Burroughs

A Series Physical I/O Overview

(Relative to the Mark 3.6 System Software Release)

Distribution Code SE

Priced Item
Printed in U.S.A.
December 1985

1169943

Comments or suggestions regarding this document should be submitted on a Field Communication Form (FCF) with the Class specified as "2" (System Software), the Type specified as "1" (F.T.R.), and the Product specified as the 7-digit form number of the manual (for example, "1169943"). The FCF should be sent to the following address:

Burroughs Corporation
PA&S/Orange County
19 Morgan
Irvine, CA 92718

CONTENTS

1 INTRODUCTION

This manual provides a high-level overview of the Physical Input/Output (I/O) function on Burroughs computer systems that support Universal I/O (UIO) on Message-Level Interface Processor (MLIP) systems (A 3, A 3K, A 9, A 10, B 5900, and B 6900). This manual is primarily intended as an introduction for systems programmers who will later study the software implementation of Physical I/O, but it may also be of interest to those for whom an overview of the major components involved in the I/O process is sufficient. It is assumed that the reader is familiar with the basic structure and function of the Master Control Program (MCP) and the New Programming (NEWP) language, although knowledge of the details of the MCP and of the machine architecture is not required.

The PHYSICALIO module of the MCP is the central component in the Physical I/O process. There are four versions of this module, the appropriate one of which is selected during system initialization according to the type of I/O subsystem the machine supports:

1. MLIP (A 3, A 3K, A 9, A 10, B 5900, and B 6900 systems)

2. Multiplexor (B 6800 systems)

3. I/O Module (B 7700 and B 7800 systems)

4. Host Data Unit (HDU) (A 15 and B 7900 systems)

The four versions of the PHYSICALIO module have the same interface to the rest of the MCP, freeing the majority of the MCP from having to handle the idiosyncrasies of four different I/O subsystems. Only the MLIP module version is described in this manual.

The systems using the MLIP I/O subsystem are divided into two groups:

1. The A 3, A 9, B 5900, and B 6900 systems

2. The A 3K and A 10 systems

The basic I/O process is the same for these systems, but some of the procedures, mechanisms, and data structures used by the MLIP I/O subsystem for each group are different. The systems are identified when a distinction must be made between the two groups of systems.

The A 3K system has a dual processor configuration but can function with only one processor working. Other A 3 models are single-processor systems.

PHYSICAL I/O OVERVIEW

NOTE

This document is not intended to be a complete specification of the Physical I/O implementation for MLIP systems. Most specific details and exception conditions have been omitted for brevity and to better highlight the general structure of the I/O software and hardware. Although the overall design is stable, the details of the implementation are subject to change from Mark Release to Mark Release.

STRUCTURE OF THIS MANUAL

Sections

1 INTRODUCTION

The purpose of this overview is described, and the PHYSICALIO module versions are listed.

2 SYSTEM OVERVIEW

The general I/O path between the MCP and the UIO subsystem is described and illustrated.

3 MCP REQUESTORS

The MCP modules that interface with PHYSICALIO to initiate I/Os and to request unit information are described.

4 REQUESTOR/PHYSICALIO INTERFACE

Input/Output Control Block (IOCB) and unit interfaces that exchange information between the MCP and the PHYSICALIO module are described.

5 PHYSICALIO MODULE

PHYSICALIO's responsibilities for I/O processing, exception handling, and responding to unit inquiries are described.

6 PHYSICALIO/MLIP INTERFACE

The mechanisms that are used to communicate between PHYSICALIO and the MLIP are described. The mechanisms and data structures described are IOCBs, Communicate with Universal I/O (CUIO) operator, Signal Processing Element Set (SPES) operator, IOCB queue, command queue, unit queue, MLIP I/O Table, horizontal queue, result queue, and I/O Finish interrupt.

7 MLIP

The functions of the MLIP are listed and MLIP commands are described.

PHYSICAL I/O OVERVIEW

8 MLIP/UIO INTERFACE

The Message-Level Interface (MLI) (the mechanism used to allow communication between the MLIP and the UIO subsystem) and the actions performed when an I/O is transferred to a DLP are described.

9 UIO SUBSYSTEM

The rules and conventions used to establish a valid UIO subsystem configuration are described. The host, UIO base, Data Link Processor (DLP), unit, and path are identified as parts of a UIO subsystem configuration.

10 SYSTEM INITIALIZATION

The initialization procedures for the UIO subsystem are described.

11 PERIPHERAL STATUS

The method of monitoring a peripheral's status is described.

12 DATA COMMUNICATIONS

Data communication handled by special-purpose DLPs--Network Support Processors (NSPs), Line Support Processors (LSPs), and Data Communications Data Link Processors (DC-DLPs)--in the I/O subsystem is described.

Appendix

A glossary appears at the end of this manual.

RELATED DOCUMENTS

Document	Form No.
Operator Display Terminal (ODT) Reference Manual	1169612

2 SYSTEM OVERVIEW

The PHYSICALIO module is the Master Control Program's (MCP) interface to the I/O subsystem. PHYSICALIO provides two notable functions relating to the I/O subsystem: the handling of I/Os and the management of units. Units are peripheral devices such as printers, magnetic tape drives, card readers, and so on.

MAJOR COMPONENTS

Figure 1 shows the basic flow of information between the MCP and the I/O subsystem:

```
               - - - - - - - - - - - - -
             /                           \
            |          MCP Requestors     |
             \                           /
               - - - - - - - - - - - - -
                        |     A
                        |     |
              <Requestor/PHYSICALIO Interface>
                        |     |
                        V     |
      - - - - - - - - - - - - - - - - - - - - - - - -
     |                                                |
     |               PHYSICALIO Module                |
     |                                                |
      - - - - - - - - - - - - - - - - - - - - - - - -
                        |     A
                        |     |
                <PHYSICALIO/MLIP Interface>
                        |     |
                        V     |
      - - - - - - - - - - - - - - - - - - - - - - - -
     |                                                |
     |  Message-Level Interface Processor (MLIP)      |
     |                                                |
      - - - - - - - - - - - - - - - - - - - - - - - -
                        |     A
                        |     |
              <Message-Level Interface (MLI)>
                        |     |
                        V     |
               - - - - - - - - - - - - -
             /                           \
            |  Universal I/O (UIO) Subsystem |
             \                           /
               - - - - - - - - - - - - -
```

Figure 1. Information Flow between MCP and UIO Subsystem

PHYSICAL I/O OVERVIEW

Most I/Os follow this standard path through the system:

1. An MCP requestor calls the PHYSICALIO module to perform a specific I/O operation.

2. The PHYSICALIO module builds a data structure representing the operation to be performed and calls the MLIP.

3. The MLIP interprets the data structure, generates an I/O request, and passes it to the UIO subsystem.

The results of the I/O follow the reverse path, from the UIO subsystem to the MLIP to PHYSICALIO to the requestors.

Each of the components and interfaces shown in Figure 1 will be discussed later in the manual. However, because the structure and terminology of the UIO subsystem may be unfamiliar to some readers, a brief functional description of the UIO subsystem is presented here.

UIO FUNCTIONAL OVERVIEW

Figure 2 illustrates the configuration of components used to pass data from the MLIP to the UIO bases.

```
            |-------------------|
            |                   |
            |     Processor     |
            |                   |
            |===================|
            |       MLIP        |
            |-------------------|
            |Port|Port|Port|Port|
              |    |    |    |
             MLI
              |
          |--|--|       <--|
          |     |          |
          |-----|          |
          |     |          |
          |-----|          B
          | DLP |          A
          |-----|          S
          | DLP |          E
          |-----|          |
          | DLP |          |
          |-----|          |
          |     |          |
          |-----|       <--|
```

Figure 2. MLIP-to-UIO-Base Configuration

Each processor uses an MLIP and associated hardware to pass an I/O to a peripheral. The following is a typical I/O path from an MLIP to a DLP.

1. MLIP

2. MLI ports

3. MLI

4. UIO base

5. Data Link Processors (DLPs)

Each base contains one or more DLPs, which can connect either directly to the peripheral or to a peripheral controller (depending on the type of peripheral). In addition, each base includes the components necessary to route messages between the DLPs and the host.

There is, in general, a different type of DLP for each type of peripheral. In Figure 3, a Train Printer DLP (TP-DLP) is shown connected to a train printer, a Card Reader DLP (CR-DLP) is shown connected to a card reader, and a Host Transfer DLP (HT-DLP) is shown connected to a Disk Pack Drive Controller (DPDC).

```
  |-------------------|
  |                   |
  |     Processor     |
  |                   |
  |===================|
  |       MLIP        |
  |-------------------|
  |Port|Port|Port|Port|
    |    |    |    |
   MLI
    |
|--|---|
|      |
|------|
|      |
|------|
|TP-DLP---Train Printer
|------|
|CR-DLP---Card Reader
|------|
|HT-DLP----|-----------|
|------|   |   DPDC    |
|      |   |-----------|
|------|   | Exchange  |
           |-|-|-|-|-|-|
             | | | | |
            Disk Units
```

Figure 3. MLIP-through-DLP Configuration

Figures 2 and 3 are schematic and do not necessarily represent actual configuration possibilities, restrictions, or limits. A more detailed description of UIO subsystem configurations is contained in the "UIO Subsystem" section of this manual.

See also
UIO Subsystem 73

3 MCP REQUESTORS

The PHYSICALIO module provides the following services to the rest of the Master Control Program (MCP): maintaining unit information, initiating I/Os, monitoring peripheral status, and initializing Data Link Processors (DLPs). This section describes the MCP modules that interface with PHYSICALIO to initiate I/Os and to request unit information. Peripheral status and system initialization are described in later sections.

