = ICS2 10 = ICS3
5-0	ICS device ID in octal

STAGE 2 DIAGNOSTIC

The Stage 2 diagnostic is a runtime, self-verify test. This test is carried out after the downline load sequence has ended.

During the test, the following is displayed on the system console:

```
... Performing run-time self-verify.
```

If the Stage 2 diagnostic is successful, the following is displayed:

```
... Run-time self-verify successful.
```

Failure of the Stage 2 diagnostic results in the display of a message, containing two diagnostic status words, on the system console. The first word is 8001 (Hex). The second word is a hexadecimal number indicating the failed test as described in Table 7-3.

Table 7-3
ICS2/3 Stage 2 Diagnostic — Second Status Word

Word (Hex)	Description
001	The ICS2 controller is not up to Rev. F (which supports the extended NAD/IM field for the IBC).
002	The NVI register failed to initialize correctly.
003	The interrupt masks in the NVI register do not function correctly.
004	Unable to access the CTC's internal register.
005	Unable to get CTC 0 to count down.
006	The CTC is unable to issue an interrupt request.
007	The NVI sequence for CTC 0 failed to work correctly.
008	The segment trap sequence for CTC 2 failed to work correctly.
00D	The IBC WCS does not function correctly.
00F	The IBC line file does not function correctly.
010	IBC vectored interrupts to the Z8001 do not work correctly.
013	The IBC map RAM does not function correctly.
014	The IBC condition code bits do not work correctly.
101	Phase 1 of IBC microcode boot did not complete correctly.
102	Phase 2 of IBC microcode boot did not complete successfully.
103	LAC card cage unpowered or disconnected, or other hardware problem.
104	Invalid interrupt or trap to Z8001 during self-verify (unimplemented instruction trap, privileged instruction trap, system call trap, segment trap, NMI interrupt, NVI interrupt, unused VI interrupt). Indicates a hardware problem.

Further information on the messages displayed during stage 1 and stage 2 diagnostics can be found in Appendix B, ERROR AND STATUS MESSAGES.

APPENDIXES

A

ICS2/3 Cables and Connectors

INTRODUCTION

This appendix contains specific information on the various cables used in ICS2/3 configurations. Details include the cable number, length, and where to use the cable.

Also included are details of some ICS2/3 LAC-to-device cable connectors and their pin assignments.

This appendix is divided into two sections:

- Controller-to-LAC Card Cage Cables
- LAC-to-Device Cables

Use the diagrams in each section to find the cable numbers, and then find more detail for that cable at the end of the section.

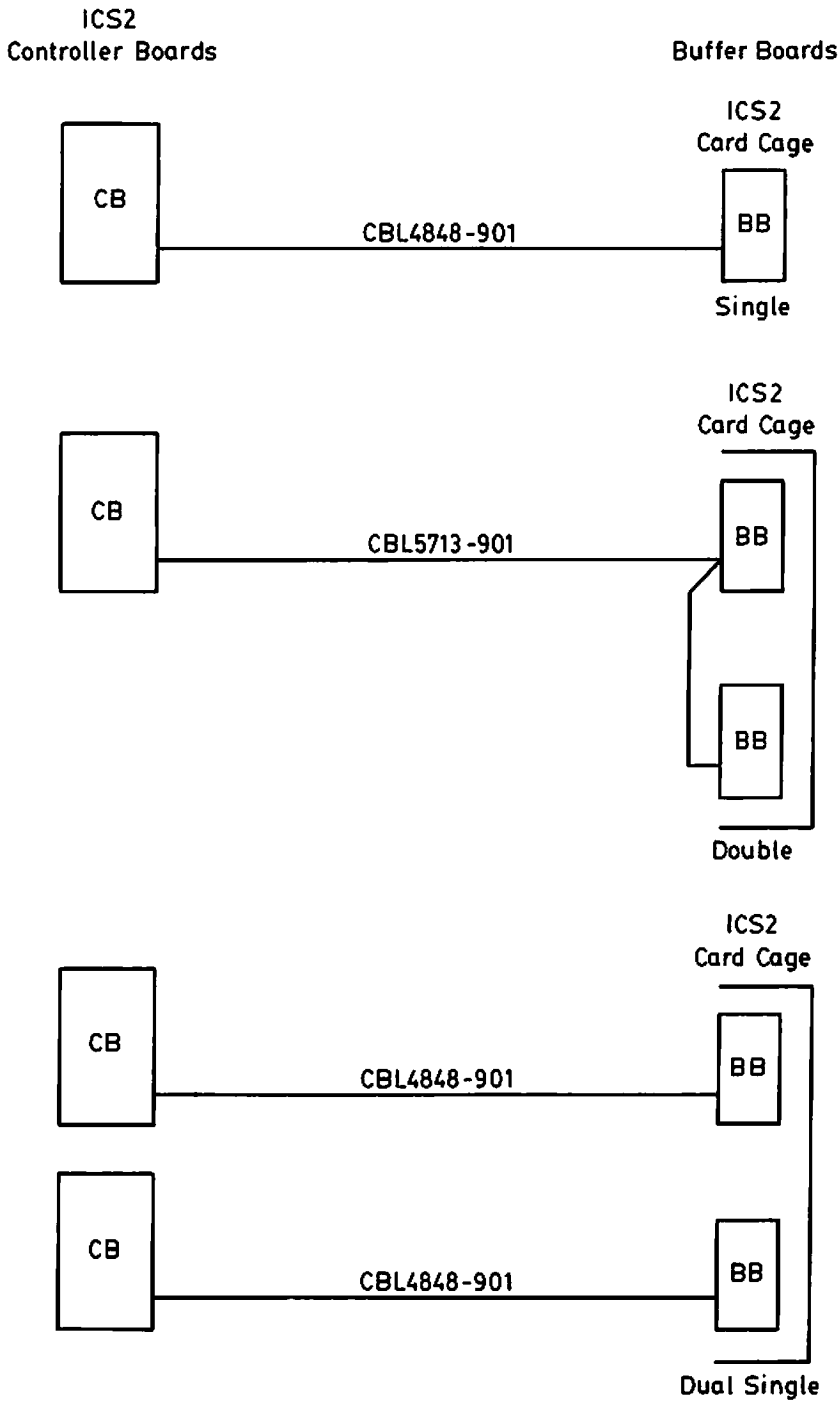
CONTROLLER TO LAC CARD CAGE CABLES

The cables connect the controller board to the LAC card cages either directly, or via the system bulkhead and peripheral bulkhead. The connectors on the ICS2 and ICS3 controller boards are different; therefore, their cables are different. This section provides cable numbers for the various configurations. Pin connection details are not provided.

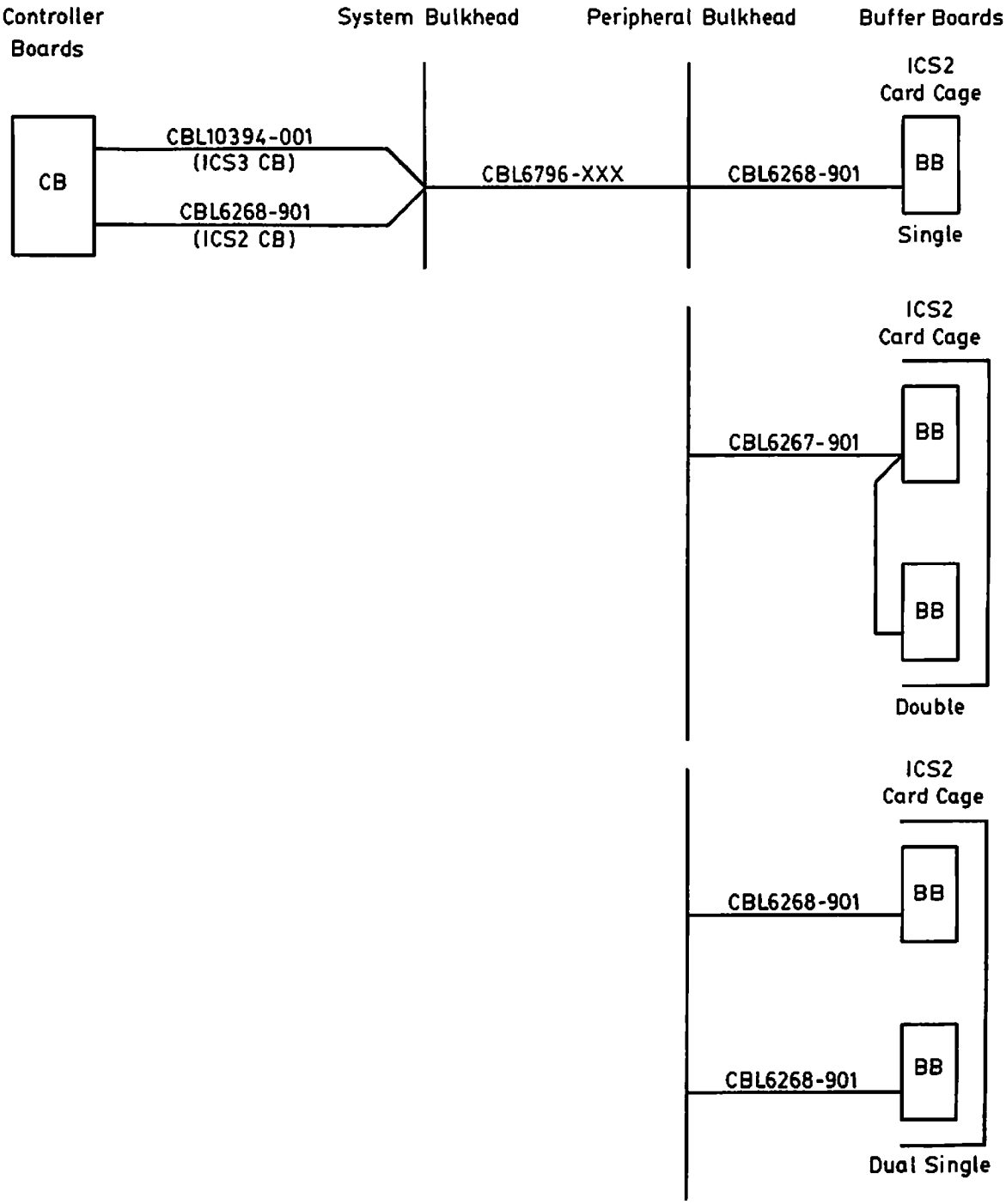
> **Caution**
>
> Do not connect or disconnect LACs to or from LAC card cages, or LAC cables to or from LACs, with the power on. Damage may result.

ICS2 LAC Card Cages

Figures A-1 and A-2 show the cables required for single, double, and dual single-configurations using the ICS2 card cage. Use these figures to determine the required cable numbers. The controller board may be either ICS2 or ICS3. If the ICS3 controller board is used with the ICS2 card cage, the cables must go via the bulkheads.



Controller Board to ICS2 LAC Card Cage, Direct
Figure A-1



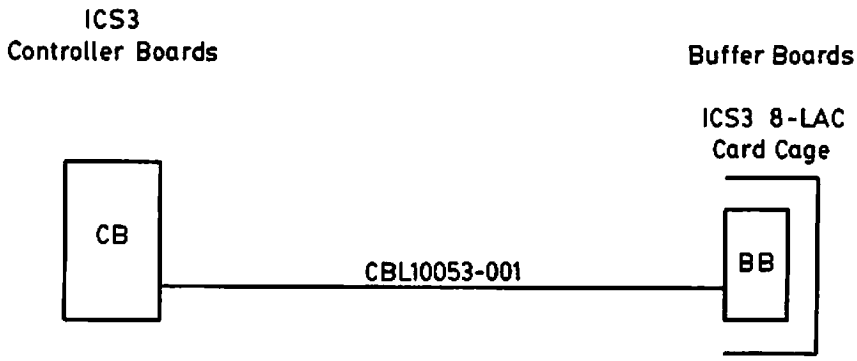
Controller Board to ICS2 LAC Card Cage, Using Bulkheads
Figure A-2

ICS3 LAC Card Cages

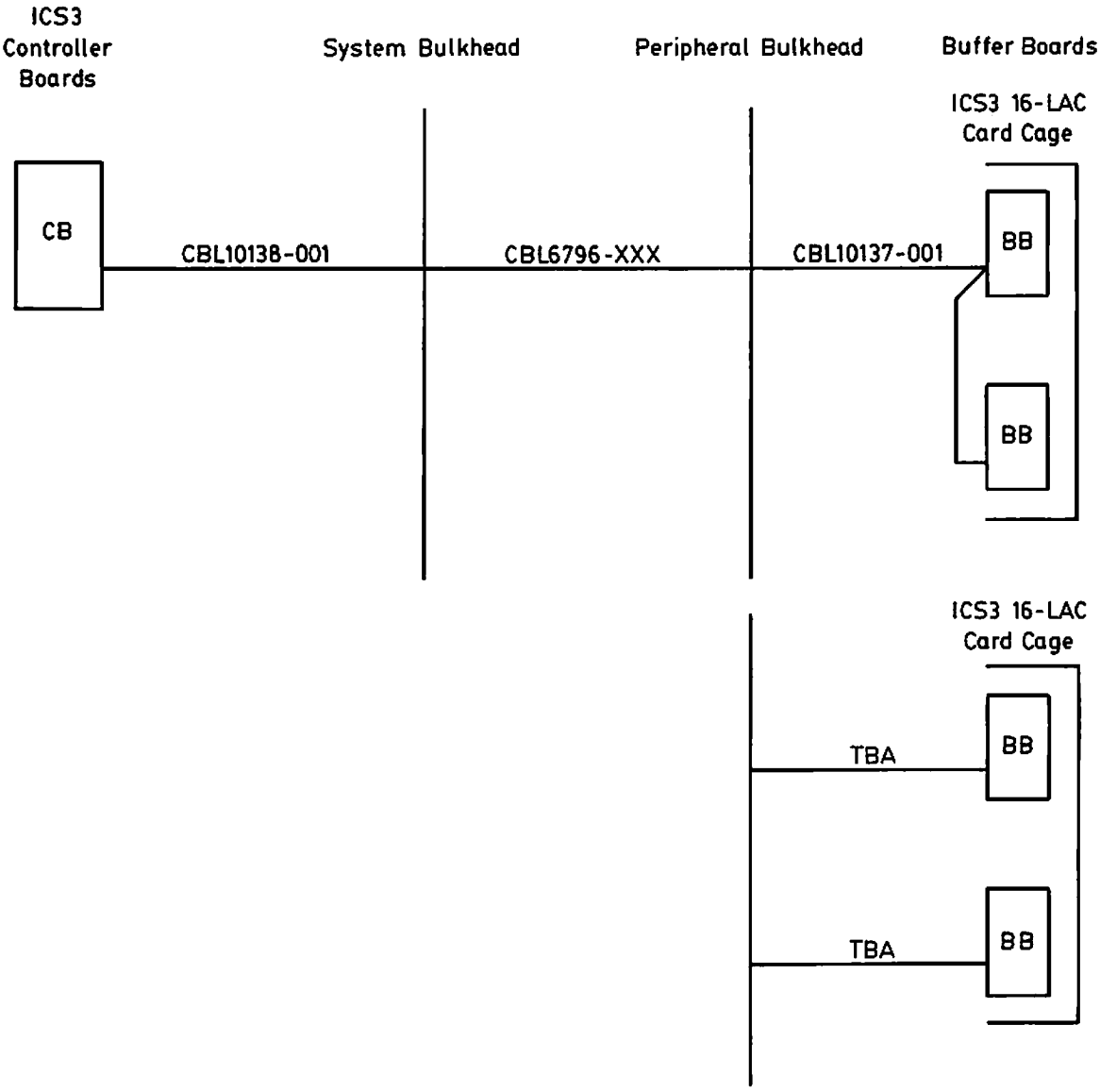
Figures A-3 and A-4 show the cables required for connecting the ICS3 controller board to the ICS3 LAC card cages. Three types of configuration are possible; one for the 8-LAC card cage, and two for the 16-LAC card cage (double, and dual single). Use these figures to find the required cable numbers for your configuration.