See also
Peripheral Status 91
System Initialization 87

I/O REQUESTORS

All I/Os required by the MCP, except some I/Os performed early in the system initialization process, are initiated by calling the PHYSICALIO module. The programmatic mechanism of communication between the requesting modules shown in Figure 4 and the PHYSICALIO module is described in the next section.

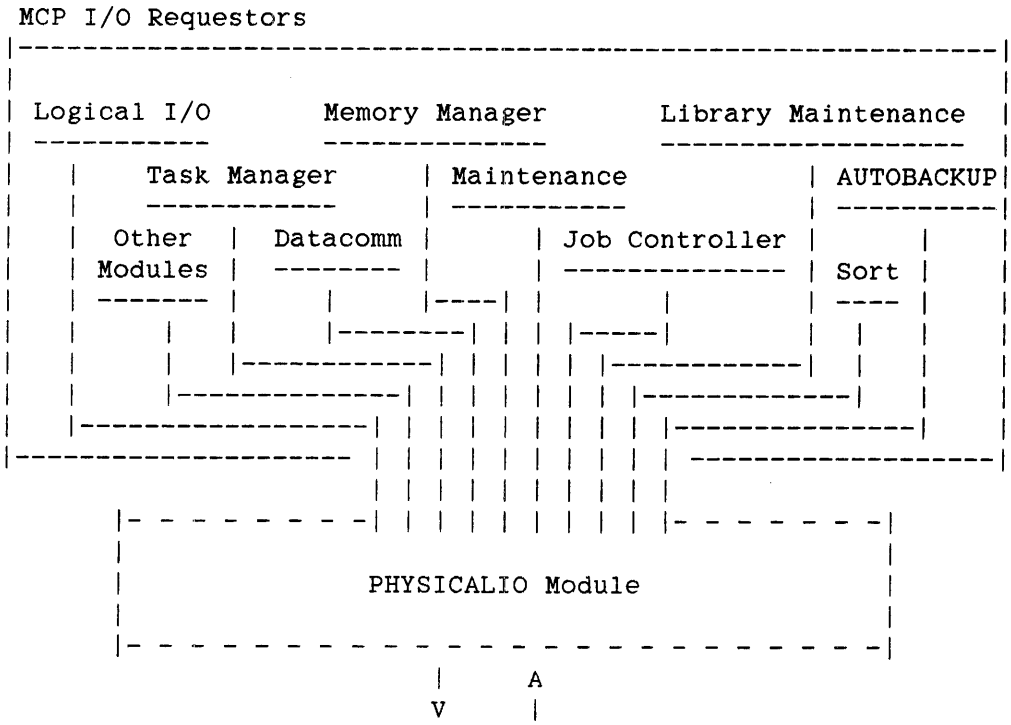

Figure 4. PHYSICALIO and Requesting Modules

PHYSICAL I/O OVERVIEW

Although there are many modules that call PHYSICALIO, PHYSICALIO views these requestors as members of certain categories or sets. These sets are based on the distinctions required to properly handle the I/O request, not on the identity or function of the module requesting the I/O. Some distinctions that PHYSICALIO makes based on the type of requestor are whether or not to allow access to privileged information or operations, whether or not to attempt error recovery, and whether or not to log errors.

As far as PHYSICALIO is concerned, all I/Os fall into one of four main I/O requestor sets:

I/O Set	Description
User	I/Os initiated on behalf of a user program. Logical I/O and library maintenance are two areas in the MCP that request user I/Os.
MCP	The general category of I/Os initiated to perform MCP functions. Memory management, SWAPPER, Datacomm, and AUTOBACKUP request MCP I/Os, as do many other functions in the MCP.
Maintenance	I/Os initiated primarily by the maintenance module to perform I/O hardware diagnostic functions.
Kernel	I/Os initiated by PHYSICALIO itself to perform such functions as retrying errors and determining peripheral status.

Because a value representing the type of requestor is passed to PHYSICALIO with each I/O request, a requesting module can initiate I/Os with different requestor values for different purposes. For example, the maintenance module initiates maintenance I/Os when it performs diagnostic tests and user I/Os (through logical I/O) when it displays the results of those tests.

The four main I/O requestor sets (described above) comprise several specific requestor values. These values allow PHYSICALIO to make more specialized distinctions when desirable and can be regrouped into additional sets.

MCP Requestors

When the distinction between user I/Os and MCP I/Os is important (for example, in exception handling), PHYSICALIO tests the requestor value for membership in either the user I/O set or the MCP I/O set.

When special handling is required for library maintenance I/Os (for example, in retrying parity errors), PHYSICALIO tests the requestor value for membership in the library maintenance I/O set.

PHYSICAL I/O OVERVIEW

UNIT INFORMATION REQUESTORS

Many of the same modules that request I/Os to be performed also require information about and/or some control over the units to which the I/Os are sent. Some of these requestors are Logical I/O, Maintenance, Datacomm, and AUTOBACKUP. In addition, some modules are functionally related to PHYSICALIO in that they maintain information about the status of units and the configuration of the I/O subsystem. Much of this information is "logical," as opposed to "physical," in nature and, thus, is appropriately maintained outside of PHYSICALIO. Three modules that maintain such information are described below.

FILEFINDER Module

The FILEFINDER module locates an input or output device upon request. FILEFINDER maintains the UNIT table, which contains data about each unit. Although most of the information is updated during the performance of other logical functions (such as reading labels, assigning units to tasks, and reserving units), information pertaining to the current physical status of a unit is obtained by calling PHYSICALIO.

UNITID Module

The UNITID module reads and writes labels on labeled units. These functions require that UNITID request unit information from PHYSICALIO. UNITID maintains the UINFO table, which contains label information for each unit.

MCPSTATUS Module

The MCPSTATUS module displays and alters unit information through the SYSTEMSTATUS, GETSTATUS, and SETSTATUS interfaces. These functions require that MCPSTATUS call PHYSICALIO to obtain and to alter unit-oriented information.

Other functional areas that require unit information include disk allocation, disk file management, directory control, and configuration control.

4 REQUESTOR/PHYSICALIO INTERFACE

The Master Control Program (MCP) requestors exchange information with the PHYSICALIO module through a group of interface procedures that are exported by the PHYSICALIO module. The names of these procedures are identical for all four PHYSICALIO version modules, so that the majority of the MCP is not required to distinguish between the possible I/O subsystems.

The two primary categories of interface procedures are Input/Output Control Block (IOCB) interfaces and unit interfaces.

IOCB INTERFACE PROCEDURES

The I/O request interface between PHYSICALIO and the Message-Level Interface Processor (MLIP) is a data structure called an IOCB. PHYSICALIO allocates and manages IOCBs, which allows the format of IOCBs and the mechanism for their allocation to vary without impact on the rest of the MCP. Some of the information required in the IOCB is provided by the requestor through interface procedures and some is inserted by PHYSICALIO before actually starting the I/O. After processing the I/O, the MLIP places result information into the IOCB; PHYSICALIO interprets this information and generates a logical result descriptor, which is accessible to the requestor through an interface procedure.

As far as the requestor is concerned, there are three stages in the I/O process:

1. Allocating the IOCB
2. Performing one or more I/Os using that IOCB
3. Deallocating the IOCB

Performing I/Os involves providing the proper parameters to PHYSICALIO and acting on the result information. The following pages describe the interface procedures provided for IOCB-related actions.

PHYSICAL I/O OVERVIEW

Setting Up an IOCB

IOCBs are allocated when the requestor calls BUILDIOCB. BUILDIOCB stores a "mom" descriptor of the new IOCB in a location specified by a reference passed as a parameter to BUILDIOCB. To initiate the I/O once the IOCB has been allocated, the requestor must provide PHYSICALIO with the following information:

- A reference to the IOCB
- An I/O Control Word (IOCW) describing the operation to be performed
- A buffer for the data
- An offset into the buffer indicating the start of the area available for possible data transfer
- The length of the area available for possible data transfer
- A mask with bits set to indicate the exception conditions under which PHYSICALIO is not to attempt recovery
- The unit number of the unit to which the I/O is to be directed
- A value representing the requestor type

If the I/O is to be performed asynchronously (that is, while the requestor continues executing), another piece of information is required:

- A reference to an event to be caused when the I/O has finished

If the I/O is a user I/O, an additional parameter is required:

- A reference to the File Information Block (FIB) of the IOCB

Because information changes from one I/O to the next in most cases, all of this information is passed to PHYSICALIO when I/O initiation is requested. However, in some cases, most notably for normal (nondirect) user I/Os requested through logical I/O, the information is relatively constant from one I/O operation to the next. Several interface procedures are provided to allow many of the I/O parameters to be set before the I/O initiation is requested; thus, information for the IOCB is transferred only when it changes.

Requestor/PHYSICALIO Interface

For a requestor such as logical I/O, the interface procedures called "SET<item>" (such as SETIOMASK) would be called between I/Os to alter only the IOCB information that must change before the next I/O. The I/O initiate procedure for these I/Os (described in "Initiating an I/O") requires very few parameters because most of the information has already been stored in the IOCB.

Initiating an I/O

Many interface procedures are provided for initiating I/Os. The most important characteristics of each procedure can be readily identified by each procedure name.