Figure A-3 shows the direct connection between the ICS3 controller board and the 8-LAC ICS3 card cage. This is a short, inside CPU cabinet cable, used only for this card cage.

Figure A-4 shows the connection (via system bulkhead and peripheral bulkhead) between the ICS3 controller board and the ICS3 LAC card cages.



ICS3 Controller Board-to-ICS3 LAC Card Cages, Direct
Figure A-3



ICS3 Controller Board-to-ICS3 LAC Card Cages, Using Bulkheads
Figure A-4

Cable Details

Cable CBL6796-XXX: This cable is used to connect the system bulkhead to the peripheral bulkhead, where XXX can be one of two possible dash numbers used to specify cable length. The available cable lengths are:

Cable Number	Length (feet)
CBL6796-001	10.00
CBL6796-002	30.00

Cable CBL10053-001: This cable connects the ICS3 controller board to the 8-LAC card cage and is 3.67 feet long.

Cable CBL10137-001: This cable connects the 16-LAC card cage to the peripheral bulkhead and is 4.17 feet long. There are two connectors at the card cage end; connect one to each of the two buffer boards.

Cable CBL10138-001: This cable connects the ICS3 controller board to the system bulkhead and is 3 feet long.

Cable CBL10394-001: This cable is used for upgrading ICS2 systems to use the ICS3 controller. The cable connects the ICS3 controller board to the system bulkhead and is 3 feet long.

LAC TO DEVICE CABLES

The ICS2 and ICS3 card cages use the same LAC-to-device cables. The ICS3 LACs use their integral bulkhead, on the edge of the LAC, to connect directly to the device cable. The ICS2 LAC, however, does not have integral bulkheading so it requires an additional cable to connect the LAC to the peripheral bulkhead. Device cables then connect to the peripheral bulkhead.

ICS2 and ICS3 LACs are different sizes and are therefore, not plug-compatible; nor will they fit in the other's card cages. There are three basic types of LACs (in each size); asynchronous, synchronous (V.24), and synchronous (V.35).

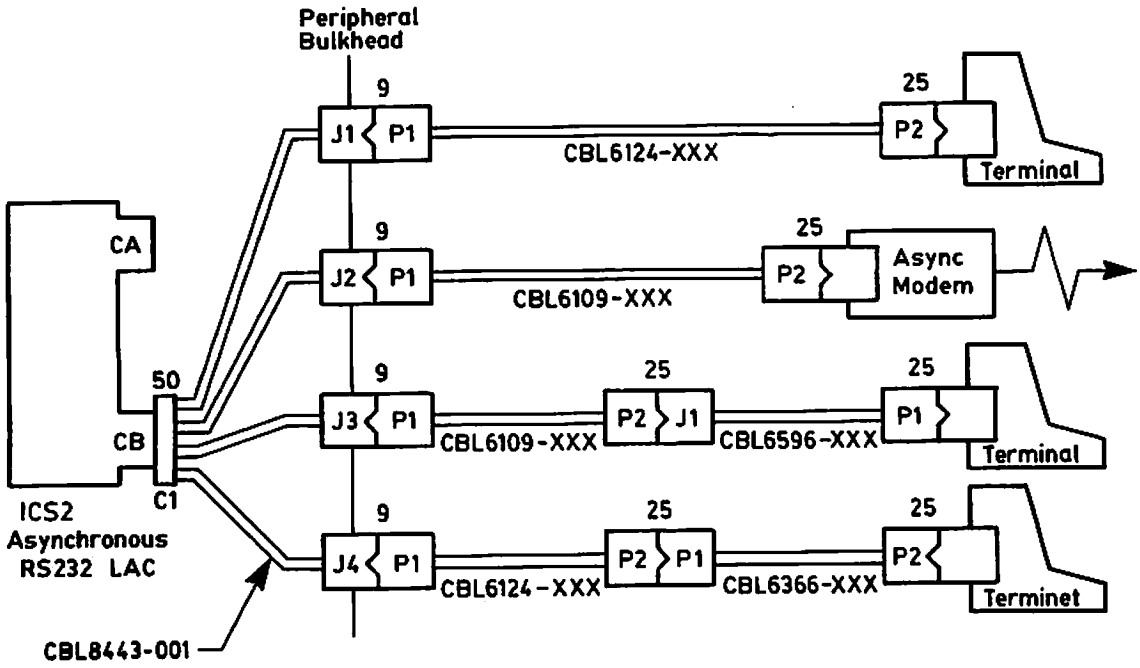
Additional cable detail, including pin connections, appears at the end of this section.

ICS2 LAC Cables

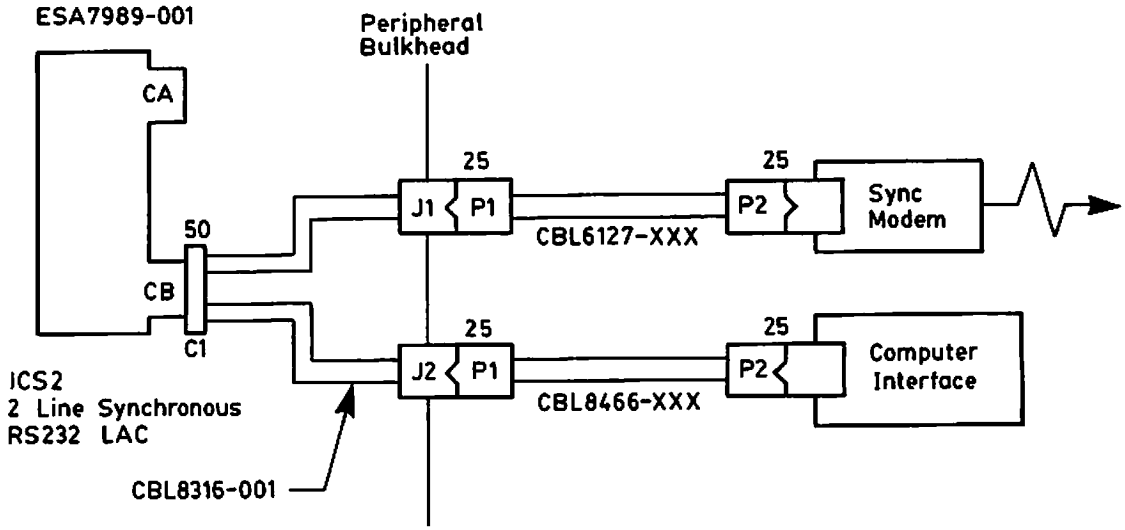
Figures A-5, A-6, and A-7 show the cable connections for the three types of ICS2 LACs.

ICS3 LAC Cables

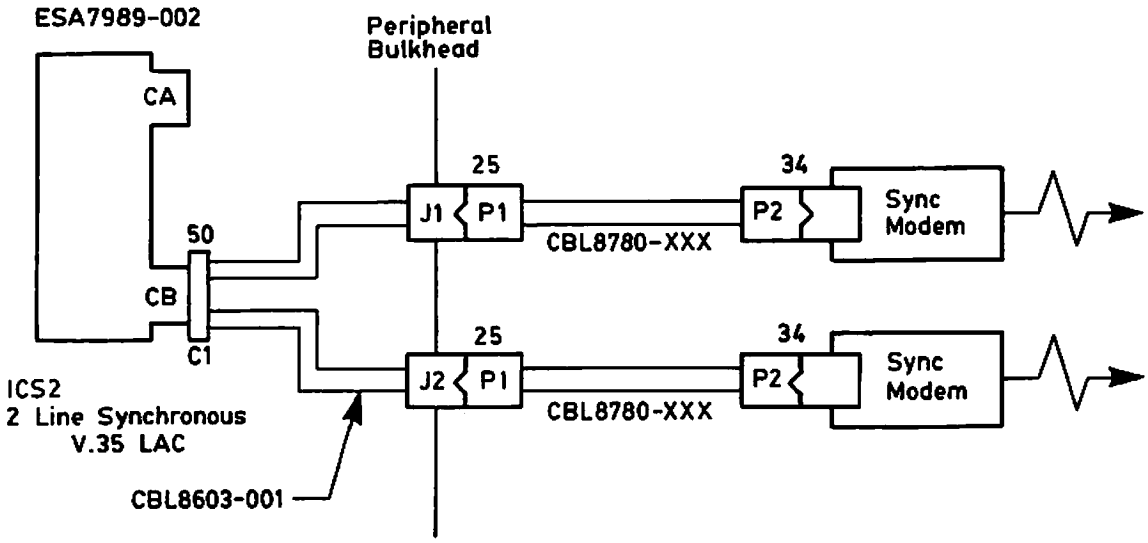
Figures A-8, A-9, and A-10 show the cable connections for the three types of ICS3 LACs.



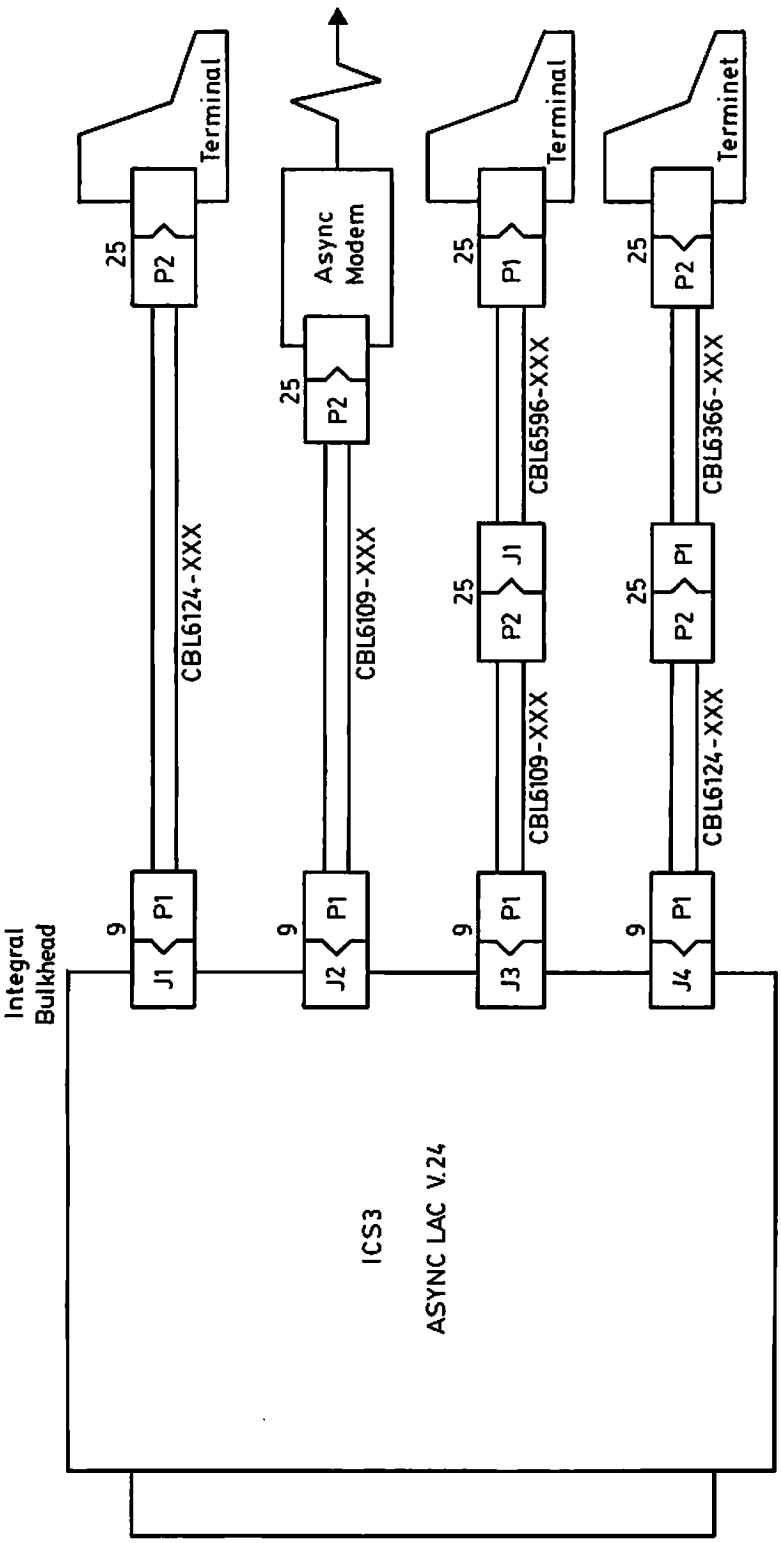
Cable Connections for ICS2 Asynchronous LAC
Figure A-5



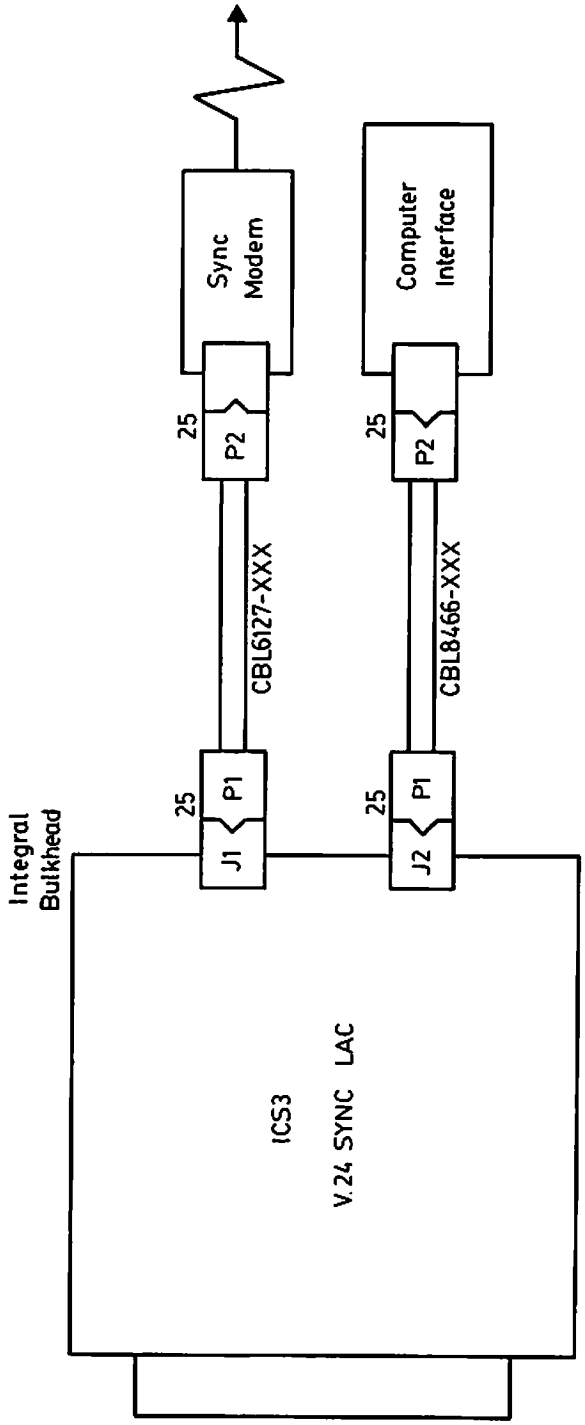
Cable Connections for ICS2 Synchronous (V.24) LACs
Figure A-6