The first part of each procedure name is either INITIATE or DO. INITIATE procedures initiate an I/O asynchronously; that is, control is returned to the requestor after the I/O is initiated without waiting for the I/O to finish. DO procedures initiate an I/O and wait until the I/O finishes before returning control to the requestor.

The second part of each procedure name contains one or more of the following keywords:

Keyword	Type of I/O
CHAR	A character-oriented I/O
WORD	A 48-bit-word-oriented I/O
MEMORY	A 51-bit-word-oriented I/O
USER	A user I/O
MAINTENANCE	A maintenance I/O
DIRECT	A direct I/O
PSEUDO	An I/O not actually to be performed; the requestor is calling PHYSICALIO just for bookkeeping purposes
KERNEL	A kernel I/O

"IO" is appended to complete the procedure name.

PHYSICAL I/O OVERVIEW

Not all combinations of these keywords represent actual interface procedures because many of the combinations are not required. The following identifiers are examples of I/O initiation interface procedures: DOCHARIO, DOMEMORYIO, INITIATEUSERIO, INITIATEUSERDIRECTIO, INITIATEDIRECTPSEUDOIO.

Accessing Information in an IOCB

The information in an IOCB cannot be accessed or changed while the I/O is in process. A Boolean interface procedure called IOINPROCESS allows the requestor to determine whether or not the I/O is still active.

If the I/O is not in process, various items stored in the IOCB can be requested through the "GET<item>" interface procedures. For example, GETIOTIME returns the I/O time stored in the IOCB. GETLOGICALRESULT returns a "soft" result descriptor that is common for all I/O subsystems. This result is similar in content to the STATE file attribute.

Deallocating an IOCB

An IOCB is deallocated by calling the interface procedure FORGETIOCB.

Requestor/PHYSICALIO Interface

UNIT INTERFACE PROCEDURES

The unit interface procedures allow PHYSICALIO and the requestors to exchange information about the status and visibility of the I/O units. Information about a particular unit is requested by specifying a unit number, as described below.

Each unit has two unique unit numbers: a logical unit number and a physical unit number. Logical unit numbers are used within the MCP for two major reasons:

1. Logical unit numbers refer only to units for which there is a Data Link Processor (DLP) physically present (rather than referring to the entire tree of addressable units). This makes the UNIT tables much smaller than they otherwise might be.

2. Logical unit numbers allow a physical unit to "move" without affecting its logical connection to the MCP.

Physical unit numbers are used when the MCP communicates with the system operator or the I/O subsystem. PHYSICALIO maintains the correspondence between logical and physical unit numbers.

The following paragraphs describe some of the major unit interface procedures.

TAKEUNIT and GIVEUNIT

TAKEUNIT and GIVEUNIT are important interfaces for coordinating the transfer of a unit's logical control from PHYSICALIO to its requestors (TAKEUNIT) and from the requestors back to PHYSICALIO (GIVEUNIT).

For single-user devices (for example, train printers), TAKEUNIT is called when the unit is assigned to a task and GIVEUNIT is called when the unit becomes unassigned. For multiple-user devices (for example, disk packs), TAKEUNIT is called before the label is read and GIVEUNIT is called in response to an Operator Display Terminal (ODT) command (such as CLOSE and UR).

Two of the parameters to GIVEUNIT indicate whether or not PHYSICALIO monitors peripheral status for the unit and, if so, whether or not the unit is logically ready. The initiation of peripheral status monitoring is described in the "Peripheral Status" section.

GETUNITINFO and SETUNITINFO

These procedures allow the requestors to access and, in some cases, alter various items of unit information. For example, GETUNITINFO will return the unit's subtype, an indication of whether or not there is a path to the unit, or the unit's reliability factor, depending on the item requested by the procedure's selection parameter.

GETLOGICALUNITSTATUS

This procedure returns device-dependent information about the unit; for a train printer, for example, GETLOGICALUNITSTATUS will indicate whether or not a TRAIN table has been loaded, and for a magnetic tape, whether or not the tape is rewinding.

See also
Peripheral Status 91

5 PHYSICALIO MODULE

The PHYSICALIO module participates in system initialization by creating and maintaining the configuration tables describing the I/O subsystem. This topic is described in the "System Initialization" section. PHYSICALIO also monitors peripheral status, which is discussed in the "Peripheral Status" section. PHYSICALIO's responsibilities in the areas of I/O processing, exception handling, and responding to inquiries about units are described in the following paragraphs.

PROCESSING OF I/Os

The PHYSICALIO module provides two methods of I/O initiation for the Message-Level Interface Processor (MLIP) I/O subsystem.

1. One method supports the A 3, A 9, B 5900, and B 6900 systems.

2. The other method supports the A 3K and A 10 systems.

Both methods of I/O initiation are discussed in this section. The systems are identified when the I/O processing method is unique to a systems group.

At initialization time, the Master Control Program (MCP) interrogates the result of the WATI operator to determine which I/O method is used on the system. The WATI operator identifies the system in use and provides implementation-level information about the system.

The result of the WATI operator determines which I/O initiation method to use for the system. The I/O initiation method in turn determines whether or not the Communicate with Universal I/O (CUIO) operator is valid for the system. The CUIO operator is used to pass the address of an Input/Output Control Block (IOCB) to an MLIP for initiation. The CUIO operator is used in the A 3, A 9, B 5900, and B 6900 systems and is described in "CUIO Operator." On A 3K and A 10 systems, an I/O is issued by queuing an IOCB to an IOCB queue and using the Signal Processing Element Set (SPES) operator to signal the MLIP. The SPES operator is described in the "PHYSICALIO/MLIP Interface" section.

PHYSICAL I/O OVERVIEW

I/O Processing for A 3, A 9, B 5900, and B 6900 Systems

The I/O processing method for A 3, A 9, B 5900, and B 6900 systems through the PHYSICALIO module is described by the following example. INITIATECHARIO was chosen as the interface procedure for the I/O initiation example because it is a commonly used interface for MCP I/Os. At decision points in the processing, the most straightforward choice is taken. For example, it is assumed here that the I/O was completed without exceptions (exception handling is described in "Exception Handling").

Example

1. After allocating an IOCB through BUILDIOCB, the requestor calls INITIATECHARIO, passing in the following required parameters:

   a. The IOCB

   b. A reference to the event to be caused when the I/O finishes

   c. The requestor type

   d. An I/O exception mask

   e. An Input/Output Control Word (IOCW)

   f. A buffer

   g. An index into the buffer

   h. An I/O length

   i. A unit number

2. INITIATECHARIO marks the IOCB "in process," stores the parameters in the IOCB, and calls INITIATEIO.

3. INITIATEIO coordinates the remainder of the I/O initiation processing by calling SETUPIOCB, GETPATH, and FIREIOCB.

4. SETUPIOCB converts the IOCW into the appropriate operation code for the Universal I/O (UIO) subsystem and sets up the information required for the proper queuing of the IOCB by the MLIP.

5. GETPATH selects a path to the unit and inserts the path information into the IOCB.

6. FIREIOCB marks the IOCB "active" and gives the IOCB to the MLIP through the CUIO operator.

The I/O is now in progress. The following actions occur when the MLIP informs the MCP that the I/O has been completed.

7. HARDWAREINTERRUPT is invoked when the I/O Finish interrupt is received from the MLIP; HARDWAREINTERRUPT calls IOFINISH68.

   Because the I/O Finish interrupt does not designate which I/O has completed, IOFINISH68 searches through the result queues to find completed I/Os. When one is found, IOFINISH68 selects the appropriate routine to handle the completed I/O; in most cases (in particular, when there are no exceptions reported), it calls FINISHIO.

8. FINISHIO determines whether or not it is running on the processor that initiated the I/O. If so, it calls FINISHOFFIO.

9. FINISHOFFIO inserts into the logical result descriptor a value indicating the number of units transferred, computes and bills the I/O time, sets the state of the IOCB to "inactive," and causes the I/O completion event, a reference to which was passed as a parameter to the interface procedure.

At this point, the requestor has been notified that the I/O has completed and it can now access result information in the IOCB.

See also
I/O Finish Interrupt. 62

I/O Processing for A 3K and A 10 Systems

The I/O processing method for A 3K and A 10 systems is similar to the method described above, except that the procedure names are different and additional procedures were created. The procedures that perform the same functions as the method above are INITIATEIO_ASIO, FIREIOCB_ASIO, and IOFINISH_ASIO. INITIATEDATACOMIO_ASIO and PATHRES_ASIO are new procedures that perform ancillary functions for data communication operations and path reservation.

In the following example, the requestor calls INITIATECHARIO for I/O initialization. As in the example above, the same parameters are passed to the allocated IOCB.

Example

1. INITIATECHARIO marks the IOCB in process, stores the parameters in the IOCB, and calls INITIATEIO_ASIO.

2. INITIATEIO_ASIO coordinates the remainder of the I/O initiation processing by calling SETUPIOCB. The path to the peripheral device is either specified by GETPATH or determined by the optimization defines. When the path is selected, FIREIOCB_ASIO is called.

3. SETUPIOCB converts the IOCW into the appropriate operation code for the UIO subsystem and sets up the information required for the proper queuing of the IOCB by the MLIP.

4. The path to a peripheral device can be selected by the MCP through GETPATH, or if the Simple Path Selection field in the MLIP Control Word of the IOCB is set to TRUE, then the MLIP uses the path specified in the Path Table Pointer word of the unit queue. If GETPATH is used, the path information is inserted into the IOCB.