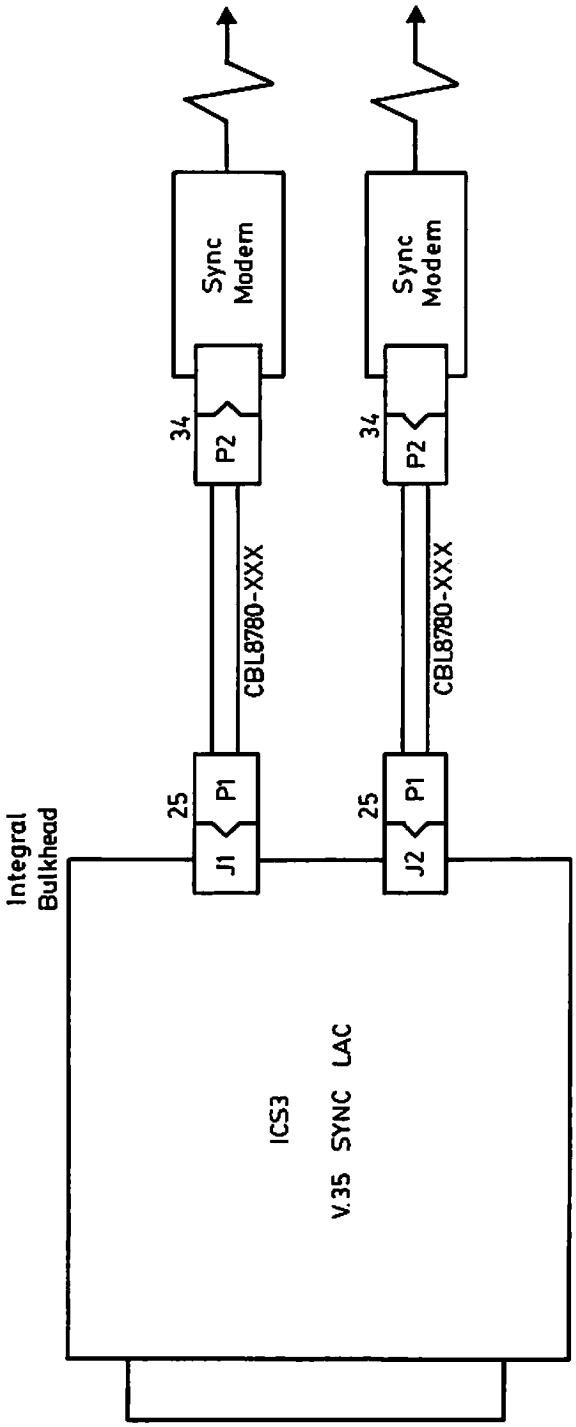
Cable Connections for ICS2 Synchronous (V.35) LACs
Figure A-7



Cable Connections for ICS3 Asynchronous LACs
Figure A-8



Cable Connections for ICS3 Synchronous (V.24) LACs
Figure A-9



Cable Connections for ICS3 Synchronous (V.35) LACs
Figure A-10

Cable Details

Cable CBL8443-001: This cable (formerly CBL6205-901) connects the ICS2 asynchronous LAC connector CB to the peripheral bulkhead. The cable is six feet long and has five connectors, C1, J1, J2, J3, and J4. The C1 connector connects to the 50-pin CB edge connector of the LAC. The J1, J2, J3, and J4 connectors are 9-pin D-type female connector filtered connectors that connect to the peripheral bulkhead. Figure A-5 illustrated the system configuration for this cable and Table A-1 lists the cable connection data.

Table A-1
Connection Data for Cable CBL8443-001 (formerly CBL6205-901)

LAC	Bulkhead	Signal Name
C1-44	J1-2	XTDATA0-
C1-41	J1-3	XRDATA0-
C1-39	J1-4	XRTS0+
C1-31	J1-5	XCTS0+
C1-26	J1-7	GND
C1-47	J1-8	XDCD0+
C1-48	J1-9	XDTR0+
C1-19	J2-2	XTDATA1-
C1-16	J2-3	XRDATA1-
C1-14	J2-4	XRTS1+
C1-06	J2-5	XCTS1+
C1-22	J2-8	XDCD1+
C1-23	J2-9	XDTR1+
C1-27	J3-2	XTDATA2-
C1-28	J3-3	XRDATA2-
C1-29	J3-4	XRTS2+
C1-30	J3-5	XCTS2+
C1-32	J3-7	GND
C1-33	J3-8	XDCD2+
C1-45	J3-9	XDTR2+
C1-02	J4-2	XTDATA3-
C1-03	J4-3	XRDATA3-
C1-04	J4-4	XRTS3+
C1-05	J4-5	XCTS3+
C1-08	J4-8	XDCD3+
C1-20	J4-9	XDTR3+
J1-07	J2-7	GND
J3-07	J4-7	GND

Cable CBL6124-XXX: Direct connection of terminals to the peripheral bulkhead requires cable CBL6124-XXX, where XXX is one of eleven possible dash numbers for the cable lengths below:

Cable Number	Length (meters)
CBL6124-901	.25
CBL6124-902	3.00
CBL6124-903	10.00
CBL6124-904	15.00
CBL6124-905	25.00
CBL6124-906	35.00
CBL6124-907	50.00
CBL6124-908	75.00
CBL6124-909	100.00
CBL6124-910	150.00
CBL6124-911	200.00

Cable CBL6124-XXX has two connectors, P1 and P2. The P1 connector is a 9-pin D-type male connector that connects to the peripheral bulkhead. The P2 connector is a 25-pin D-type male connector that connects to the terminal. Table A-2 lists the pin connections.

Table A-2
Connection Data for Cable CBL6124-XXX

From	To
P1-02	P2-03
P1-03	P2-02
P1-07	P2-07
P1-08	P2-08
P1-09	P2-06
P1-04	P1-05
P2-04	P2-05
P2-08	P2-20

Cable CBL6109-XXX: Direct connection of asynchronous modems to the peripheral bulkhead is supported using CBL6109-XXX, where XXX can be one of six possible dash numbers used to specify cable length. The available cable lengths are:

Cable Number	Length (meters)
CBL6109-901	19.80
CBL6109-902	12.20
CBL6109-903	7.60
CBL6109-904	3.00
CBL6109-905	30.50
CBL6109-906	15.00

Cable CBL6109-XXX has two connectors, P1 and P2. The P1 connector is a 9-pin D-type male connector that connects to the bulkhead. The P2 connector is a 25-pin D-type male connector that connects to an asynchronous modem. Table A-3 lists the pin connections.

Table A-3
Connection Data for Cable CBL6109-XXX

From	To
P1-02	P2-02
P1-07	P2-07
P1-03	P2-03
P1-07	P2-07
P1-04	P2-04
P1-07	P2-07
P1-05	P2-05
P1-07	P2-07
P1-08	P2-08
P1-07	P2-07
P1-09	P2-20
P1-07	P2-07

Cable CBL6596-XXX: This cable connects terminals to asynchronous modem cables (CBL6109-XXX). The cable is a female-to-male adapter cable with four possible dash numbers specifying cable length. The available cable lengths are:

Cable Number	Length (meters)
CBL6596-901	.25
CBL6596-902	3.00
CBL6596-903	10.00
CBL6596-904	15.00

Cable CBL6596-XXX has two connectors, J1 and P2. The J1 connector is a 25-pin D-type female connector that connects to the P2 connector of CBL6109-XXX. The P2 connector is a 25-pin D-type male connector that that connects to a terminal. Table A-4 lists the pin connections.

Table A-4
Connection Data for Cable CBL6596-XXX

From	To
J1-02	P2-03
J1-03	P2-02
J1-07	P2-07
J1-08	P2-08
J1-20	P2-06
J1-04	J1-05
P2-04	P2-05
P2-08	P2-20

Cable CBL6366-XXX: Connection of Terminet hard copy terminals is supported using CBL6366-XXX, where XXX can be one of two possible dash numbers used to specify the connector screw type. The two options are:

Cable Number	Screw Type
CBL6366-001	#4-40
CBL6366-002	3.00 mm

Cable CBL6366-XXX has two connectors, P1 and P2. The P1 connector is a 25-pin D-type female connector that connects to the P2 connector of CBL6124-XXX. The P2 connector is a 25-pin D-type female connector that connects to a Terminet hard copy terminal. Table A-5 lists the pin connections.

Table A-5
Connection Data for Cable CBL6366-XXX

From	To
P1-02	P2-03
P1-03	P2-02
P1-07	P2-07
P1-08	P2-08
P1-09	P2-06
P1-04	P1-05
P2-04	P2-05
P2-08	P2-20

Cable CBL8316-001: This cable connects the CB connector of the SYNC (V.24) LAC to the peripheral bulkhead. This cable is six feet long and has three connectors: C1, J1, and J2. The C1 connector connects to the 50-pin CB edge connector of the LAC. The J1 and J2 connectors are 25-pin D-type female connectors that connect to the peripheral bulkhead. Figure A-6 illustrated the system configuration for this cable, and Table A-6 lists the pin connections.

Table A-6
Connection Data for Cable CBL8316-001

From	To	Signal
C1-01/07	J1-7	GND
C1-02	J1-2	TDATA0-
C1-03	J1-3	RDATA0-
C1-04	J1-4	XRTS0-
C1-05	J1-6	XDSR0-
C1-06	J1-8	XDCD0-
C1-08	J1-5	XCTS0-
C1-09		RDATA0
C1-10	J1-25	TRCLOCK0+
C1-11		RRCLOCK0
C1-12	J1-24	TTCLOCK0+
C1-13		TDATA0+
C1-14	J2-4	XRTS1-
C1-15	J1-15	RTCLOCK0+
C1-16		RTCLOCK0-
C1-17	J1-17	RRCLOCK0+
C1-18	J1-11	TCLKIN0+
C1-19		CABLE10N
C1-20	J1-19	XDSC0-
C1-21		TTCLOCK0-
C1-22	J1-22	XR10-
C1-23	J2-19	XDSC1-
C1-24	J1-21	UNUSED
C1-25	J1-18	RCLKIN0+
C1-26/32	J2-7	GND
C1-27	J2-2	TDATA1-
C1-28	J2-3	RDATA-
C1-29	J1-20	XDTR0-
C1-30	J2-6	XDSR1-
C1-31	J2-8	XDCD1-
C1-33	J2-5	XCTS1-
C1-34		RDATA1+
C1-35	J2-25	TRCLOCK1+
C1-36		RRCLOCK1-
C1-37	J2-24	TTCLOCK1+
C1-38		TDATA1+
C1-39	J2-20	XDTR1-
C1-40	J2-15	RTCLOCK1+
C1-41		RTCLOCK1-
C1-42	J2-17	RRCLOCK1+
C1-43	J2-11	TCLKIN1+
C1-44		CABLE00N
C1-45	J1-23	XSPS0-
C1-46		TTCLOCK1-
C1-47	J2-22	XRI1
C1-48	J2-23	XSPS1-
C1-49	J2-21	UNUSED
C1-50	J2-18	RCLKIN1+

Note

Pins C1-01, C1-19, and C1-44 are tied together. This ties CABLE1ON- (C1-19) and CABLE00N- (C1-44) to ground (C1-01).

The term "unused" means the connector pin is available but currently not used. The term "reserved", however, means that the connector pin is unused and also unavailable for any future use.

Cable CBL6127-XXX: Direct connection of synchronous modems (V.24) to the peripheral bulkhead is supported using CBL6127-XXX, where XXX can be one of six possible dash numbers specifying cable length. The available cable lengths are:

Cable Number	Length (meters)
CBL6127-901	3.00
CBL6127-902	7.60
CBL6127-903	12.20
CBL6127-904	19.80
CBL6127-905	30.50
CBL6127-906	15.00

Cable CBL6127-XXX has two connectors, P1 and P2. The P1 connector is a 25-pin D-type male connector that connects to the bulkhead. The P2 connector is a 25-pin D-type male connector that connects to a synchronous modem. Table A-7 lists the pin connections.

Table A-7
Connector Data for Cable CBL6127-XXX

From	To	From	To
P1-01	P2-01	P1-15	P2-15
P1-07	P2-07	P1-07	P2-07
P1-02	P2-02	P1-17	P2-17
P1-07	P2-07	P1-07	P2-07
P1-03	P2-03	P1-20	P2-20
P1-07	P2-07	P1-07	P2-07
P1-04	P2-04	P1-21	P2-21
P1-07	P2-07	P1-07	P2-07
P1-05	P2-05	P1-22	P2-22
P1-07	P2-07	P1-07	P2-07
P1-06	P2-06	P1-23	P2-23
P1-07	P2-07	P1-07	P2-07
P1-08	P2-08	P1-24	P2-24
P1-07	P2-07	P1-07	P2-07
P1-14	P2-14		
P1-07	P2-07		

Cable CBL8466-XXX: This cable will connect the peripheral bulkhead to a computer interface. Contact your local Customer Field Service engineer about its availability.

Cable CBL8603-001: This CB connector of the synchronous V.35 LAC is connected to the peripheral bulkhead by cable CBL8603-001. This cable is six feet long and has three connectors: C1, J1, and J2. The C1 connector connects to the 50-pin CB edge connector of the LAC. The J1 and J2 connectors are 25-pin D-type female connectors that connect to the peripheral bulkhead. Figure A-7 illustrated the system configuration for this cable, and Table A-8 lists the pin connections.

Table A-8
Connection Data for Cable CBL8603-001

From	To	Signal Name
C1-01/07	J1-7	GND
C1-02	J1-2	TDATA0-
C1-03	J1-3	RDATA0-
C1-04	J1-4	XRTS0-
C1-05	J1-6	XDSR0-
C1-06	J1-8	XDCD0-
C1-08	J1-5	XCTS0-
C1-09	J1-16	RDATA0+
C1-10		TRCLOCK0+
C1-11	J1-25	RRCLOCK0-
C1-12	J1-24	TTCLOCK0+
C1-13	J1-14	TDATA0+
C1-14	J2-4	XRTS1-
C1-15	J1-15	RTCLOCK0+
C1-16	J1-21	RTCLOCK0-
C1-17	J1-17	RRCLOCK0+
C1-18	J1-11	TCLKIN0+
C1-19		CABLE10N
C1-20	J1-19	XDSC0-
C1-21	J1-18	TTCLOCK0-
C1-22	J1-22	XR10-
C1-23	J2-19	XDSC1-
C1-24		UNUSED
C1-25		RCLKIN0+
C1-26/32	J2-7	GND
C1-27	J2-2	TDATA1-
C1-28	J2-3	RDATA1-
C1-29	J1-20	XDTR0-
C1-30	J2-6	XDSR1-
C1-31	J2-8	XDCD1-
C1-33	J2-5	XCTS1-
C1-34	J2-16	RDATA1+
C1-35		TRCLOCK1+
C1-36	J2-25	RRCLOCK1-
C1-37	J2-24	TTCLOCK1+
C1-38	J2-14	TDATA1+
C1-39	J2-20	XDTR1-
C1-40	J2-15	RTCLOCK1+
C1-41	J2-21	RTCLOCK1-
C1-42	J2-17	RRCLOCK1+
C1-43	J2-11	TCLKIN1+
C1-44		CABLE00N
C1-45	J1-23	XSPS0-
C1-46	J2-18	TTCLOCK1-
C1-47	J2-22	XRI1
C1-48	J2-23	XSPS1-
C1-49		UNUSED
C1-50		RCLKIN1+

Note

Pins C1-01, C1-19, and C1-44 are tied together. This ties CABLE1ON- (C1-19) and CABLE00N- (C1-44) to ground (C1-01).

Cable CBL8780-XXX: Direct connection of synchronous modems (V.35) to the peripheral bulkhead is supported using CBL8780-XXX, where XXX can be one of three possible dash numbers specifying cable length. The available cable lengths are:

Cable Number	Length (meters)
CBL8780-001	5.00
CBL8780-002	15.00
CBL8780-003	35.00

Cable CBL8780-XXX has two connectors, P1 and P2. The P1 connector is a 25-pin D-type male connector that connects to the peripheral bulkhead. The P2 connector is a 34-pin male connector that connects to the V.35 sync modem. Table A-9 lists the pin connections.

Table A-9
Connection Data for Cable CBL8780-XXX

From	To	Signal Name
P1-7	P2-B	GND
P1-4	P2-C	RTS
P1-2	P2-P	TDATA-
P1-14	P2-S	TDATA+
P1-7	P2-B	GND
P1-5	P2-D	CTS
P1-7	P2-B	GND
P1-22	P2-J	RI
P1-3	P2-R	RDATA-
P1-16	P2-T	RDATA+
P1-7	P2-B	GND
P1-6	P2-E	DSR
P1-15	P2-Y	TCLOCK+
P1-21	P2-AA	TCLOCK-
P1-7	P2-B	GND
P1-8	P2-F	DCD
P1-17	P2-V	RCLOCK+
P1-25	P2-X	RCLOCK-
P1-7	P2-B	GND
P1-20	P2-H	DTR

Notes

The ground wires common to P1-7 and P2-B must be pigtailed, soldered, and covered with heat-shrink sleeving.

Cable CBL8780-XXX is common to both two-line synchronous LACs and the MDLC.

B

Error and Status Messages

INTRODUCTION

This chapter details the errors/warnings that may occur during the normal operation of the ICS2/3, from initial startup to shutdown. They have been presented in such a way as to show the fault, its cause, and the action that has to be taken to remedy the fault.

Many messages use ICS2 or ICS3 in their text. In this book, where messages can be either ICS2 or ICS3, the explanation states ICS2/3. If the real message contains only ICS2, or only ICS3, then the explanation will show which one.

The chapter has been divided into the following topics:

- Downline load error messages
- PRIMENET/X.25 error messages
- RJE error messages
- SDLC error messages
- Asynchronous Error Messages

The summary tables, at the end of the chapter, provide a quick guide to error messages, with corresponding error reference numbers. These reference numbers are only for use in this chapter; they do not appear with the messages.

DOWNLINE LOAD SEQUENCE

The error messages associated with the downline load for all protocols are portrayed, in Table B-1, against a normal downline load sequence. The error reference numbers on the right-hand side of the sequence table correspond to possible downline load errors and where they may occur in the downline load sequence.

Table B-1
Normal Downline Load Sequence Status Messages

Status Message	Error Ref. Number
ICSn Device Address dd: Seg0_QCBs = xx ... Logical Connections = xx	1
ICSn Device Address dd: Dl_File = ICSn_xx.DL Type = File Header Details Date = Date File Created Time = Time File Created Program_size = dd bytes	2
ICSn Device Address dd: Status requested by host	3-7
...Good status returned to host	8-10
...Memory verified, size = yyyk bytes ...Down line load started ***	11
...Down line load finished *** ...Performing run time self-verify	12
...Run time self-verify successful	13
ICSn Device Address dd: Controller successfully booted ***	14-17

Notes

- n is 2 for the ICS2, or 3 for the ICS3.
- yyy is the memory size.
- The message "...Memory verified, size = yyyk bytes" is not issued for a warm start because no memory verification is performed.
- Any error occurring after Error Message No. 2 results in the display of the appropriate error/warning message, after which the processing skips to the end of the downline load sequence.
- dd denotes the Prime backplane address in octal (for example '10, '11, or some other octal number).
- xx denotes a varying integer.
- During the downline load sequence, error messages can be displayed in various combinations.
- Additional information on the run-time self-verify sequence can be found in Chapter 7 (DIAGNOSTICS).

Controller Status

The status of all controllers in the network can be displayed on any terminal, using the STATUS COMM command. A typical example follows:

Controller	Type	Device Address	Total-Lines Async	Total-Lines Sync	Bad-Lines Async	Bad-Lines Sync
ICS2	F-XX	10	56	2	0	0
MDLC	5646	50	0	4	No information	
AMLC	DMQ	53	16	0	No information	
AMLC	DMQ	54	16	0	No information	

Note

For the ICS2 controller above: F means Downline-Load (DL) File, the hyphen symbol (-) is the delimiter, and the XX is the DL file number. For example, F-05 means downline load file ICS2_05.DL is running in the ICS2.

DOWNLINE LOAD ERROR MESSAGES

The following error messages are associated with the downline load sequence. The error message is displayed on the system console.

ERROR No. 1

- Error: protocol combination not supported on ICS2 device address dd (BTPCC).

Cause

Wrong protocol tokens specified in the SYNC CNTRLR configuration directive.

Remedial Action

Specify the correct protocol token combinaton for the downline load file supported on your system. Restart the downline load sequence.

ERROR No. 2

- Error: whilst opening/rewinding dl file for ICS2/3 on device address dd (BTPCC).

Cause

The disk copy of the downline load file under the UFD DOWN_LINE_LOAD* has been corrupted or damaged.

Remedial Action

Replace the disk copy with a correct, up-to-date version of the file. Restart the downline load sequence by reinitiating the cold start.

ERROR No. 3

- Error: whilst loading device dd (PCCBS).

Cause

Unable to read the "header record" in the disk copy of the downline load file; that is, the file has been opened but cannot be read.

Note that this message may appear with other error messages, in which case this message is secondary to the other messages and indicates only that the error occurred during the loading of the specified device.

Remedial Action

Replace the disk copy with a correct, up-to-date version of the file. Restart the downline load sequence by reinitiating the cold start.

ERROR No. 4

- Error: dl file not formatted for DMT (PCCBS).

Cause

The downline load file is not formatted correctly for DMT type data transfers. Note that Error No. 3 also appears.

Remedial Action

Replace the disk copy with a correct, up-to-date version of the file. Restart the downline load sequence by reinitiating the cold start.

If the error condition still exists after taking the above action, call the Customer Field Service engineer.

ERROR No. 5

- Error: dl file packets are too large (PCCBS).

Cause

The downline load file must have been created with an incorrect packet size; that is, the packets are too large for the seg-0 buffers.

Remedial Action

Replace the disk copy with a correct up-to-date version of the file. Restart the downline load sequence by reinitiating the cold start.

If the error condition still exists after you take the above action, call the Customer Field Service engineer.

ERROR No. 6

- Error: no data packets in dl file (PCCBS).

Cause

The downline load file does not contain any data packets.

Remedial Action

Replace the disk copy with a correct, up-to-date version of the file. Restart the downline load sequence by reinitiating the cold start.

If the error condition still exists after taking the above action, call the Customer Field Service engineer.

ERROR No. 7

- Warning: timed notify not available (PCCBS).
  Warning: system will hang if controller fails to respond (PCCBS).

Cause

These warning messages indicate that it was not possible to force a timed notify on the process semaphore. Downline loading of the controller continues, but the system may hang (lockup) if the controller does not respond correctly. Error No. 3 also appears with these messages, together with an error code associated with the fault condition.

Remedial Action

There is no remedial action required unless the system hangs, in which case check error code in system manuals and ascertain what caused the fault condition; if possible, rectify problem.

If the error condition still exists after you take the above action, call the Customer Field Service engineer.

ERROR No. 8

- IPQNM error codes xx xx (PCCBS).

Cause

A call to an IPQNM routine caused an error condition to be raised. The octal codes in the message denote the cause.

Remedial Action

Write down the message and error codes and check for any logical inconsistencies in your configuration file (CONFIG); if possible rectify problem.

If the error condition still exists after taking the above action, call the Customer Field Service engineer, giving details of the message/error.

ERROR No. 9

- Error: controller has not responded to cold/warm start OCP (PCCBS).

Cause

A timeout occurred while waiting for a response from a controller. Error No. 3 denotes the address of the controller at fault.

Remedial Action

Check that the ICS2/3 card cage assembly power is ON. If not, turn it on and reboot.

If error condition still exists after checking the above, call the Customer Field Service engineer.

ERROR No. 10

- Error: controller codes= xxxx, xxxx, xxxx, xxxx (PCCBS).

Cause

An error status was received from the controller being loaded; the octal values contain the returned error code.

Remedial Action

A fault in the controller. Write down the error message and codes, and call the Customer Field Service engineer, giving details of the error message and error codes.

ERROR No. 11

- Error: program too large for controller (PCCBS).

Cause

The controller's memory is insufficient to contain the specified downline load file.

Remedial Action

Verify that the protocol token combination can be supported on this controller and that they are specified correctly.

If the error condition still exists after taking the above action, call the Customer Field Service engineer.

ERROR No. 12

- Error: controller has not responded to a down line load packet (PCCBS).

Cause

A timeout occurred while waiting for a response from a controller. Error No. 3 denotes the address of the controller at fault.

Remedial Action

Check that the ICS2/3 card cage assembly power is ON. If it is not, turn it on and reboot.

If the error condition still exists after you have checked the above action, call the Customer Field Service engineer.

ERROR No. 13

- Error: ICS2/3 has not responded to run time self-verify (PCCBS).

Cause

A timeout occurred while waiting for a response from a controller. Error No. 3 denotes the address of the controller at fault.

Remedial Action

Check that the ICS2/3 card cage assembly power is ON. If it is not, turn it on and reboot.

If the error condition still exists after you have checked the above action, call the Customer Field Service engineer.

ERROR No. 14

- Error: controller has returned xx words of (hex) status: xxxx xxxx xxxx xxxx

Cause

Failed to boot the programmable controller; the returned codes represent the last data returned from the controller.

Remedial Action

There is a probable fault in the controller. Write down the error message and codes, and pass this on to the Customer Field Service engineer.

ERROR No. 15

- Error: async line dd (Jxx) on line card in slot xx is inoperable (BTPCC).
- Error: rs232 sync line dd (Jxx) on line card in slot xx is inoperable (BTPCC).
- Error: v35 sync line dd (Jxx) on line card in slot xx is inoperable (BTPCC).

Cause

The specified line is faulty/inoperable.

Remedial Action

No action if line is not required for operational use; however, Customer Service should be called to rectify or replace the faulty LAC.

ERROR No. 16

- Error: line card in slot xx is inoperable (BTPCC).

Cause

The specified LAC is faulty and none of the lines on the card are usable.

Remedial Action

No action if line is not required for operational use; however, Customer Service should be called to rectify/replace the faulty Line Adaptor Card (LAC).

ERROR No. 17

- Error: line card in slot xx is unrecognizable (BTPCC).

Cause

The returned id from the LAC is unknown or unrecognizable.

Remedial Action

Check that the correct LAC type (ASYNC, RS232, V35) is installed at the specified slot in the LAC card cage assembly.

If it is the correct LAC type, then the LAC is probably faulty. Call the Customer Field Service engineer.

Notes

Errors 15, 16, and 17 do not cause the boot to fail. They are, however, indications of faulty equipment.

If the boot of a programmable controller fails for any reason, it is likely the following messages will also appear.

- Error: Boot failed on ICS device address xx (BTPCC).
- Error: ICS cold start configuration failure (COMINI).

There may also be messages generated by the PRIMOS operating system ASYNC code in the event of errors associated with ASYNC processing when requested. These messages are not listed in this guide.

ERROR No. 18

- Error: Not permissible to have ICS3 LACs on ICS2.

Cause

At start up time, an ICS3 LAC was detected in an ICS2 LAC card cage.

Remedial Action

Call your Customer Field Service engineer, who will arrange removal of the offending LAC. Do not remove the LAC yourself.

ERRORS No. 19 through 29 are reserved for general messages.

PRIMENET/X.25 ERROR MESSAGES

The error messages associated with PRIMENET/X.25 have been divided into two areas:

- During initial startup
- During normal operation

This breakdown should provide a more comprehensive interpretation of a PRIMENET/X.25 error message.

During Initial Startup

During the initial startup of PRIMENET/X.25 (after a START_NET command is issued), the following error messages may occur. These error messages are displayed only on the system console and are not logged into the Network Event Logging File (NETREC).

ERROR No. 30

- Error: ICS2/3 received bad config, already configured.
  Line xx (PRIMENET).

Cause

1. Data may have been corrupted during transmission across the backplane to the ICS2/3.
2. Software may be corrupted.
3. Possible ICS2/3 hardware malfunction.

Remedial Action

Call the Customer Field Service engineer. However, as an interim measure, try to warm start or to cold start the complete machine.

ERROR No. 31

- Error: ICS2/3 received bad config, bad revision #.
  Line xx (PRIMENET).

Cause

The downline load file has been incorrectly generated.

Remedial Action

Reinstall correct, compatible software.

ERROR No. 32

- Error: ICS2/3 received bad config, bad framing type.
  Line xx (PRIMENET).

Cause

1. The downline load file does not support the requested protocol.
2. Data may have been corrupted during transmission across the backplane to the ICS2/3.
3. Software may be corrupted.
4. Possible ICS2/3 hardware malfunction.

Remedial Action

Check that requested protocols are valid. If the requested protocols are valid, call the Customer Field Service engineer.

ERROR No. 33

- Error: ICS2/3 received bad config, not full duplex.
  Line xx (PRIMENET).