5. FIREIOCB_ASIO marks the IOCB "active" and gives the IOCB to the MLIP by queuing it in the IOCB queue.

The I/O is now in progress. The following actions occur when the MLIP informs the MCP that the I/O has completed.

6. HARDWAREINTERRUPT is invoked when the I/O Finish interrupt is received from the MLIP; HARDWAREINTERRUPT calls IOFINISH_ASIO.

   Because the I/O Finish interrupt does not designate which I/O has been completed, IOFINISH_ASIO searches through the result queues to find completed I/Os. When one is found, IOFINISH_ASIO selects the appropriate routine to handle the completed I/O; in most cases (in particular, when there are no exceptions reported), it calls FINISHIO.

7. FINISHIO calls FINISHOFFIO.

8. FINISHOFFIO inserts into the logical result descriptor a value indicating the number of units transferred, computes and bills the I/O time, sets the state of the IOCB to "inactive," and causes the I/O completion event, a reference to which was passed as a parameter to the interface procedure.

At this point, the requestor has been notified that the I/O has completed and it can now access result information in the IOCB.

PHYSICALIO Module

See also
I/O Finish Interrupt. 62
IOCB. 33

EXCEPTION HANDLING

The following paragraphs describe how PHYSICALIO handles I/Os that are completed with exceptions. The exception-handling mechanism is discussed in general terms, ignoring most device dependencies. Although handling of device dependencies is really the largest part of the exception-handling task, the basic algorithm is similar for most devices. For the following situations, the A 3, A 9, B 5900, and B 6900 systems' I/O subsystem uses IOFINISH68 and the command queue, and the A 3K and A 10 systems' I/O subsystem uses IOFINISH_ASIO and the unit queue.

Regardless of whether or not the I/O is completed normally, HARDWAREINTERRUPT calls IOFINISH68 or IOFINISH_ASIO, depending on which I/O processing method is used, to handle the I/O Finish interrupt. When IOFINISH68 or IOFINISH_ASIO determines that an exception condition has occurred, in most cases it calls IOEXCEPTION to take the appropriate action.

IOEXCEPTION calls BUILDLOGICALRD, which returns a logical result descriptor generated by analyzing the physical results received from the Data Link Processor (DLP) and the MLIP. Back in IOEXCEPTION, the logical result is then masked with the I/O mask that the requestor passed as a parameter to the I/O initiation procedure; this process may eliminate some exception conditions from consideration. The masked result is then further masked, this time to eliminate exception conditions that do not require handling by PHYSICALIO.

If no exceptions remain after masking the logical result, IOEXCEPTION activates the suspended command queue or unit queue, depending on the I/O processing method used by the MLIP I/O subsystem (see "Command Queue" and "Unit Queue"), and calls FINISHIO to complete the processing of the IOCB (see "Processing of I/Os"). If the only exceptions that remain are simple enough to be handled in IOEXCEPTION (for example, end-of-page on a printer), IOEXCEPTION takes the appropriate action. However, if a more serious exception remains to be handled, IOEXCEPTION calls IOERROR.

IOERROR selects a retry procedure based on the type of unit to which the original I/O was directed. The retry procedure retries the I/O operation until the operation is successful, until an irrecoverable error occurs, or until the retry limit is reached. At this point, IOERROR logs the original error and its retry history. In most cases, IOERROR then calls FINISHIO to return the IOCB to the requestor and activates the command queue or unit queue that was suspended by the MLIP because of the original exception.

PHYSICALIO Module

PHYSICAL I/O OVERVIEW

UNIT INFORMATION

The unit information interface procedures access and alter items in PHYSICALIO's unit information tables, the most important of which are UNITCONTROL, UNITMAP, and UNITSTATUS.

UNITCONTROL Table

The UNITCONTROL table contains information used in making decisions about the management of units. This information includes the unit type, the "ownership" of the unit (in the TAKEUNIT/GIVEUNIT sense), and special status information for the unit. GETUNITINFO references this table to return information as to whether or not the unit is being moved, whether or not the unit has been disabled (through BLASTUNIT), whether or not the unit is the Halt/Load unit. whether or not the initiation of I/O operations to the unit has been suspended because the unit has encountered an end-of-file condition, and so on.

UNITMAP Table

The UNITMAP table maintains the correspondence between logical and physical unit numbers. When GETUNITINFO is requested to return logical or physical unit numbers, it accesses this table.

UNITSTATUS Table

The UNITSTATUS table contains the logical and physical ready/not-ready status for each unit and the paths to those units. When GETUNITINFO is requested to return a value indicating whether or not the specified unit exists or whether or not there is a path to the unit, GETUNITINFO accesses this table.

6 PHYSICALIO/MLIP INTERFACE

The Master Control Program (MCP) communicates with the Message-Level Interface Processor (MLIP) through the PHYSICALIO module. The A 3, A 9, B 5900, and B 6900 systems use the following mechanisms for processing I/O between PHYSICALIO and the MLIP:

- The Communicate with Universal I/O (CUIO) operator
- Shared data structures, which include Input/Output Control Blocks (IOCBs), command queues, horizontal queues, and result queues
- The I/O Finish interrupt

The A 3K and A 10 systems use the following mechanisms for processing I/O between PHYSICALIO and the MLIP:

- The Signal Processing Element Set (SPES) operator/IOCB queue
- Shared data structures, which include IOCBs, IOCB queue, unit queues, MLIP I/O Tables, horizontal queues, and result queues
- The I/O Finish interrupt

The following describes how these mechanisms are used to initiate, process, and complete an I/O at the MLIP level for both groups of systems.

PHYSICAL I/O OVERVIEW

MECHANISMS FOR A 3, A 9, B 5900, AND B 6900 SYSTEMS

An IOCB represents an I/O operation to be performed. When PHYSICALIO has inserted into an IOCB the information required by the MLIP to perform an I/O, the CUIO operator is executed to pass the memory address of the IOCB to the MLIP. The MLIP then queues the IOCB into a command queue, from which the IOCB is later initiated. If the MLIP attempts to initiate an IOCB to a busy Data Link Processor (DLP), the command queue may be temporarily linked into a horizontal queue. When an IOCB is initiated, it is unlinked from the command queue. When the I/O operation associated with an IOCB finishes, the IOCB is linked into a result queue. The MLIP then determines whether or not an I/O Finish interrupt should be generated to inform the processor that an IOCB has finished.

The following data structures are used for PHYSICALIO/MLIP communication in the A 3, A 9, B 5900, and B 6900 systems and are set up by the MCP for the MLIP:

- Command queue. A list of IOCBs that are to be initiated, usually to the same unit.

- Horizontal queue. Mechanism for queuing command queues when the DLP path to the unit is busy.

- Result queue. A list of completed IOCBs. A result queue is built for each processor.

The PHYSICALIO/MLIP interfaces are described individually on the following pages. Because the handling of MLIP I/O subsystem errors may involve all of the interface mechanisms described here, error handling is discussed in "MLIP Error Handling," following the descriptions of the interfaces.

See also

Command Queue 42
Horizontal Queue. 58
Result Queue. 60

MECHANISMS FOR A 3K AND A 10 SYSTEMS

After PHYSICALIO inserts the information required by the MLIP to perform an I/O into an IOCB, PHYSICALIO locks the IOCB queue, queues the IOCB to the tail of the IOCB queue, and unlocks the IOCB queue. If the IOCB queue is empty when the IOCB is queued to it, PHYSICALIO uses the SPES operator to send the MLIP an Initiate I/O signal.

When the MLIP receives the Initiate I/O signal, the signaled MLIP locks the IOCB queue, takes the IOCB at the head of the IOCB queue, locks the unit queue specified by the IOCB, and then unlocks the IOCB queue.

If the IOCB queue contains more IOCBs, the MLIP sends the Initiate I/O signal to the other MLIPs specified in the MLIP Destination Set (Word 6) of the MLIP I/O table (see "MLIP I/O Table"). The MLIP then queues the IOCB to the unit queue. If the IOCB is queued at the head of the unit queue, the MLIP initiates the unit queue. If the MLIP attempts to initiate an IOCB to a busy DLP, the unit queue is temporarily linked into a horizontal queue.

When the I/O is finished, the MLIP puts the IOCB into the appropriate result queue. After queuing an IOCB into an empty result queue, the MLIP uses the EMP Destination Set (Word 5) of the MLIP I/O table to signal one of the E-Mode processors (EMPs) in the set to handle an I/O Finish (see "MLIP I/O Table").

Because processing elements (data and I/O) share data structures, access to the data structures is synchronized through a lock bit in the lock word of each data structure.

The IOCB queue is locked by MLIPs and the MCP. All other data structures are locked by the MLIP only. After successfully locking and using the data structure, the MLIP or MCP writes the lock word back to memory, which unlocks the data structure.

The following data structures are used for PHYSICALIO/MLIP communication in the A 3K and A 10 systems and are set up by the MCP for the MLIP:

- IOCB queue. Used by PHYSICALIO to send IOCBs to the MLIPs.
- Unit queue. Allocated such that all I/Os for one unit go through the assigned unit queue, regardless of the number of paths to the unit.

PHYSICAL I/O OVERVIEW

- MLIP I/O table. Contains the MLIP queue and other information used by the MLIP. The MLIP uses the MLIP queue to pass unit queues to other MLIPs. There is one MLIP queue in each MLIP I/O table and one MLIP I/O table for each MLIP in the partition. There are several words in the table. The EMP and MLIP destination set words in the table indicate which processing elements are available for I/O processing. The other words in the table are described when necessary for the overview.

- Horizontal queue. Mechanism for queuing unit queues when the DLP path to the unit is busy.

- Result queue. A list of completed IOCBs. Only one result queue is initialized.

The MLIP uses disk seek optimization to increase I/O throughput. Disk seek optimization is performed during the initiation of the IOCB from the unit queue. It selects the IOCB that specifies a disk address closest to the current position of the disk head. Disk seek optimization depends on the single point of control provided by unit queues.

The PHYSICALIO/MLIP interfaces are described individually on the following pages. Because the handling of MLIP I/O subsystem errors may involve all of the interface mechanisms described here, error handling is discussed in "MLIP Error Handling," following the descriptions of the interfaces.

See also

Horizontal Queue. 58
IOCB Queue. 41
MLIP I/O Table. 56
Result Queue. 60
Unit queue. 48

PHYSICALIO/MLIP Interface

IOCB

An IOCB is an MCP-allocated memory area that either contains or references all of the information required by the MLIP to perform an I/O operation. It also contains areas that are used by the MLIP to temporarily store information during the processing of an I/O and to return information to PHYSICALIO when the I/O has finished. This information is formatted into the 15 IOCB words shown in Table 1. The table also shows which IOCB words are assigned meaningful (nonzero) values by the MCP prior to initiation of the IOCB and which are assigned values by the MLIP during processing of the IOCB.

Table 1. IOCB I/O Table

		Values Assigned by	
		MCP	MLIP
Word 0:	MLIP Control Word	*	
Word 1:	DLP Address Word	*	*
Word 2:	Command or Unit Queue Header Pointer	*	
Word 3:	IOCB Self Pointer	*	
Word 4:	DLP I/O Command Pointer	*	
Word 5:	DLP I/O Result Pointer	*	
Word 6:	DLP Command/Result Lengths	*	
Word 7:	Result Mask	*	
Word 8:	Result Queue Head Pointer	*	
Word 9:	Next IOCB Link		*
Word 10:	MLIP Current Data Area Pointer	*	*
Word 11:	MLIP Current I/O Length	*	*
Word 12:	MLIP State and Result		*
Word 13:	I/O Start Time		*
Word 14:	I/O Finish Time		*

PHYSICAL I/O OVERVIEW

The actual IOCB, as allocated by the MCP, is longer than 15 words. The additional space is used by the MCP to store information about the MCP's processing of I/O; for example, a reference to the event to be caused when the I/O finishes is kept in the MCP portion of the IOCB.

The following describes how each of the 15 MLIP-visible IOCB words, and their fields, is used by the MCP and the MLIP. References to the command queue apply to the A 3, A 9, B 5900, and B 6900 systems; references to the unit queue apply to A 3K and A 10 systems.

MLIP Control Word

Word 0 contains several fields set by the MCP to specify the I/O operation to be performed. The fields are described below, and an example of the MCP's use of each special-purpose field is provided.

IOCB Mark

This field contains the code 4'1OCB', which marks the word as the first word of an IOCB.

Queue at Head

When this one-bit field is TRUE, the MLIP will queue the IOCB at the head of the command queue or unit queue. One use of this feature is to perform retry I/Os when a device-oriented error has occurred.

MLIP/DLP Command

When this one-bit field is TRUE, the IOCB contains a command to be interpreted by the MLIP itself; the MLIP command is contained in the first word pointed to by the DLP I/O Command Pointer (Word 4) in the IOCB. The MLIP operations that the MCP can request are described in the "MLIP" section.

When this one-bit field is FALSE, the IOCB contains a command for the DLP referenced by the DLP Address Word (Word 1) of the IOCB.

Attention

When this one-bit field is TRUE, the MLIP will set both the Attention and the Exception fields in the MLIP result (see "MLIP State and Result" below). This field forces the command queue or unit queue to be suspended upon completion of the I/O operation, or ensures that the I/O will be processed by the exception handling routines. For example, all binary card reads are initiated with the Attention field set to TRUE, because IOEXCEPTION is responsible for recognizing the binary end-of-file card (see "Exception Handling").

PHYSICALIO/MLIP Interface

Cause I/O Finish Interrupt

When this one-bit field is TRUE, the MLIP will unconditionally cause an I/O Finish interrupt when the I/O completes. When this one-bit field is FALSE, the MLIP will make the decision as to whether or not to cause an interrupt based on other conditions (see "I/O Finish Interrupt").

Memory Override

When this one-bit field is TRUE, the MLIP will ignore odd-tagged (memory-protected) words in memory during data transfer. This field is set to TRUE for I/Os initiated to perform memory I/Os for requestors such as PRESENCEBIT and SWAPPER.

Input

When this one-bit field is TRUE, input from the DLP is allowed.

Output

When this one-bit field is TRUE, output to the DLP is allowed.

Output Zeros

When this one-bit field is TRUE, the MLIP will generate a data stream of all 0's (zeros) to send to the DLP. This feature is used only to perform erase operations on magnetic tape DLPs that require data to be sent, to determine the length of the erasure.

Tag Control

This three-bit field controls the setting and transfer of tags during I/O operations.

Word-Oriented Transfer

When this one-bit field is TRUE, the MLIP interprets the MLIP Current I/O Length (Word 11) in units of words, as opposed to characters.

Memory Direction

When this one-bit field is TRUE and the Input field of the MLIP Control Word is TRUE, then the MLIP transfers the received data to memory locations of descending order.

Continue Count at End of Length

When this one-bit field is TRUE, the MLIP allows the DLP to transfer more data than the originally specified I/O length (allowing the MLIP Current I/O Length (Word 11) to be decremented past 0), but does not store the excess data in memory. This field is set to TRUE when the software requires information about the actual length of a block read from a variable-record-length device such as tape.

Ignore Count Error
: When this one-bit field is TRUE, the MLIP will not report a Count Error, even if the MLIP Current I/O Length (Word 11) is not 0 when the I/O has finished (see "MLIP State and Result" below). This field is set to TRUE for some I/Os to fixed-record-length devices and for some tape read operations (especially when the Continue Count at End of Length is TRUE). Ignore Count Error is specifically set to FALSE for I/Os for which the length of the data transfer must be exact (for example, for disk, Network Support Processor (NSP), and tape write operations).

Don't Count
: When this one-bit field is TRUE, the MLIP will not increment or decrement the command or unit queue's Active Count for this I/O. This feature is used only when initiating a Cancel operation (used only for DLPs that do not support a Discontinue operation), because the Cancel does not finish normally.

Ignore Suspend All Queues
: The MLIP has a bit (MLIP Suspend All Queues field) set to unconditionally suspend the command or unit queue when the I/O completes. However, if the Ignore Suspend All Queues field is TRUE, the MLIP will not suspend the command or unit queue when the I/O completes. This feature is used only when initiating Error IOCBs (see "MLIP Error Handling").

Immediate
: When this one-bit field is TRUE, the MLIP will ignore the value of the command or unit queue's Active Count and Suspended fields when considering initiating the IOCB. This field is usually set for MLIP operations (see the "MLIP" section), for peripheral status operations (see the "Peripheral Status" section), and for retry I/Os.

Disk Seek Optimization
: When this one-bit field is TRUE, the MLIP is allowed to reorder the IOCB with respect to other IOCBs in the unit queue. Disk seek optimization is available only on A 3K and A 10 systems.

PHYSICALIO/MLIP Interface

Simple Path Selection

When this one-bit field is TRUE, the MLIP uses the path specified in the unit queue. The Path Table Pointer (Word 9) of the unit queue specifies the path (see "Unit Queue"). Simple path selection is available only on A 3K and A 10 systems.

DLP Address Word

Word 1 contains several fields used by the MLIP to address a DLP. For A 3, A 9, B 5900, and B 6900 systems, this word contains an MLI port number, a Line Expansion Module (LEM) port number, and a DLP address within the base (see "Path Specification"). The A 3K and A 10 systems also use the above fields and have three additional fields: an MLIP processor ID, a Host Return field, and a DLP index.

Command or Unit Queue Header Pointer

Word 2 contains a reference to the command queue header for the command queue, or the unit queue header for the unit queue, in which the IOCB is to be queued.

IOCB Self Pointer

Word 3 contains a reference to the IOCB itself.

DLP I/O Command Pointer

Word 4 contains a reference to the memory area containing the command to be sent to the DLP.

DLP I/O Result Pointer

Word 5 contains a reference to the memory area in which the DLP result is to be stored.

DLP Command/Result Lengths

Word 6 contains the length of the DLP command to be sent and the length of the expected DLP result. For A 3K and A 10 systems it also contains a Disk Address field. If the IOCB indicates that disk seek optimization is to be performed, this field contains the 16 most significant bits of the disk address at which the data transfer begins.

Result Mask

Word 7 contains a mask indicating which DLP results are not to be considered exceptions for the purpose of reporting DLP errors in the MLIP result (see "MLIP State and Result" below).

Result Queue Head Pointer

Word 8 contains a reference to the result queue head for the result queue into which this IOCB is to be queued upon completion.

Next IOCB Link

Word 9 contains a link to the next IOCB in the command queue or result queue for A 3, A 9, B 5900, and B 6900 systems. The MLIP updates this link as the I/O is being processed. For A 3K and A 10 systems this word contains a link to the next IOCB in the IOCB queue, unit queue, or result queue. The MCP updates the link when queuing IOCBs in the IOCB queue.

MLIP Current Data Area Pointer

Word 10 contains a reference to the data area for the data to be transferred. It is initialized using the buffer descriptor, offset, and length passed to PHYSICALIO by the I/O requestor, and is updated by the MLIP as data is transferred.