Cause

Half-duplex line was specified in CONFIG_NET and the downline load file in use does not support half-duplex (HDX) operation.

Remedial Action

Use a downline load file that does support half-duplex (if available); otherwise do not attempt to use half duplex PRIMENET lines.

Note

HDX PRIMENET is not supported at Rev. 20 of PRIMOS.

ERROR No. 34

- Error: ICS2/3 received bad config, bad diagnostic value.
  Line xx (PRIMENET).

Cause

1. Data may have been corrupted during transmission across the backplane to the ICS2/3.
2. Software may be corrupted.
3. Possible ICS2/3 hardware malfunction.

Remedial Action

Reinstall correct, compatible software; if the fault persists, call a Customer Field Service engineer.

ERROR No. 35

- Hardware Error: ICS2/3 has faulty IBC. Line xx (PRIMENET).

Cause

Hardware malfunction. The IBC has an unrecoverable error condition.

Remedial Action

Call the Customer Field Service engineer. However, as an interim measure, try to warm start or cold start the complete machine.

ERROR No. 36

- Error: ICS2/3 has insufficient buffers available.
  Line xx (PRIMENET).

Cause

Unable to allocate sufficient internal buffers for the requested protocol combination. The line number can be used against the configuration directives to determine the controller address, and thereby determine which controller is at fault.

Remedial Action

Call the Customer Field Service engineer to upgrade the ICS2/3 controller board from 250K bytes to 500K bytes of internal memory.

ERROR No. 37

- Error: ICS2/3 has no lac or lac not responding.
  Line xx (PRIMENET).

Cause

The LAC associated with the specified line is missing or not responding to commands from the controller.

Remedial Action

Ensure that the LAC is in the correct slot and correctly inserted. If the fault persists, call Customer Field Service engineer.

ERROR No. 38

- Error: ICS2/3 has insufficient memory for requested protocol.
  Line xx (PRIMENET).

Cause

The protocols requested in the SYNC/SMLC CNTRLR configuration directives require more memory than is currently available in the controller.

Remedial Action

Check that protocol tokens are specified correctly in the configuration directives. Check that the ICS2/3 has the correct memory size. (See downline load messages on the system console.) If the fault persists, call the Customer Field Service engineer.

Note

From Rev. 20.0, the ICS2 controller requires 256K bytes of RAM for any of the HDLC, RJE, or SDLC protocols. ASYNC only will still run in a 128K byte ICS2.

ERROR No. 39

- Error: ICS2/3 has wrong IBC micro-code for requested protocol. Line xx (PRIMENET).

Cause

Either the incorrect IBC microcode has been loaded or the downline load file has been generated with the wrong IBC microcode for the protocol that needs to be supported.

Remedial Action

Ensure that the downline load file supports the requested protocol, that is, if BSC framing has been requested, then ensure that the downline load file supports BSCX25. If the fault persists, call the Customer Field Service engineer who should supply the correct downline load file for the supported PRIMOS revision.

ERROR No. 40

- Hardware Error: ICS2/3 has incorrect lac type. Line xx (PRIMENET).

Cause

The LAC does not support the requested protocol for the line specified.

Remedial Action

Check that the configuration file (CONFIG) is consistent with the SYNC/SMLC configuration directives and the LAC type in use for this line. If the fault persists, call the Customer Field Service engineer.

ERROR No. 41

- Hardware Error: ICS2/3 has incorrect cable type. Line xx (PRIMENET).

Cause

Wrong cable in use on the controller. The line number can be matched against the configuration directives to determine the controller address and thereby, the controller that is at fault.

Remedial Action

Call the Customer Field Service engineer with details of the fault.

ERROR No. 42

- Hardware Error: ICS2/3 has unattached cable. Line xx (PRIMENET).

Cause

The cable on the controller is missing or disconnected or faulty.

Remedial Action

Check all cabling and if the fault persists, call the Customer Field Service engineer.

Error Nos. 43-49

- Error: failed to create ICS PRIMENET event queue. Line xx (PRIMENET).
- Error: failed to initialize lc, rejected by IPQNM. Line xx (PRIMENET).
- Error: failed to initialize lc, rejected by ICS. Line xx (PRIMENET).
- Error: failed to initialize lc, timeout. Line xx (PRIMENET).
- Error: ICS received incomplete config block. Line xx (PRIMENET).
- Error: ICS received invalid config block. Line xx (PRIMENET).
- Error: ICS device is inoperable. Line xx (PRIMENET).

Cause

Internal error, possibly software related, on the controller. The line number can be matched against the configuration directives to determine the controller address, and thereby determine which controller is at fault.

Remedial Action

Call the Customer Field Service engineer. However, as an interim measure, try to warm start or to cold start the complete machine.

ERROR No. 50

- Error: ICS line not started, too many sync lines specified.
  Line xx (PRIMENET).

Cause

Self explanatory.

Remedial Action

Check configuration commands.

ERROR No. 51

- Error: failed to receive config cmd response.
  Line xx (PRIMENET).

Cause

ICS2/3 software error.

Remedial Action

Try a warm start; if that does not clear the problem, try a cold start. If the problem persists, call the Customer Field Service engineer.

PRIMENET/X.25 Normal Operation

During normal operation of PRIMENET/X.25, any errors associated with either an ICS2/3 or an ICS1 controller are recorded in the Network Event Logging File (LOGREC).

This file can be closed at any time by the System Operator and the PRINT_NETLOG or LOGPRT -NET command used to translate the binary log file to readable ASCII text format (file NETLST).

For more information about LOGPRT and/or PRINT_NETLOG, see the System Operator's Guide or the PRIMENET Guide.

The following error messages are divided into two groups, 1 and 2. Group 1 error messages appear in the network event-logging file for ICS1, ICS2, and ICS3 controllers. (See Table B-2.) Group 2 error messages appear only for ICS2 or ICS3 controllers. (See Table B-3.)

Table B-2
Group 1 Messages Appearing in NETLST File

Group 1 Number		Error Ref. Number
ICS.00	(X.25) DECONFIGURE CODE WORD NOT QUEUED FOR LOGICAL LINE XX	60
ICS.01	(X.25) LOGICAL CONNECTION DELETED FOR LOGICAL LINE XX	61
ICS.02	(X.25) LOGICAL CONNECTION NOT BROKEN FOR LOGICAL LINE XX	62
ICS.03	(X.25) LCAD_ NOT FOUND IN LCB FOR LOGICAL LINE XX	63
ICS.04	(X.25) LOGICAL CONNECTION LOST FOR LOGICAL LINE XX	64
ICS.05	(X.25) FLUSH TIMEOUT FOR LOGICAL LINE XX	65
ICS.06	(X.25) ILLEGAL FLUSH COMPLETE FOR LOGICAL LINE XX	66
ICS.07	(X.25) SYNCHRONOUS LINE NOT ASSIGNED FOR LOGICAL LINE XX	67
ICS.08	(X.25) UNIDENTIFIABLE ERROR FOR LOGICAL LINE XX	68
ICS.09	(X.25) LINE NOT DEFINED: XX	69
ICS.10 to ICS.19	reserved for general ICS1/2/3 messages	70-79

Cause

The above error messages are usually generated as a result of internal hardware/software error conditions.

Remedial Action

Call the Customer Field Service engineer if the error condition interferes with normal network operations.

Table B-3
Group 2 Messages Appearing in the NETLST File

Group 2 Number	Error Ref. Number
ICS.20 (X.25) INVALID COMMAND TO IBC FOR LOGICAL LINE XX	80
ICS.21 (X.25) INVALID PROTOCOL ID FOR LOGICAL LINE XX	81
ICS.22 (X.25) LAC BUS: UNMAPPED LINE INTERRUPT FOR LOGICAL LINE XX	82
ICS.23 (X.25) LAC BUS: ADDRESS PARITY ERROR FOR LOGICAL LINE XX	83
ICS.24 (X.25) LAC BUS: DATA PARITY ERROR FOR LOGICAL LINE XX	84
ICS.25 (X.25) LAC BUS: PARITY ERROR ON IA CYCLE FOR LOGICAL LINE XX	85
ICS.26 (X.25) UNMAPPED LINE ON DMX SCAN LIST FOR LOGICAL LINE XX	86
ICS.27 to ICS.35 reserved for ICS2/3 IBC errors	87-95
ICS.36 (X.25) BISYNC FRAMING ERROR FOR LOGICAL LINE XX	96
ICS.37 (X.25) STATUS BUFFER OVERFLOW FOR LOGICAL LINE XX	97
ICS.38 to ICS.51 reserved for ICS2/3 (Z8001) errors	98-111
Not allocated	112-119

Cause

The above error messages are usually generated as a result of internal hardware and/or software error conditions. The ICS.37 message may result from an unusually high number of interrupts from a LAC (such as when RI is tied to the TX clock).

Remedial Action

Check the LACs, the LAC cable(s), and the data set(s). Call the Customer Field Service engineer if the error condition interferes with normal network operations.

RJE ERROR MESSAGES

RJE error messages are described below in two categories: errors during RJE line enabling, and errors during normal RJE running. The full format of the messages is as follows:

```
HH:MM:SS : Line xx: ERR_-8 "Message" Arg = xxxx
```

All error messages are time-tagged. If PRIMOS passes an error code to the ICS2/3, this code appears at the end of the message in the form: Arg = xxxx (as shown above), where xxxx is the PRIMOS error code. The following descriptions show an abbreviation of the complete message. Both categories of message appear on the operator console/como file (RJPROC.COMO) together with a "line down" message in the form:

```
HH:MM:SS : Line xx: DIM_-8 "ICS2/3 Line Down" Arg = xx
```

Errors In RJE Line Enabling

ERROR No. 120

- Error: ICS2/3 Line in use

Cause

This error is generated by the ICS2/3 when PRIMOS sends a configuration block for a line already in use. This is possible only if a line disable command was not correctly executed in the ICS2/3. The line could be in use by X.25 or SDLC.

Remedial Action

Call the Customer Field Service engineer. However, as an interim measure, try to warm start or cold start the machine.

ERROR No. 121

- Error: ICS2/3 received bad Config block

Cause

This error is generated by the ICS2/3 when the configuration block sent by PRIMOS does not have valid parameter values. Code xxxx indicates the wrong parameter value.

Remedial Action

Carefully check the Site Definition Files (SDFs). If no apparent problem is found, call the Customer Field Service engineer. However, as an interim measure, try to warm start or cold start the machine.

ERROR No. 122

- Error: ICS2/3 has faulty IBC

Cause

This error is generated by the ICS2/3 when the IBC malfunctions.

Remedial Action

Call the Customer Field Service engineer; the ICS2/3 controller card may have to be replaced. However, as an interim measure, try to warm start or cold start the machine.

ERROR No. 123

- Error: ICS2/3 has incorrect LAC type

Cause

This error is generated by the ICS2/3 when the LAC associated with the specified line (in the configuration block sent from PRIMOS) is not the correct type for the allocated slot.

Remedial Action

Check that the SYNC line mapping commands in the CONFIG file are correct. Also, check that the SDF specifies the correct line, and that the correct LAC is in the correct ICS2/3 LAC Card Cage slot. If the fault still exists, call the Customer Field Service engineer.

ERROR No. 124

- Error: ICS2/3 has incorrect cable type

Cause

This error is generated by the ICS2/3 when the cable connected to the specified LAC (for the physical line number specified in the configuration block sent by PRIMOS) is incorrect or not responding.

Remedial Action

Call the Customer Field Service engineer.

ERROR No. 125

- Error: ICS2/3 has no cable attached

Cause

This error is generated by the ICS2/3 when the cable connected to the specified LAC (for the physical line number specified in the configuration block sent by PRIMOS) is missing or not connected.

Remedial Action

Call the Customer Field Service engineer to replace/connect/check the cable.

ERROR No. 126

- Error: ICS2/3 has no LAC

Cause

This error is generated by the ICS2/3 when the LAC associated with the specified line (in the configuration block sent by PRIMOS) is missing or not responding.

Remedial Action

Check all SYNC directives in the configuration file (CONFIG). Also, check the SDF. Call the Customer Field Service engineer, and as an interim measure, try a new software configuration if a spare LAC is available.

ERROR No. 127

- Error: ICS2/3 has incorrect IBC microcode

Cause

This error is generated by the ICS2/3 when the IBC microcode is not relevant to the RJE session.

Remedial Action

Call the Customer Field Service engineer.

RJE Normal Running

The following errors may occur during the running of a normal RJE session. The error messages appear on the operator terminal/como file (RJPROC) together with the standard "line down" message described earlier. All error messages are time-tagged. If PRIMOS passes an error code to the ICS2/3 it is attached to the end of the error message in the form: Arg = xxxx, where xxxx is the PRIMOS error code.

ERROR No. 128

- Error: ICS2/3 has not enough memory

Cause

This error is generated by the ICS2/3 during an RJE session when the RJE process cannot obtain buffers to receive data over the line. This error could possibly occur when some other protocol is running along with the RJE.

Remedial Action

Check that the protocol tokens are specified correctly in the configuration directives. If the fault persists, call the Customer Field Service engineer, giving as much information as possible.

ERROR No. 129

- Error: ICS2/3 detected internal protocol error Arg = xxxx

Cause

This error is generated by the ICS2/3 during an RJE session whenever a software inconsistency is encountered.

Remedial Action

Take note of the Arg/code, and call the Customer Field Service engineer.

Table B-4 shows the code, the software state in which the error occurred (see Chapter 2), the error number within that state, and a brief explanation of the error.

Table B-4
Error No. 129 — Description of Codes

Error Code	RJE State	Error Number	Description
2	0	2	Could not set up a timer
3	0	3	Internal table overflow
6	0	6	Out of buffers
7	0	7	DMX failure
8	0	8	Illegal buffer type from PRIMOS
9	0	9	Could not release Z8001 buffer
10	0	10	IPQNM failure
11	0	11	Queue creation failure
12	0	12	Illegal call to Buffer Server Process
13	0	13	IPQNM failure
14	0	14	IBC failure
34	1	2	IBC failure
65	2	1	Invalid command from PRIMOS
66	2	2	Out of buffers
67	2	3	Illegal call to Buffer Server Process
68	2	4	PRIMOS pointer buffer invalid
69	2	5	Illegal buffer type from PRIMOS
70	2	6	Out of buffers
71	2	7	Could not DMX from PRIMOS
72	2	8	Invalid buffer format
73	2	9	Invalid transparent record format

Table B-4 (continued)
Error No. 129 — Description of Codes

Error Code	RJE State	Error Number	Description
74	2	10	Could not release Z8001 buffer
75	2	11	Receive complete from IBC
79	2	15	IBC failure
80		16	
81		17	
82	2	18	Transmit underrun from IBC
95	2	31	Out of buffers
99	3	3	Null frame received
100	3	4	Invalid frame received
101	3	5	Could not release Z8001 buffer
102	3	6	IPQNM failure
103		7	
104	3	8	IBC failure
127	3	31	Out of buffers
130	4	2	Receive complete from IBC
132	4	4	Transmit complete from IBC
133	4	5	Could not release Z8001 buffer
135	4	7	Transmit underrun from IBC
143	4	15	IBC failure
144		16	
145		17	
159	4	31	Out of buffers
195	6	3	Null frame received
196	6	4	Invalid frame received
197	6	5	Could not release Z8001 buffer

Table B-4 (continued)
Error No. 129 -- Description of Codes

Error Code	RJE State	Error Number	Description
198	6	6	IPQNM failure
199		7	
200	6	8	IBC failure
204	6	12	Transmit complete from IBC
205	6	13	Transmit underrun from IBC
207	6	15	IBC failure
208		16	
209		17	
223	6	31	Out of buffers
225	7	1	Receive complete from IBC
226	7	2	Receive started from IBC
227	7	3	Transmit complete from IBC
228	7	4	Received a control character from IBC
230	7	6	Transmit underrun from IBC
231	7	7	Receive parity error from IBC
232	7	8	Receive overrun from IBC
233	7	9	No receive buffer available from IBC
234	7	10	CRC error from IBC
235	7	11	Receive buffer overflow from IBC

ERROR No. 130

- Error: Line xx: Error in Sending Config Block. ICS2EVNT.