MLIP Current I/O Length

Word 11 contains the length of the data area remaining for data transfer. It is initialized to the I/O length passed to PHYSICALIO by the I/O requestor and is updated by the MLIP as data is transferred.

MLIP State and Result

Word 12 is assigned by the MLIP to report the result information for the I/O performed. The following types of exceptions are reported:

Exception

This one-bit field is set when any other exception fields are set.

Attention

This one-bit field is set in the MLIP Control Word (Word 0) in the IOCB.

DLP Error

This one-bit field is set when the result of masking the first 48 bits of the DLP result with the Result Mask (Word 7) in the IOCB is not 0.

MLIP/MLI Error

This one-bit field is set if any of the following one-bit exception fields for the MLIP State and Result word are set:

Memory Protect
Count Error
Improper IOCB Word (for example, incorrect tag)
Invalid MLIP Control Field
MLI Vertical Parity Error
MLI Longitudinal Parity Word Error
Unexpected DLP Status (MLI protocol error)
Nonpresent DLP

DLP Busy
MLI Time Out
Invalid MLIP Command

MLIP/Hardware Error
This one-bit field is set when an error occurs that must be reported through an Error IOCB (see "MLIP Error Handling").

Completed After Queue Suspended
This one-bit field is set if the I/O finished when the Suspended field in the Queue Control Word of the command queue header is TRUE (see "Command Queue").

MLIP Not Available
This one-bit field is available only on A 3K and A 10 systems and is set if the DLP Address Word (Word 1) of the IOCB references an MLIP processor ID that is not available.

I/O Start Time
The MLIP stores the time of day into Word 13 when the MLIP initiates the I/O.

I/O Finish Time
The MLIP stores the time of day into Word 14 when the I/O is finished.

Before passing the IOCB to the MLIP, PHYSICALIO ensures that the IOCB contains all of the information required for the I/O operation. Some information is not required for some types of I/O operations. For example, information pertaining to DLPs is not required for operations directed to the MLIP itself, and information pertaining to data buffers and lengths is not required for operations that do not involve data transfer. Some words in the IOCB are changed as the IOCB is queued, initiated, processed, and finished. These changes are described as the discussion proceeds through the I/O process.

See also

CUIO OPERATOR

The CUIO operator is used in A 3, A 9, B 5900, and B 6900 systems and is executed by the MCP to pass the address of an IOCB to the MLIP for initiation. The processor executing the CUIO operator verifies that its parameter is the address of an IOCB by checking the IOCB Mark field in the MLIP Control Word (Word 0). If the mark is not correct, an Invalid Operand interrupt is generated; if it is correct, the address is passed to the MLIP, and the operator completes when the MLIP indicates that it has received the address.

The MLIP also verifies the IOCB Mark field. If the mark is not correct, an Error IOCB is completed (see "MLIP Error Handling"). If the mark is correct, the MLIP queues the referenced IOCB into the command queue specified by the Command Queue Header Pointer (Word 2) and determines whether or not to initiate the first I/O in that command queue. The I/O initiation process involves some changes to the IOCB (described in "IOCB Queue") and may require communication with the UIO subsystem (described in the "MLIP/UIO Interface" section).

See also
MLIP Error Handling 63
MLIP/UIO Interface. 71

PHYSICALIO/MLIP Interface

IOCB QUEUE

The IOCB queue is used by PHYSICALIO in the A 3K and A 10 systems to send IOCBs to the MLIPs. To assure that I/Os are initiated in the correct order for single-user devices such as tape drives and printers, only one IOCB queue should be created per partition. Each MLIP's I/O table contains a pointer to the IOCB queue in the IOCB Queue Head Pointer (Word 1).

The IOCB queue is composed of the three words shown in Table 2 and described below.

Table 2. IOCB Queue

Word 0:	IOCB Queue Control Word
Word 1:	IOCB Queue Head
Word 2:	IOCB Queue Tail

IOCB Queue Control Word
Word 0 contains the IOCB Queue Mark 4'10C1' and a lock bit. The lock bit is used by PHYSICALIO and the MLIP to synchronize access to the IOCB queue.

IOCB Queue Head
Word 1 contains a reference to the first IOCB in the IOCB queue.

IOCB Queue Tail
Word 2 contains a reference to the last IOCB in the IOCB queue.

COMMAND QUEUE

A command queue is used in A 3, A 9, B 5900, and B 6900 systems and is a linked list of IOCBs that are to be initiated, usually to the same unit. A command queue is controlled by a data structure called a command queue header, which is composed of the five words shown in Table 3 and described below.

Table 3. Command Queue

Word 0:	Queue Control Word
Word 1:	Head IOCB Link
Word 2:	Tail IOCB Link
Word 3:	Horizontal Queue Head Pointer
Word 4:	Horizontal Queue Link

Queue Control Word

Word 0 contains several fields that provide parameter information established by the MCP and queue status information maintained by the MLIP.

Command Queue Header Mark

This 16-bit field contains the code 4'10CC', which marks the word as the first word of a command queue header.

Inactive Count

This eight-bit field contains the number of IOCBs that are currently queued but have not been initiated.

Active Count

This eight-bit field contains the number of IOCBs that have been initiated out of the queue and are still in progress.

Active Limit

This eight-bit field is initialized by the MCP to the maximum number of IOCBs from this queue that may be simultaneously in progress. The MLIP will not initiate any IOCBs from the queue if the Active Count is greater than or equal to the Active Limit, unless the IOCB being considered for initiation has the Immediate field in the MLIP Control Word (Word 0) set to TRUE (see "IOCB").

Suspended

This one-bit field is set to TRUE under the following conditions:

1. When an IOCB initiated out of the queue completes with the Exception field in the MLIP State and Result (Word 12) set to TRUE

2. When an IOCB initiated out of the queue is completed while the MLIP Suspend All Queues field is TRUE and the Ignore Suspend All Queues field in the IOCB's MLIP Control Word (Word 0) is FALSE

3. When the MCP requests the MLIP to deactivate the queue (see "MLIP")

If the command queue's Suspended field is TRUE, the MLIP will initiate an IOCB out of the queue only if the IOCB being considered for initiation has the Immediate field in the MLIP Control Word (Word 0) set to TRUE.

Waiting

This one-bit field is set to TRUE by the MLIP when the command queue is linked into a horizontal queue (see "Horizontal Queue").

Horizontal Queue Present

This one-bit field is set to TRUE by the MCP to indicate that the MLIP can link this queue into the horizontal queue specified in the Horizontal Queue Head Pointer field (see below) when necessary.

Head IOCB Link

Word 1 contains a reference to the first (inactive) IOCB in the command queue. It is updated whenever an IOCB is unlinked from the queue and whenever a new IOCB is inserted at the head of the queue because the Queue at Head field in the MLIP Control Word (Word 0) of the IOCB was TRUE.

Tail IOCB Link

Word 2 contains a reference to the last (inactive) IOCB in the command queue. It is updated whenever a new IOCB is linked at the end of the queue and whenever the last IOCB is initiated out of the queue.

Horizontal Queue Head Pointer

Word 3 is initialized by the MCP and contains a reference to the head of the horizontal queue into which this command queue is to be linked if necessary (see "Horizontal Queue").

Horizontal Queue Link
    Word 4 is used by the MLIP to link the command queue into a horizontal queue when necessary (see "Horizontal Queue").

For most devices, PHYSICALIO allocates one command queue per unit per path to that unit. For Operator Display Terminals (ODTs), there are two command queues per unit: one for writes and one for reads and test/wait for transmit operations. For Network Support Processors (NSPs), three command queues for each NSP are allocated (see the "Data Communications" section).

When an I/O is linked into a command queue, the MLIP updates the Inactive Count field of the Queue Control Word (Word 0) to indicate that another IOCB has been queued. If the IOCB was queued at the head of the command queue, the Head IOCB Link (Word 1) in the command queue header and the Next IOCB Link (Word 9) of the IOCB following the new IOCB are updated; if the IOCB was queued at the tail of the command queue, the Tail IOCB Link (Word 2) of the command queue header and the Next IOCB Link (Word 9) of the IOCB preceding the new IOCB are updated. Figure 5 shows three IOCBs linked into command queue header "A".

PHYSICALIO/MLIP Interface

```
                    Command Queue Header "A"
                   |-----------------------|
          |------->| Queue Control Word    |
          |        |-----------------------|
|---------)--------- Head IOCB Link        |
|         |        |-----------------------|
|         |        | Tail IOCB Link        ---------------------|
|         |        |-----------------------|                    |
|         |        | HQ Head Pointer       |                    |
|         |        |-----------------------|                    |
|         |        | HQ Link               |                    |
|         |        |-----------------------|                    |
|         |                                                     |
|         |<-------+<-------------------+<-------------------+  |
|                  |                    |                    |  |
|       IOCB 1     |         IOCB 2     |         IOCB 3     |  |
|   |-----------|  |     |-----------|  |     |-----------|  |  |
|-->|           |  | |-->|           |  | |-->|           |<-)--|
    |-----------|  | |   |-----------|  | |   |-----------|  |
    | CQ Header ---| |   | CQ Header ---| |   | CQ Header ---|
    |-----------|    |   |-----------|    |   |-----------|
    |           |    |   |           |    |   |           |
    |-----------|    |   |-----------|    |   |-----------|
    | Next Link -----|   | Next Link -----|   | Next=NULL |
    |-----------|        |-----------|        |-----------|
    |           |        |           |        |           |
    |-----------|        |-----------|        |-----------|
```

CQ = Command queue
HQ = Horizontal queue

Figure 5. Multiple IOCBs Linked to a Command Queue

The command queue header links are static throughout the processing of the IOCB and are not shown in subsequent figures.