Cause

This error is generated by the RJE Protocol Handler (in PRIMOS) when it cannot successfully send the configuration block to the ICS2 or ICS3, via the logical connection. This error is detected in the ICS2EVNT module.

Remedial Action

Disable the line and try to start the RJE session by re-enabling it. If the fault persists, call the Customer Field Service engineer.

ERROR No. 131

- Error: Line xx: Error in Dequeuing from IPQNM. Code = 58

Cause

This error is generated by the RJE Protocol Handler (in PRIMOS) when it detects an error while dequeuing an item from the data queue of line xx.

Remedial Action

Disable the line, and try to start the RJE session by re-enabling it. If the fault persists, call the Customer Field Service engineer.

ERROR No. 132

- Error: Line xx: Error in ICS2/3 Line Event. Code = nn

Cause

This error is generated by the RJE Protocol Handler (in PRIMOS) when it detects the following error conditions during the RJE session on the line xx. The code number nn (between 42 to 59) indicates the error.

Table B-5 shows the codes and their descriptions.

Table B-5
Error No. 132 — Description of Codes

Code	Description
42	No logical connection with ICS2/3 for given line.
43	Line does not belong to ICS2/3.
44	ENQ$A failure (sending message) with ICS2/3.
45	Invalid ICS2/3 controller address.
46	Event queue creation failed (ICS2/3).
47	LCINT call failed - (with ICS2/3).
48	Logical connection rejected by ICS2/3.
49	ENQ$A failure (sending CFG block).
50	ENQ$A failure (sending CMD word).
51	No free blocks in free pool.
52	Bad event from ICS2/3.
53	Wrong encoded response from ICS2/3.
54	Block unexpected from ICS2/3.
55	More entries in pointer buffer from ICS2/3.
56	Unexpected data block from ICS2/3.
57	Logical connection deletion failed (ICS2/3).
58	No entries in the ICS2/3 event queue.
59	Dequeuing failure from the ICS2/3 queue.

Remedial Action

The line is disabled; try to start the RJE session by re-enabling it. If the fault persists, call the Customer Field Service engineer.

ERROR Nos. 133 to 149 are reserved for additional RJE error messages.

SDLC ERROR MESSAGES

SDLC error messages are fully described in Appendix A of the PRIME/SNA Operator's Guide. Only ICS2/3 or LAC messages are mentioned here.

Notes

When the message displays ICSx, the description applies to the controller using SDLC protocol. The controller type number is not shown.

Error message numbers apply to this appendix only; they are not connected with any other message numbering scheme.

Errors 150 to 155 occur during attempts to start the SERVER.

Errors 156 to 170 occur during attempts to start the remote system. The error message descriptions show only the reasons; however, the complete message is formatted as follows:

Remote system cannot be started : <remote-system-name>
— <reason>

ERROR No. 150

- Request failed because there is no controller with address <address>.

Cause

When trying to start the SERVER, the controller number specified in the SNA configuration file did not match any of the installed ICS2/3's device addresses.

Remedial Action

Check SNA configuration file against installed controller device addresses.

ERROR No. 151

- Request failed because there is insufficient ICSx memory to support SDLC.

Cause

When trying to start the SERVER, the ICS2/3 was found to have less than 256K bytes of memory to support SDLC.

Remedial Action

If memory is believed to be 256K bytes or larger, consult your Customer Field Service engineer.

ERROR No. 152

- Request failed because there are insufficient ICSx buffers to support SDLC.

Cause

Although the ICS2/3 may have 256K bytes of memory, there are not enough ICS2/3 buffers to support the remote system connection.

Remedial Action

Consult your Customer Field Service engineer.

ERROR No. 153

- Request failed because of an incompatibility between the SERVER and SDLC.

Cause

Incompatible revisions of the SERVER and SDLC (downline load file).

Remedial Action

Message PRIME/SNA0998 in the SERVER log file contains the required and installed revision numbers for SDLC support.

ERROR No. 154

- Request failed because the SDLC support is currently stopping.

Cause

This can indicate either a problem with the SDLC support stopping while the SERVER is attempting to start a connection (timing window problem) or a problem with support in the ICS2/3.

Remedial Action

Reissue the SNA_SERVER -START command, or, if the ICS2/3 is suspected, warm start to reset the ICS2/3. If the problem persists, contact your Customer Field Service engineer.

ERROR No. 155

- Request failed because the SERVER and SDLC are not in synch.

Cause

The SERVER is attempting to establish its first connection with the ICS2/3 controller, but SDLC already has one or more connections with the SERVER for the controller. There is an error in either the SERVER or the ICS2/3 support.

Remedial Action

Try a warm start to reset the ICS2/3. If the problem persists contact your Customer Field Service engineer.

ERROR No. 156

- cannot establish line signals

Cause

The ICS2/3 could perform the necessary handshake with the local modem, but the local modem could not perform the necessary handshake with the remote modem.

Remedial Action

Typically, this problem occurs when the line between the modems is faulty, the remote modem is not switched on, or the remote modem is faulty. Check all of the above. If the problem persists, call the Customer Field Service engineer.

ERROR No. 157

- cannot raise line signals

Cause

The ICS2/3 could not perform the necessary handshake with the local modem.

Remedial Action

Typically, this occurs when the cable between the bulkhead and the modem is not properly connected, the modem is not powered on, or the ICS2/3 or modem is faulty. Check all of the above. If the problem persists, call the Customer Field Service engineer.

ERROR No. 158

- fatal error occurred — code <nnnn>

Cause

The SERVER does not have a specific mapping for the type of failure that occurred. As a result, the code associated with the failure is displayed. In some cases, the SERVER adds the value 1000 or 2000 to the code to help identify the failure to Prime personnel.

Remedial Action

In most cases, contact your Customer Field Service engineer and provide details of the error code. Table B-6 describes codes 1150 through 1163 which relate to fatal ICS2/3 errors.

Table B-6
Error No. 158 — Description of Codes

Code	Description
1150	LAC bus address parity error
1151	LAC bus data parity error
1152	Invalid IBC PC error
1153	Invalid protocol ID
1154	Unmapped LAC bus interrupt
1155	Unknown IBC status function
1156	Unknown IBC error status
1157	Unknown IBC data channel status
1158	Invalid IBC command function
1159	Hardware initialization error
1160	System status lost
1161	Logical interrupts lost
1162	Protocol status on foreign line
1163	SDLC status lost

ERROR No. 159

- incompatible ICSx controller

Cause

The controller specified in the SNA configuration is incompatible with the controller being used for an already-started remote system.

Remedial Action

The SNA configuration file must have been edited since the other remote system was started.

ERROR No. 160

- incompatible LAC line number

Cause

The LAC number specified in the SNA configuration is incompatible with the LAC number being used for an already-started remote system.

Remedial Action

Consult your Customer Field Service engineer.

ERROR No. 161

- insufficient ICSx receive buffers

Cause

The ICS2/3 does not have sufficient resources to support the remote system connection.

Remedial Action

Consult your Customer Field Service engineer.

ERROR No. 162

- insufficient SDLC memory

Cause

The ICS2/3 does not have sufficient resources to support the remote system connection.

Remedial Action

Consult your Customer Field Service engineer.

ERROR No. 163

- insufficient SDLC table memory

Cause

The ICS2/3 does not have sufficient resources to support the remote system connection.

Remedial Action

Consult your Customer Field Service engineer.

ERROR No. 164

- invalid LAC type (RS-232 or V.35)

Cause

The LAC installed is the wrong type; that is, RS232 where V.35 should be fitted, or V.35 where RS232 should be fitted.

Remedial Action

Check the type fitted and consult your Customer Field Service engineer if it needs changing.

ERROR No. 165

- no buffer to perform DMQ

Cause

There is a shortage of wired buffers for transferring data (via the I/O channel) between the Prime CPU and the ICS2/3. Starting another connection calls for more buffers; therefore, the -START request is rejected.

Remedial Action

The shortage may be temporary; try starting again later.

ERROR No. 166

- no SDLC connections are available

Cause

No communications paths are available between the SERVER and the SDLC support in the ICS2/3.

Remedial Action

This problem should not occur; if it does, contact your Customer Field Service engineer.

ERROR No. 167

- SDLC didn't respond

Cause

This indicates an error in either the SERVER or ICS2/3 support.

Remedial Action

Try a warm start. If the problem persists, consult your Customer Field Service engineer.

ERROR No. 168

- the LAC cable is not connected

Cause

The LAC card to bulkhead cable is not properly connected.

Remedial Action

Check the cable.

ERROR No. 169

- the LAC line does not exist

Cause

No LAC is in the slot designated by the LAC number in the SNA configuration file.

Remedial Action

Check the LAC configuration.

ERROR No. 170

- the line is already being used

Cause

The line has been designated for use by another protocol.

Remedial Action

Check the LAC configuration.

ERROR No. 171

- unable to communicate with the ICSx

Cause

The SERVER was unable to communicate with the ICS2/3.

Remedial Action

Contact your Customer Field Service engineer.

ASYNCHRONOUS ERROR MESSAGES

The following asynchronous line error messages are described in the System Administrator's Guide. The messages appear on the supervisor terminal.

ERROR No. 180

- System console command only. (AMLC)

Cause

An attempt has been made to use the AMLC command from a terminal other than the supervisor terminal. No action is taken on the command.

Remedial Action

None necessary.

ERROR No. 181

- Line number is required. (AMLC)

Cause

A line number has not been specified in the command. No action is taken on the command.

Remedial Action

None necessary.

ERROR No. 182

- Line number ooo is out of range, highest line number is 3777. (AMLC)

Cause

The line number (ooo octal) provided is less than zero or higher than 377. No action is taken on the command.

Remedial Action

Correct the line number.

ERROR No. 183

- protocol is not a valid protocol name. (AMLC)

Cause

An unsupported protocol name (protocol) has been specified for the line. Valid protocol names are TTY, TRAN, TTYUPC, TTYNOP, TT8BIT, ASD, and obsolete protocol names TTYHS, TRANHS, and TTYHUP. No action is taken on the command.

Remedial Action

Check protocol parameter in AMLC command.

ERROR No. 184

- Buffer ooo is a remote login buffer (is between lll and hhh). (AMLC)

Cause

A buffer number of ooo has been specified that is in the range (from lll octal to hhh octal) reserved for users logging in from a remote system. No action is taken on the command.

Remedial Action

Check specified buffer number.

ERROR No. 185

- Buffer number ooo is out of range, highest buffer number is hhh. (AMLC)

Cause

The buffer number of ooo (octal) is not within the range of buffer numbers configured on the system. Valid buffer numbers range from two to the sum of the values specified for the NTUSR, NRUSR, and NAMLC configuration directives minus two. (This highest value is reported as hhh octal.) No action is taken on the command.

Remedial Action

Check the buffer number.

ERROR No. 186

- Line is already assigned to user ddd, must unassign first. (AMLC)

Cause

An attempt has been made to zero the buffer number of a line that is already assigned to user ddd (decimal). Any line may be unassigned from the supervisor terminal using the UNASSIGN AMLC command. No action is taken on the command.

Remedial Action

Check configuration.

ERROR No. 187

- Buffer number bbb is already in use by line ooo. (AMLC)

Cause

The user number bbb (octal) is currently associated with another line, ooo (octal). No more than one line can be associated with the same user number. No action is taken on the command.

Remedial Action

Check configuration.

ERROR No. 188

- Bad parameter. (ALCONF)

Cause

An attempt has been made to specify a line speed (baud rate) on an ICS async line that does not support that rate. This can occur if the speed set for the software programmable clock (set by the AMLCLK directive) is not supported by ICS hardware.

Remedial Action

Check line speeds, AMLCLCK directive, and hardware.

ERROR No. 189

- Operation unsuccessful. (ALCONF)

Cause

An attempt to change the configuration word for an ICS async line failed. This error message may indicate a temporary failure of PRIMOS to communicate with the ICS controller, or a problem with the ICS controller itself. No action is taken on the command attempt; however, if the command is reissued, it may succeed.

Remedial Action

Check configuration word and PRIMOS to ICS communications.

ERROR No. 190

- Device not available. (ALCONF)

Cause

An attempt to change the configuration word for an ICS async line failed. This error message indicates that PRIMOS is unable to communicate with the ICS controller, and has marked the controller as being unusable until the next warm start or cold start of PRIMOS. No action is taken on the command. Unlike other errors described here, an error condition is not raised in this case, because the system could still continue to operate by ignoring the unusable controller.

Remedial Action

Check the ICS controller.

ERROR No. 191

- Line ooo now supports user login, user number is ddd. (AMLC)

Cause

This message indicates that a user can now log in on the line number specified ooo (octal) as the indicated user number ddd (decimal).

Remedial Action

None required.

ERROR No. 192

- Line ooo is now an assignable line. (AMLC)

Cause

This message indicates that the line ooo (octal), was made assignable by specifying a buffer number of zero.

Remedial Action

None required.

ERROR Nos. 193-202

- Inconsistent ASYNC cold start configuration for ICS2/3 device dd: an async line card has been found where not expected in slot ss.
- Inconsistent ASYNC cold start configuration for ICS2/3 device dd: faulty line card in slot ss where an async line card was expected.
- Inconsistent ASYNC cold start configuration for ICS2/3 device dd: sync line card in slot ss where an async line card was expected.
- Inconsistent ASYNC cold start configuration for ICS2/3 device dd: slot ss is empty where an async line card was expected.
- ICS2/3 device dd has returned the wrong number (m) of status words.
- Inconsistent ASYNC warm start configuration for ICS2/3 device dd: an async line card has been inserted into slot ss.
- Inconsistent ASYNC warm start configuration for ICS2/3 device dd: the async line card in slot ss in now inoperable.
- Inconsistent ASYNC warm start configuration for ICS2/3 device dd: the async line card in slot ss has been removed or is now inoperable.
- Inconsistent ASYNC warm start configuration for ICS2/3 device dd: a sync line card has been inserted into slot ss.
- Bad key value (n) in call (ICSCFG).

Cause

The LAC configuration was checked, at cold or warm start, and the configuration found did not match the expected configuration. The messages are self explanatory.

Remedial Action

Check asynchronous LAC configuration using the information contained in the error message to identify the LAC slot which is at fault.

ERROR SUMMARY TABLES

The following Tables, B-7 through B-13, give the errors/warnings together with their relevant error numbers. Note some errors in RJE and X.25 have similar descriptions; so, please be sure you consult the correct section of error messages. In most cases, the error/warning message appears on one line; however, in these tables, some are shown on two or more lines. Error reference numbers are used in this chapter only, NOT in the messages.

Table B-7
Error/Warning Messages (General)

Error/Warning Message	Error Ref. Number
Error: protocol combination not supported on ICS2 device address dd (BTPCC).	1
Error: whilst opening/rewinding dl file for ICS2/3 on device address dd (BTPCC).	2
Error: whilst loading device dd (PCCBS).	3
Error: dl file not formatted for DMT (PCCBS).	4
Error: dl file packets are too large (PCCBS).	5
Error: no data packets in dl file (PCCBS).	6
Warning: timed notify not available (PCCBS). Warning: system will hang if controller fails to respond (PCCBS).	7
IPQNM error codes xx xx (PCCBS).	8
Error: controller has not responded to cold/warm start OCP (PCCBS).	9

Table B-7 (continued)
Error/Warning Messages (General)

Error/Warning Message	Error Ref. Number
Error: controller codes= xxxx, xxxx, xxxx, xxxx (PCCBS).	10
Error: program too large for controller (PCCBS).	11
Error: controller has not responded to a down line load packet (PCCBS).	12
Error: ICS2/3 has not responded to run time self-verify (PCCBS).	13
Error: controller has returned xx words of (hex) status: xxxx xxxx xxxx xxxx	14
Error: async line dd (Jxx) on line card in slot xx is inoperable (BTPCC).	
Error: rs232 sync line dd (Jxx) on line card in slot xx is inoperable (BTPCC).	
Error: v35 sync line dd (Jxx) on line card in slot xx is inoperable (BTPCC).	15
Error: line card in slot xx is inoperable (BTPCC).	16
Error: line card in slot xx is unrecognizable (BTPCC).	17
Error: Not permissible to have ICS3 LACs on ICS2	18
Reserved for general error messages.	19-29

Table B-8
Error/Warning Messages (PRIMENET/X.25)

Error/Warning Message	Error Ref. Number
Error: ICS received bad config, already configured. Line xx (PRIMENET).	30
Error: ICS received bad config, bad revision #. Line xx (PRIMENET).	31
Error: ICS received bad config, bad framing type. Line xx (PRIMENET).	32
Error: ICS received bad config, not full duplex. Line xx (PRIMENET).	33
Error: ICS received bad config, bad diagnostic value. Line xx (PRIMENET).	34
Hardware Error: ICS2/3 has faulty IBC. Line xx (PRIMENET).	35
Error: ICS2/3 has insufficient buffers available. Line xx (PRIMENET).	36
Error: ICS2/3 has no lac or lac not responding. Line xx (PRIMENET).	37
Error: ICS2/3 has insufficient memory for requested protocol. Line xx (PRIMENET).	38
Error: ICS2/3 has wrong IBC microcode for requested protocol. Line xx (PRIMENET).	39
Hardware Error: ICS2/3 has incorrect lac type. Line xx (PRIMENET).	40
Hardware Error: ICS2/3 has incorrect cable type. Line xx (PRIMENET).	41
Hardware Error: ICS2/3 has unattached cable. Line xx (PRIMENET).	42

Table B-8 (continued)
Error/Warning Messages (PRIMENET/X.25)

Error/Warning Message	Error Ref. Number
Error: failed to create ICS PRIMENET event queue. Line xx (PRIMENET).	43
Error: failed to initialize lc, rejected by IPQNM. Line xx (PRIMENET).	44
Error: failed to initialize lc, rejected by ICS. Line (PRIMENET).	45
Error: failed to initialize lc, timeout. Line (PRIMENET).	46
Error: ICS received incomplete config block. Line xx (PRIMENET).	47
Error: ICS received invalid config block. Line xx (PRIMENET).	48
Error: ICS device is inoperable. Line xx (PRIMENET).	49
Error: ICS line not started, too many sync lines specified. Line xx (PRIMENET).	50
Error: failed to receive config cmd response. Line xx (PRIMENET).	51
Reserved for PRIMENET Error/Warning Messages.	52-59

Table B-9
Error/Warning Messages (LOGPRT/PRINT_NETLOG)

Error/Warning Message	Error Ref. Number
ICS.00 (X.25) DECONFIGURE CODE WORD NOT QUEUED FOR LOGICAL LINE XX	60
ICS.01 (X.25) LOGICAL CONNECTION DELETED FOR LOGICAL LINE XX	61
ICS.02 (X.25) LOGICAL CONNECTION NOT BROKEN FOR LOGICAL LINE XX	62
ICS.03 (X.25) LCADl_ NOT FOUND IN LCB FOR LOGICAL LINE XX	63
ICS.04 (X.25) LOGICAL CONNECTION LOST FOR LOGICAL LINE XX	64
ICS.05 (X.25) FLUSH TIMEOUT FOR LOGICAL LINE XX	65
ICS.06 (X.25) ILLEGAL FLUSH COMPLETE FOR LOGICAL LINE XX	66
ICS.07 (X.25) SYNCHRONOUS LINE NOT ASSIGNED FOR LOGICAL LINE XX	67
ICS.08 (X.25) UNIDENTIFIABLE ERROR FOR LOGICAL LINE XX	68
ICS.09 (X.25) LINE NOT DEFINED: XX	69
ICS.10 to ICS.19 are reserved for more general messages	70-79
ICS.20 (X.25) INVALID COMMAND TO IBC FOR LOGICAL LINE XX	80
ICS.21 (X.25) INVALID PROTOCOL ID FOR LOGICAL LINE XX	81
ICS.22 (X.25) LAC BUS: UNMAPPED LINE INTERRUPT FOR LOGICAL LINE XX	82
ICS.23 (X.25) LAC BUS: ADDRESS PARITY ERROR FOR LOGICAL LINE XX	83

Table B-9 (continued)
Error/Warning Messages (LOGPRT/PRINT_NETLOG)

Error/Warning Message	Error Ref. Number
ICS.24 (X.25) LAC BUS: DATA PARITY ERROR FOR LOGICAL LINE XX	84
ICS.25 (X.25) LAC BUS: PARITY ERROR ON IA CYCLE FOR LOGICAL LINE XX	85
ICS.26 (X.25) UNMAPPED LINE ON DMX SCAN LIST FOR LOGICAL LINE XX	86
ICS.27 through ICS.35 reserved for ICS2/3 IBC errors	87-95
ICS.36 (X.25) BISYNC FRAMING ERROR FOR LOGICAL LINE XX	96
ICS.37 (X.25) STATUS BUFFER OVERFLOW FOR LOGICAL LINE XX	97
ICS.38 through ICS.51 - Reserved for ICS2/3 (Z8001) errors.	98-111
Not allocated.	112-119

Table B-10
Error/Warning Messages (RJE)

Error/Warning Message	Error Ref. Number
Error: ICS2/3 line in use.	120
Error: ICS2/3 received bad Config block.	121
Error: ICS2/3 has faulty IBC.	122
Error: ICS2/3 has incorrect LAC type.	123
Error: ICS2/3 has incorrect cable type.	124
Error: ICS2/3 has no cable attached.	125
Error: ICS2/3 has no LAC.	126
Error: ICS2/3 has incorrect IBC microcode.	127
Error: ICS2/3 has not enough memory.	128
Error: ICS2/3 detected internal protocol error (code = xxxx).	129
Error: Line xx: Error in Sending Config Block. ICS2EVNT.	130
Error: Line xx: Error in Dequeuing from IPQNM. Code = 58	131
Error: Line xx: Error in ICS2/3 Line Event. Code = nn	132
ERROR Nos. 133 to 149 are reserved for future RJE messages	

Table B-11
Error Messages (SDLC — SERVER)

Error Message	Error Ref. Number
Request failed because there is no controller with address <address>.	150
Request failed because there is insufficient ICS2/3 memory to support SDLC	151
Request failed because there are insufficient ICS2/3 buffers to support SDLC.	152
Request failed because of an imcompatibility between the SERVER and SDLC.	153
Request failed because the SDLC support is currently stopping.	154
Request failed because the SERVER and SDLC are not in synch.	155

Table B-12
Error Messages (SDLC — Remote System/Communications)

Error Message	Error Ref. Number
Remote system cannot be started : <remote-system-name> — <reason>	
cannot establish line signals	156
cannot raise line signals	157
fatal error occurred — code <nnnn>	158
incompatible ICS2 controller	159
incompatible LAC line number	160
insufficient ICS2/3 receive buffers	161
insufficient SDLC memory	162
insufficient SDLC table memory	163
invalid LAC type (RS232 or V.35)	164
no buffer to perform DMQ	165
no SDLC connections are available	166
SDLC didn't respond	167
the LAC cable is not connected	168
the LAC line does not exist	169
the line is already being used	170
unable to communicate with the ICS2/3	171
Reserved for additional SDLC error messages.	172-179

Table B-13
Error Messages (Asynchronous)

Error Message	Error Ref. Number
System console command only. (AMLC)	180
Line number is required. (AMLC)	181
Line number ooo is out of range, highest line number is 377 (AMLC)	182
protocol is not a valid protocol name. (AMLC)	183
Buffer ooo is a remote login buffer (is between lll and hhh) (AMLC)	184
Buffer number ooo is out of range, highest buffer number is hhh. (AMLC)	185
Line is already assigned to user ddd, must unassign first. (AMLC)	186
Buffer number bbb is already in use by line ooo. (AMLC)	187
Bad parameter. (ALCONF)	188
Operation unsuccessful. (ALCONF)	189
Device not available. (ALCONF)	190
Line ooo now supports user login, user number is ddd. (AMLC)	191
Line ooo is now an assignable line. (AMLC)	192
Inconsistent ASYNC cold start configuration for ICS2/3 device dd:	
an async line card has been found where not expected in slot ss.	193
faulty line card in slot ss where an async line card was expected.	194
sync line card in slot ss where an async line card was expected.	195

Table B-13 (continued)
Error Messages (Asynchronous)

Error Message	Error Ref. Number
Inconsistent ASYNC cold start configuration for ICS2/3 device dd:	
slot ss is empty where an async line card was expected.	196
ICS2/3 device dd has returned the wrong number (m) of status words.	197
Inconsistent ASYNC warm start configuration for ICS2/3 device dd:	
an async line card has been inserted into slot ss.	198
the async line card in slot ss in now inoperable.	199
the async line card in slot ss has been removed or is now inoperable.	200
a sync line card has been inserted into slot ss.	201
Bad key value (n) in call (ICSCFG).	202
Reserved for additional asynchronous messages.	203-220

C
Glossary

ACK
See Acknowledge.

Acknowledge (ACK)
Where error detection schemes are employed, a message sent in response to having received the transmission without error.

American Standard Code for Information Interchange (ASCII)
The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. Control characters and graphic characters comprise the ASCII set.

AMLC
See Asynchronous Multi-Line Controller.

ASCII
See American Standard Code for Information Interchange.

asynchronous communication
A serial stream of data sent as generated. Characters are delimited by start and stop bits whose function is to synchronize character bit timing.

Asynchronous Multi-line Controller (AMLC)
A Prime controller supporting 8 or 16 asynchronous lines. Each controller takes one backplane slot. Being replaced by the ICS1/2.

BCC
See Block Check Character.

BCD
See Binary Coded Decimal.

BHA
See Block Header Array.

Binary Coded Decimal (BCD)
The 8421 system represents each decimal digit with four binary digits with the place value of each bit equal to 8, 4, 2, or 1.

Binary Synchronous Communication (BSC)
A uniform procedure, using a standardized set of control characters and control character sequences, for synchronous transmission of binary coded data between stations in a data communication system. It is a byte/character oriented protocol.

BISYNC
See Binary Synchronous Communication.

bisynchronous
See Binary Synchronous Communication.

bit
The smallest quantity of data processing information. The bit can only assume the value of zero or of one.

bits per second
In serial transmission, the instantaneous bit speed with which a device or channel transmits a character.

blocks
Data is transferred to and from the host computer in blocks.

Block Check Character (BCC)
The Block Check Character(s) is dependent on line protocol and is either a Cyclic Redundancy Check (CRC) or a Longitudinal Redundancy Check (LRC/VRC) where VRC depends on parity.

Block Header Array (BHA)
An eight-word array associated with each block of memory in the free pool. The BHA contains the size, address, and other details of the block. The blocks are always referred to and maintained through their BHAs.

bps
See bits per second.

BSC
See Binary Synchronous Communication.

bus
A data transfer path over which information is transferred from any of several sources to any of several destinations.

byte
A term used to indicate a specific number of consecutive bits treated as a single entity. A byte consists of eight bits, which might represent one character or status information.

CCITT
See International Telegraph and Telephone Consultative Committee.

Central Processor Unit (CPU)
The central processor unit of a computer consists of the main storage memory, the arithmetic logic unit, control registers, and scratchpad memory. The CPU is thereby able to control the interpretation and execution of instructions, access memory, and perform arithmetic calculations.

Characters Per Second (CPS)
The number of values transferred per second.

CICS
See Customer Information Control System.

Clear to Send (CTS)
This is an RS232 (or V.24) control signal sent from the modem to the computer, indicating that the modem is ready to receive data from the computer. It sometimes indicates that the local phone is on or off the hook.

CLK
See clock.

clock
A component that provides a time base used in a transmission system to control the timing of certain functions.

clock signal
The output of a device that generates periodic signals used for synchronization. Synonymous with the term clock.

command
Sent from the PRIMOS operating system to the ICS2; can be one or more words.

CPS
See Characters Per Second.

CPU
See Central Processor Unit.

CRC
See Cyclic Redundancy Check.

CTS
See Clear to Send.

Customer Information Control System (CICS)
IBM's commonly used application program for monitoring communications.

Cyclic Redundancy Check (CRC)
An error detection scheme calling for the generation and verification of a check value using a polynomial calculation against the contents of the message, such as CRC-16 for bisync and CRC-CCITT for HDLC/SDLC.

Data Carrier Detect (DCD)
A modem signal indicating that the carrier is being received from the remote modem, which is thus able and ready to transmit.

data communications
The movement of data messages to and from remote systems through a medium.

Data Communications (or Circuit-terminating) Equipment (DCE)
The equipment installed at the user's premises that provide all the functions required to establish, maintain, and terminate a connection, and the signal conversion and coding between the Data Terminal Equipment (DTE) and the line.

data link
The physical means of connecting one location with another for the purpose of transmitting and receiving data in accordance with a link protocol.

Data Link Escape (DLE)
A bisync control character that signifies the start of transparent text (such as text with embedded control sequence).

Data Set Ready (DSR)
A control output from the DCE (for example, a modem), used to indicate to the DTE that the DCE is powered up and ready to transfer data.