The MLIP begins processing a particular command queue under any of the following circumstances:

1. When PHYSICALIO initiates an IOCB to that queue.

2. When an I/O operation for an IOCB in that queue is complete (for example, when an activate queue MLIP operation, simply by ending, causes that command queue to be processed by the MLIP).

3. When the command queue is reached by following the links in a horizontal queue (see "Horizontal Queue").

PHYSICAL I/O OVERVIEW

Whenever the MLIP is processing a particular command queue, it attempts to initiate as many IOCBs as possible from that queue. The number of IOCBs that can be initiated depends on factors such as whether or not

- The command queue's Active Count is less than the command queue's Active Limit.
- The Suspended field is TRUE.
- The IOCB's Immediate field is TRUE.
- The command queue is already queued in a horizontal queue waiting for the DLP (see "Horizontal Queue").

When an IOCB is initiated, the IOCB's Next IOCB Link (Word 9) is set to 1 and the IOCB is unlinked from the command queue, as shown in Figure 6.

```
                     Command Queue Header "A"
                     |------------------------|
                     | Queue Control Word     |
                     |------------------------|
          |---------- Head IOCB Link          |
          |          |------------------------|
          |          | Tail IOCB Link          -------------------|
          |          |------------------------|                   |
          |          | HQ Head Pointer        |                   |
          |          |------------------------|                   |
          |          | HQ Link                |                   |
          |          |------------------------|                   |
          |                                                       |
          |----------|                                            |
                     |                                            |
   IOCB 1            |      IOCB 2                IOCB 3          |
|------------|       |  |------------|       |------------|       |
|            |       |-->|            |    |-->|            |<--|
|------------|          |------------|    |  |------------|
| CQ Header  |          | CQ Header  |    |  | CQ Header  |
|------------|          |------------|    |  |------------|
|            |          |            |    |  |            |
|------------|          |------------|    |  |------------|
|  Next=1    |          | Next Link -----|  | Next=NULL  |
|------------|          |------------|       |------------|
|            |          |            |       |            |
|------------|          |------------|       |------------|
```

CQ = Command queue
HQ = Horizontal queue

Figure 6. Initiated IOCB Unlinked from a Command Queue

PHYSICALIO/MLIP Interface

At the time the I/O is initiated, the MLIP sets the I/O Start Time (Word 13) of the IOCB to the time of day.

For I/O operations that involve data transfer, the data is transferred in bursts when the DLP indicates that it is ready to receive or transmit data (see the "MLIP/UIO Interface" section). As data is sent to or received from the DLP, the MLIP adjusts the MLIP Current Data Area Pointer (Word 10) and the MLIP Current I/O Length (Word 11) of the IOCB to correspond to the location and amount of data left to be transmitted.

See also

UNIT QUEUE

A unit queue is used in A 3K and A 10 systems and is a linked list of IOCBs that are to be initiated, usually to the same unit. A unit queue is controlled by a data structure called a unit queue header, which is composed of the 11 words shown in Table 4 and described below.

Table 4. Unit Queue

Word 0:	Queue Control Word
Word 1:	Head IOCB Link
Word 2:	Tail IOCB Link
Word 3:	Horizontal Queue Head Pointer
Word 4:	Dynamic Unit Queue Link
Word 5:	Last IOCB Initiated
Word 6:	Spare
Word 7:	I/O Optimization Word 1
Word 8:	I/O Optimization Word 2
Word 9:	Path Table Pointer
Word 10:	Unit-Related Path Information

Queue Control Word

Word 0 contains fields that provide parameter information established by the MCP and queue status information maintained by the MLIP.

Unit Queue Header Mark

This 16-bit field contains the code 4'10CC', which marks the word as the first word of a unit queue header.

Inactive Count

This eight-bit field contains the number of IOCBs that are currently queued but have not been initiated.

Active Count

This eight-bit field contains the number of IOCBs that have been initiated out of the queue and are still in progress.

Active Limit
This eight-bit field is initialized by the MCP to the maximum number of IOCBs from this queue that can be simultaneously in progress. The MLIP will not initiate any IOCBs from the queue if the Active Count is greater than or equal to the Active Limit, unless the IOCB being considered for initiation has the Immediate field in the MLIP Control Word (Word 0) set to TRUE (see "IOCB").

Suspended
This one-bit field is set to TRUE under the following conditions:

1. When an IOCB initiated out of the queue is completed with the Exception field in the MLIP State and Result (Word 12) set to TRUE

2. When an IOCB initiated out of the queue is completed while the MLIP Suspend All Queues field is TRUE and the Ignore Suspend All Queues field in the IOCB's MLIP Control Word (Word 0) is FALSE

3. When the MCP requests the MLIP to deactivate the queue (see the "MLIP" section)

If the unit queue's Suspended field is TRUE, the MLIP will initiate an IOCB out of the queue only if the IOCB being considered for initiation has the Immediate field in the MLIP Control Word (Word 0) set to TRUE.

Waiting
This one-bit field is set to TRUE by the MLIP when the unit queue is linked into a horizontal queue (see "Horizontal Queue").

Horizontal Queue Present
This one-bit field is set to TRUE by the MCP to indicate that the MLIP can link this queue into the horizontal queue specified by the Horizontal Queue Head Pointer field (see below) when necessary.

Path Selected
This one-bit field tells whether or not the MLIP has executed the Path Selection algorithm for the top IOCB in the unit queue.

Connection in Progress

If this one-bit field is set, an MLIP has initiated a Poll Test to the DLP specified by the Data Link Processor Address Word (DLPAW) of the top IOCB in the unit queue.

Waiting in MLIP Queue

If this one-bit field is set, the unit queue is currently dynamically linked in an MLIP queue.

Lock Bit

This one-bit field is used by the MLIPs to synchronize access to the unit queue.

Head IOCB Link

Word 1 contains a reference to the first (inactive) IOCB in the unit queue. It is updated whenever an IOCB is unlinked from the queue and whenever a new IOCB is inserted at the head of the queue because the Queue at Head field in the MLIP Control Word (Word 0) of the IOCB was TRUE.

Tail IOCB Link

Word 2 contains a reference to the last (inactive) IOCB in the unit queue. It is updated whenever a new IOCB is linked at the end of the queue and whenever the last IOCB is initiated out of the queue.

Horizontal Queue Head Pointer

Word 3 is initialized by the MCP and contains a reference to the head of the horizontal queue into which this unit queue is to be linked if necessary (see "Horizontal Queues").

Dynamic Unit Queue Link

Word 4 is used by the MLIP to dynamically link unit queues in horizontal queues and MLIP queues.

Last IOCB Initiated

Word 5 references the last non-MLIP command IOCB initiated from the unit queue.

Spare

Word 6 is reserved for future expansion.

I/O Optimization Word 1

Word 7 contains information that is used to optimize the throughput of disk I/Os.

Physical Integrity Check

This one-bit field determines whether the MLIP guarantees that overlapping I/Os are not reordered with respect to one another.

PHYSICALIO/MLIP Interface

Maximum IOCBs to Examine

This six-bit field determines the maximum number of IOCBs that are examined by the Disk Seek Optimization algorithm.

Bypass Limit

This four-bit field contains the maximum number of times that the top IOCB can be bypassed in favor of another IOCB in the unit queue at IOCB initiation time.

Bypass Count

This four-bit field contains the number of times that the top IOCB has been passed over in favor of another IOCB in the unit queue at IOCB initiation time.

Disk Address

This 16-bit field contains the value from the Disk Address field in the DLP Command/Result Lengths (Word 6) in the IOCB selected as the best IOCB by the Disk Seek Optimization algorithm.

I/O Optimization Word 2

Word 8 contains information that is used to optimize the throughput of disk I/Os. If the Physical Integrity Check field of I/O Optimization Word 1 is set, PHYSICALIO must initialize the Cylinder Size and Minimum Difference fields.

Cylinder Size

This 28-bit field contains the size of a disk cylinder in bytes.

Minimum Difference

This 20-bit field contains the information to determine if I/Os overlap.

Path Table Pointer

Word 9 contains the address of the DLP for simple path selection.

Unit-Related Path Information

Word 10 contains unit-related path information and the following field.

Unit DLP Mask

This field specifies the DLPs through which there is a ready physical path to the unit.

PHYSICALIO allocates one unit queue per unit, regardless of the number of paths to the unit. The name "unit queue" implies that all I/Os for one unit are sent to a single queue.