Data Set Status (DSS)
Gives an indication of the modem status.

data terminal
The end user device used to access local and/or remote data processing systems.

Data Terminal Equipment (DTE)
That part of a data station that serves as a data source, data sink, or both, and provides for the data communication control function according to the used protocols.

Data Terminal Ready (DTR)
A control lead from the DTE, which indicates to the DCE that the terminal is ready.

DCD
See Data Carrier Detect.

DCE
See Data Communications (or Circuit-terminating) Equipment.

Device Interface Module (DIM)
A special PRIMOS process that is initialized at cold start; its primary purpose is to communicate with a specific type of hardware controller, for example, ICS1/ICS2.

DIM
See Device Interface Module.

Direct Memory Access (DMA)
The direct memory access mechanism transfers data directly between the IBC memory and the peripheral devices. The Z8001 is involved only in the setting up of the transfer; the transfer takes place with no processor intervention on a cycle stealing basis, under the control of the IBC.

Direct Memory Control (DMC)
A microcoded form of DMA — similar to DMQ.

Direct Memory Queue (DMQ)
A microcoded form of DMA, uses a control block in memory to control the source destination and length of transfer.

Direct Memory Transfer (DMT)
A high speed form of direct memory access.

DLE
See Data Link Escape.

DMA
See Direct Memory Access.

DMC
See Direct Memory Control.

DMQ
See Direct Memory Queue.

DMT
See Direct Memory Transfer.

DMX
Any direct-memory operation such as DMA, DMC, DMQ, or DMT.

downline load
The direct transfer of code from the host computer to the ICS2.

DSR
See Data Set Ready.

DSS
See Data Set Status

DTE
See Data Terminal Equipment.

DTR
See Data Terminal Ready.

EIA
See Electronics Industries Association.

Electronics Industries Association (EIA)
A trade association of the electronics industry which formulates technical standards, disseminates marketing data, and maintains contact with government agencies in matters relating to the electronics industry.

EOT
End-of-tape or End-of-transmission.

ESC
See Escape Character.

Escape Character (ESC)
A character code, the function of which is to control transmission/reception. Encoding schemes normally provide for at least one.

ETB
End of Transmission Block.

FAM
See File Access Manager.

FDX
See Full-duplex Transmission.

FIFO
First-in First-out memory stack.

File Access Manager (FAM)
The file access manager performs all remote file access automatically.

Full-duplex Transmission (FDX)
The mode of communication whereby simultaneous bidirectional transmission of data takes place.

Half-duplex Transmission (HDX)
The mode of communication whereby bidirectional transmission of data takes place, but only in a single direction at any one time.

hardwired
Logic that is built in by the manufacturer and is not subject to change by programming.

HASP
Houston Automatic Spooling Program, which is an IBM Bisync RJE protocol.

HDLC
See High Level Data Link Control.

HDX
See Half-duplex Transmission.

High Level Data Link Control (HDLC)
A bit-oriented protocol that adheres to the standards set by the CCITT and the International Standards Organization.

host computer
Any Prime 50 Series computer connected to an ICS2 controller.

HSSMLC
High Speed Synchronous Multi-line Controller. (An earlier version of the MDLC.)

IBC
See Inter-bus Controller.

ICS
Intelligent Communications Subsystem.

Interactive Subsystem.
PRIME/SNA's program that emulates most of the functions of the IBM 3270 Information Display System. It consists of four parts: 3270 LU Manager, terminal emulation, printer emulation, and configurator.

Inter-bus Controller (IBC)
A microcoded intelligent DMA controller that handles transfers on a maximum of 64 separate full-duplex communication lines.

International Telegraph and Telephone Consultative Committee (CCITT)
An international organization concerned with devising and proposing recommendations for international telecommunications.

Inter-process Queuing and Notification Mechanism (IPQNM)
A set of routines that create and use logical connections, consisting of high and low priority queues to transfer data between communicating processes.

interrupt
A signal or event that breaks into the normal flow of the system operation to enable a specific action to take place; the system operation resumes from the point of the break at a later time.

IPQNM
See Inter-process Queuing and Notification Mechanism.

ISQ
Input Steering Queue.

ITI
Interactive Terminal Interface.

KB
Kilobyte.

LAC
See Line Adaptor Card.

LAP
See Link Access Procedure.

LAPB
See Link Access Procedure — Balanced.

LCAD
Logical Connection Address.

LCB
Line Control Block.

LCCB
Logical Connection Control Block.

LCID
Logical Connection ID.

LCN
Logical Connection Number.

LED
Light Emitting Diode.

LIFO
Last-in First-out memory stack.

Line Adaptor Card (LAC)
A printed circuit board which provides the interface between asynchronous/ synchronous DTE and the LAC bus and the ICS2 controller. There are three types: ASYNC (RS232), SYNC (V.24) and SYNC (V.35)

line turnaround
IBM bisync terminology used to describe the procedure of enabling the receiver after transmission and vice versa, for example, in half-duplex mode, the action of switching the line between sending and receiving.

Link Access Procedure (LAP)
The Link Access Procedures (LAP and LAPB) are described as the link level element and are used for data interchange between a DCE and a DTE operating in user classes of service 8 to 11 as indicated in Recommendation X.1 [1] of the CCITT Yellow Book Vol. VIII — Fascicle VIII. 2. The procedures use the principles and terminology of the High Level Data Link Control (HDLC) procedures specified by the International Standards Organization (ISO).

Link Access Procedure — Balanced (LAPB)
See Link Access Procedure.

Logical Unit (LU)
A type of SNA network addressable unit concerned primarily with the handling of end-user or application data.

LSD
Least Significant Digit.

LU
See Logical Unit.

MDLC
See Multiple Data Link Controller.

modem (or data set)
See modulator-demodulator unit.

modulator-demodulator unit
A communication device which converts digital signals to analog and vice versa.

Multiple Data Link Controller (MDLC)
A single-board 2-line or 4-line Prime controller. It is microsequencer-controlled, supports V24 and V35 synchronous lines, and occupies one backplane slot. Optional PROMS select different protocols. Largely superseded by the ICS1/2/3.

multiplexing
Combining several communications channels into one for transmission, and then separating them (demultiplexing) after receipt.

NRZI
See Non-return to Zero Inverted.

Non-return to Zero Inverted (NRZI)
A signalling technique, used in SDLC, in which a zero or a one bit is indicated by the absence or presence of a transition on the line rather than a current state.

OSQ
Output Steering Queue.

output string
A character sequence passed from PRIMOS to the ICS2 in a buffer ready for output. This term is used to avoid any confusion with block.

parity
The integrity of each character transmitted over a communications link can be tested by the generation and subsequent checking of character parity. Parity is generated by adding an extra bit causing the result of the addition of the bit to be even or odd. The ICS2 uses even parity checking for bisync ASCII RJE with LRC block check.

PDN
Public Data Network.

Peripheral Node
IBM's name for a node that requires support from a host or communications controller to use SNA network facilities.

PIO
Programmed Input/Output.

POLARIS
Prime online error reporting.

PRIMENET
Prime's X.25-based networking.

PROM
Programmable Read-only Memory.

protocol
A set of conventions, or rules, between communicating processes on the format and the content of messages to be exchanged. To make implementation and usage more convenient, in sophisticated networks, higher-level protocols may use lower-level protocols in a layered fashion.

PSN
Public Switched Network.

QCB
Queue Control Block.

queue
A line of items waiting for attention.

RAM
Random Access Memory.

Remote Job Entry (RJE)
Prime's remote job entry emulator products allow Prime systems to emulate IBM, CDC, Honeywell, Univac, and ICL remote job entry terminals. This enables communications between a Prime system and a host (mainframe) computer. Communications are usually accomplished via the transfer of files from one system to another.

Request to Send (RTS)
A control output from the DTE used to indicate to the DCE that the terminal device would like to send data.

reset
Clears to zero (false).

response
One word containing status information sent to PRIMOS from the ICS2.

response time
The interval between executing a command or inquiry at a terminal and the receipt of a response at the same terminal.

RI
See Ring Indicator.

Ring Indicator (RI)
An indicator that warns an operator of an incoming call on a dial-up line, requiring action.

RINGNET
Prime's token ring system.

RJE
See Remote Job Entry.

RJI
RJE software interface between the ICS2 and PRIMOS.

RJOP
The RJE operator interface.

RJQ
The RJE queue interface.

RNR
Receive Not Ready.

ROM
Read Only Memory.

RR
Receive Ready.

RS-232
A defined standard interface between data terminals and data communication devices using serial data transfers.

RTS
See Request to Send.

SCC
Serial Communications Controller, such as the Zilog Z8530.

SDF
See Site Definition File.

SDLC
See Synchronous Data Link Control.

SDLMAN
See Synchronous Data Link Manager.

SDLMTR
The interrupt handler for the Synchronous Data Link Manager (SDLMAN).

serial transmission
Transmission of one bit at a time.

Server Subsystem
PRIME/SNA's main processer, or kernel, that performs the following: control of up to four SDLC lines, support for up to eight remote systems, basic support for a peripheral node's logical units, and administration and configuration of the peripheral nodes.

session
The time between a line's being configured and deconfigured.

set
Sets bit to one (true).

SIO
Serial input/output (predecessor to the Zilog 8530 SCC).

Site Definition File (SDF)
Defines the RJE data of the remote RJE site.

SMLC
See Synchronous Multi-Line Controller.

SNA
See Systems Network Architecture.

start bit
Serial asynchronous data transmission relies upon the start bit to signify to the receiver that a character follows. The start of a character is typically signified by two start bits. The start bits allow the receiver to achieve synchronization with the transmitter before receiving the data character.

stop bit(s)
Serial asynchronous data transmission relies upon the stop bit(s) to signify to the receiver that no more data bits follow.

SYN
Synchronous idle. A transmission control character used by a synchronous transmission system, in the absence of any other character (idle condition) to provide a signal from which synchronization may be achieved or retained between data terminal equipment.

Synchronous Data Link Control (SDLC)
An IBM computer networking discipline for managing synchronous, code-transparent, serial information transfer over a link connection. Transmission exchanges may be full-duplex or half-duplex over switched or unswitched links.

Synchronous Data Link Manager (SDLMAN)
SDLMAN, the synchronous data link manager, is the SDLC protocol processor in PRIME/SNA, and is responsible for the linking of the SDLMTR processes and the SNA software in PRIMOS.

Synchronous Multi-line Controller (SMLC)
An earlier version of the multiple data link controller.

Systems Network Architecture (SNA)
IBM's description of the logical structure, formats, protocols, and operational sequences for transmitting information units through networks and controlling the configuration and operation of those networks.

Time Share Option (TSO)
IBM's interactive software development tool for MVS operating system users.

TSO
See Time Share Option.

T$SLC0
Block structured interface for multiple data link controller.

transparency
A data communication mode that enables equipment to send and receive bit patterns of any form, without regard to interpretation as control characters.

UA
Unnumbered Acknowledge.

UART
See Universal Asynchronous Receiver Transmitter.

Universal Asynchronous Receiver Transmitter (UART)
A semiconductor device that provides flexible asynchronous communication support.

Universal Synchronous/Asynchronous Receiver Transmitter (USART)
A semiconductor device designed to provide flexible synchronous or asynchronous communication support.

uP
Microprocessor.

USART
See Universal Synchronous/Asynchronous Receiver Transmitter.

VC
See Virtual Circuit.

VCID
Virtual Circuit ID.

VCN
Virtual Circuit Number.

Virtual Circuit (VC)
Logical connection of communicating devices for the duration of transmission.

virtual memory
Increased effective memory capacity that results from paging programs/data between primary and secondary storage.

Virtual Terminal Access Method (VTAM)
IBM's main data communications access method used in SNA networks.

VTAM
See Virtual Terminal Access Method

V.35
The CCITT recommendation for the transmission of data over physical links at data rates in excess of 48 kilobits per second.

WACK
Wait Acknowledgment.

word
Consists of 16 bits. The bit numbering system used is the Z8001 notation, where bit 0 is the least significant bit.

X.25
The CCITT recommendation defining the interface between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for terminals operating in the packet mode on public data networks.

Z8001
A 16-bit Zilog microprocessor used in the ICS2.

INDEX

Index

SURVEY

READER RESPONSE FORM

DOC10094-1LA ICS User's Guide First Edition

Your feedback will help us continue to improve the quality, accuracy, and organization of our user publications.

1. How do you rate the document for overall usefulness?

___excellent ___very good ___good ___fair ___poor

2. Please rate the document in the following areas:

Readability: ___hard to understand ___average ___very clear

Technical level: ___too simple ___about right ___too technical

Technical accuracy: ___poor ___average ___very good

Examples: ___too many ___about right ___too few

Illustrations: ___too many ___about right ___too few

3. What features did you find most useful? _____

_____

_____

_____

4. What faults or errors gave you problems? _____

_____

_____

_____

Name: _____ Position: _____

Company: _____

Address: _____

_____ Zip: _____

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

First Class Permit #531 Natick, Massachusetts 01760

BUSINESS REPLY MAIL

Postage will be paid by:

PRIME

Attention: Technical Publications
Bldg 10B
Prime Park, Natick, Ma. 01760

READER RESPONSE FORM

DOC10094-1LA ICS User's Guide First Edition

Your feedback will help us continue to improve the quality, accuracy, and organization of our user publications.

1. How do you rate the document for overall usefulness?

___excellent ___very good ___good ___fair ___poor

2. Please rate the document in the following areas:

Readability: ___hard to understand ___average ___very clear

Technical level: ___too simple ___about right ___too technical

Technical accuracy: ___poor ___average ___very good

Examples: ___too many ___about right ___too few

Illustrations: ___too many ___about right ___too few

3. What features did you find most useful? _____

_____

_____

_____

4. What faults or errors gave you problems? _____

_____

_____

_____

Name: _____ Position: _____

Company: _____

Address: _____

_____ Zip: _____

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

First Class Permit #531 Natick, Massachusetts 01760

BUSINESS REPLY MAIL

Postage will be paid by:

PRIME

Attention: Technical Publications
Bldg 10B
Prime Park, Natick, Ma. 01760

READER RESPONSE FORM

DOC10094-1LA ICS User's Guide First Edition

Your feedback will help us continue to improve the quality, accuracy, and organization of our user publications.

1. How do you rate the document for overall usefulness?

___excellent ___very good ___good ___fair ___poor

2. Please rate the document in the following areas:

Readability: ___hard to understand ___average ___very clear

Technical level: ___too simple ___about right ___too technical

Technical accuracy: ___poor ___average ___very good

Examples: ___too many ___about right ___too few

Illustrations: ___too many ___about right ___too few

3. What features did you find most useful? _____

_____

_____

_____

4. What faults or errors gave you problems? _____

_____

_____

_____

Name: _____ Position: _____

Company: _____

Address: _____

_____ Zip: _____

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

First Class Permit #531 Natick, Massachusetts 01760

BUSINESS REPLY MAIL

Postage will be paid by:

PRIME

Attention: Technical Publications
Bldg 10B
Prime Park, Natick, Ma. 01760