PHYSICAL I/O OVERVIEW

When an I/O is linked into a unit queue, the MLIP updates the Inactive Count field of the Queue Control Word (Word 0) to indicate that another IOCB has been queued. If the IOCB was queued at the head of the unit queue, the Head IOCB Link (Word 1) in the unit queue header and the Next IOCB Link (Word 9) of the IOCB following the new IOCB are updated; if the IOCB was queued at the tail of the unit queue, the Tail IOCB Link (Word 2) of the unit queue header and the Next IOCB Link (Word 9) of the IOCB preceding the new IOCB are updated. Figure 7 shows three IOCBs linked into unit queue header "A":

```
                        Unit Queue Header "A"
                       |-----------------------|
              |------->| Queue Control Word    |
              |        |-----------------------|
    |---------)--------- Head IOCB Link        |
    |         |        |-----------------------|
    |         |        | Tail IOCB Link        ----------------------|
    |         |        |-----------------------|                     |
    |         |        | HQ Head Pointer       |                     |
    |         |        |-----------------------|                     |
    |         |        | Dynamic UQ Link       |                     |
    |         |        |-----------------------|                     |
    |         |                    .                                 |
    |         |                    .                                 |
    |         |                    .                                 |
    |         |        |-----------------------|                     |
    |         |        | Unit-Related Path Info|                     |
    |         |        |-----------------------|                     |
    |         |                                                      |
    |         |<-------+<-------------------+<-------------------+   |
    |                  |                    |                    |   |
    |       IOCB 1     |         IOCB 2     |         IOCB 3     |   |
    |   |-----------|  |     |-----------|  |     |-----------|  |   |
    |-->|           |  | |-->|           |  | |-->|           |<-)---|
        |-----------|  | |   |-----------|  | |   |-----------|  |
        | UQ Header ---| | | UQ Header ---| | | UQ Header ---|
        |-----------|    |   |-----------|    |   |-----------|
        |           |    |   |           |    |   |           |
        |-----------|    |   |-----------|    |   |-----------|
        | Next Link -----|   | Next Link -----|   | Next=NULL  |
        |-----------|        |-----------|        |-----------|
        |           |        |           |        |           |
        |-----------|        |-----------|        |-----------|
```

HQ = Horizontal queue
UQ = Unit queue

Figure 7. Multiple IOCBs linked to a Unit Queue

PHYSICALIO/MLIP Interface

The unit queue header links are static throughout the processing of the IOCB and are not shown in subsequent figures.

The MLIP begins processing a particular unit queue under any of the following circumstances:

1. When PHYSICALIO initiates an IOCB to that queue
2. When an I/O operation for an IOCB in that queue is complete (for example, when an Activate Queue MLIP operation, simply by ending, causes that unit queue to be processed by the MLIP)
3. When the unit queue is reached by following the links in a horizontal queue (see "Horizontal Queue")

Whenever the MLIP is processing a particular unit queue, it attempts to initiate as many IOCBs as possible from that queue. The number that can be initiated depends on factors such as whether or not the unit queue's Active Count is less than the unit queue's Active Limit, whether or not the Suspended field is TRUE, whether or not the IOCB's Immediate field is TRUE, and whether or not the unit queue is already queued in a horizontal queue waiting for the DLP (see "Horizontal Queue").

When an IOCB is initiated, the IOCB's Next IOCB Link (Word 9) is set to 1 and the IOCB is unlinked from the unit queue, as shown in Figure 8.

```
                        Unit Queue Header "A"
                       |-----------------------|
                       | Queue Control Word    |
                       |-----------------------|
              |--------- Head IOCB Link        |
              |        |-----------------------|
              |        | Tail IOCB Link        -------------------|
              |        |-----------------------|                  |
              |        | HQ Head Pointer       |                  |
              |        |-----------------------|                  |
              |        | Dynamic UQ Link       |                  |
              |        |-----------------------|                  |
              |                    .                              |
              |                    .                              |
              |                    .                              |
              |        |-----------------------|                  |
              |        | Unit-Related Path Info|                  |
              |        |-----------------------|                  |
              |                                                   |
              |----------|                                        |
                         |                                        |
            IOCB 1       |       IOCB 2               IOCB 3      |
        |-----------|    |   |-----------|        |-----------|   |
        |           |    |-->|           |    |-->|           |<--|
        |-----------|        |-----------|    |   |-----------|
        | UQ Header |        | UQ Header |    |   | UQ Header |
        |-----------|        |-----------|    |   |-----------|
        |           |        |           |    |   |           |
        |-----------|        |-----------|    |   |-----------|
        |  Next=1   |        | Next Link -----|   | Next=NULL |
        |-----------|        |-----------|        |-----------|
        |           |        |           |        |           |
        |-----------|        |-----------|        |-----------|
```

HQ = Horizontal queue
UQ = Unit queue

Figure 8. Initiated IOCB Unlinked from a Unit Queue

At the time the I/O is initiated, the MLIP sets the I/O Start Time (Word 13) of the IOCB to the time of day.

PHYSICALIO/MLIP Interface

For I/O operations that involve data transfer, the data is transferred in bursts when the DLP indicates that it is ready to receive or transmit data (see the "MLIP/UIO Interface"section). As data is sent to or received from the DLP, the MLIP adjusts the MLIP Current Data Area Pointer and the MLIP Current I/O Length (Words 10 and 11) of the IOCB to correspond to the location and amount of data left to be transmitted.

See also

MLIP I/O TABLE

The MLIP I/O table is used by the MLIP in the A 3K and A 10 systems. The MCP sets up an MLIP I/O Table for each MLIP in the partition. Once the table has been set up, the MCP is not allowed to update or access any word in the table except through an MLIP I/O operation.

The MLIP I/O table, shown in Table 5 below, contains an IOCB queue head pointer, the MLIP queue, destination sets, and a scratch area. Descriptions of these words and fields are given below.

Table 5. MLIP I/O Table

Word 0:	MLIP I/O Table Control Word
Word 1:	IOCB Queue Head Pointer
Word 2:	MLIP Queue Control Word
Word 3:	MLIP Queue Head Unit Queue Link Data Descriptor
Word 4:	MLIP Queue Tail Unit Queue Link Data Descriptor
Word 5:	EMP Destination Set
Word 6:	MLIP Destination Set
Word 7:	MLIP Scratch Area Word 0
	.
	.
	.
Word 16:	MLIP Scratch Area Word 9

MLIP I/O Table Control Word
Word 0 consists of the MLIP I/O table control mark field containing the code 4'10C0'.

IOCB Queue Head Pointer
Word 1 contains a pointer to the IOCB queue. The MCP initializes the IOCB queue head pointer, which cannot be altered by the MLIP or the MCP.

PHYSICALIO/MLIP Interface

MLIP Queue Control Word

Word 2 consists of a MLIP queue mark field, an undefined area, and a lock bit.

MLIP Queue Mark

The MLIP Queue Mark field contains the code 4'10C2'.

Lock Bit

MLIPs use the lock bit to synchronize access to the MLIP queue.

MLIP Queue Head Unit Queue Link Data Descriptor

Word 3 contains the pointer to the first unit queue in the MLIP queue. PHYSICALIO initializes this word to 0.

MLIP Queue Tail Unit Queue Link Data Descriptor

Word 4 contains the pointer to the last unit queue in the MLIP queue. PHYSICALIO initializes this word to 0.

EMP Destination Set

Word 5 contains the destination set of EMPs that are signaled for I/O Finish. The EMP destination set is a bit mask where bit 1 corresponds to EMP #1, bit 2 corresponds to EMP #2, and so on. The MCP sets a bit for each running EMP in the partition. If the partition configuration changes, the MCP updates the word by using an MLIP command.

MLIP Destination Set

Word 6 contains the destination set of MLIPs. An MLIP can communicate only to MLIPs specified in this word. The MLIP destination set is a bit mask where bit 1 corresponds to MLIP #1, bit 2 corresponds to MLIP #2, and so on. The MCP sets a bit for each running MLIP in the partition. If the partition configuration changes, the MCP updates the word by using the MLIP command.

MLIP Scratch Area Word 0 through Word 9

The MLIP uses this as a scratch area that consists of 10 words (Word 0 through Word 9). The MCP initializes the scratch area to 0.

HORIZONTAL QUEUE

A horizontal queue is a mechanism for queuing command queues or unit queues when the DLP path to the unit is busy. The A 3, A 9, B 5900, and B 6900 systems use the command queue, and the A 3K and A 10 systems use the unit queue, to initiate the MLIP. The term "command/unit queue" is used to refer to the queue for its respective computer system.

During system initialization, PHYSICALIO builds a horizontal queue array containing a one-word horizontal queue header for each of several horizontal queues (Word 0 of the array contains a Queue Length field and the code 4'10CE' as a Horizontal Queue Mark). In general, PHYSICALIO assigns a horizontal queue header for each DLP that can accept multiple I/O requests but might be momentarily unable to accept a command because it is actively processing a previous command.

When the MLIP attempts to initiate an I/O to a DLP that is busy, the MLIP checks the Horizontal Queue Present field in the Queue Control Word (Word 0) in the command/unit queue header to determine whether or not this command/unit queue is eligible to be queued into a horizontal queue. If so, the MLIP queues the command/unit queue in the horizontal queue specified by the Horizontal Queue Head Pointer (Word 3) in the command/unit queue header. If this command/unit queue is the first in the horizontal queue, the Horizontal Queue Head is set to point to this command/unit queue. If it is not the first (that is, if there is already at least one command/unit queue in the same horizontal queue), this command/unit queue is linked to the tail of the horizontal queue. In A 3, A 9, B 5900, and B 6900 systems, the Horizontal Queue Link (Word 4) in the command queue header of the command queue (previously at the tail of the horizontal queue) is set to point to this command queue. In A 3K and A 10 systems, the Dynamic Unit Queue Link (Word 4) in the unit queue header of the unit queue is set to point to this unit queue.

In order to support unit queues, the MLIPs must share the horizontal queue array. This sharing is done by having each MLIP lock the entire horizontal queue array prior to queuing or unqueuing unit queues. Word 0 of the Horizontal Queue Array is used as the lock word.

PHYSICALIO/MLIP Interface

Figure 9 shows two command queues linked into the same horizontal queue.

```
                  Horizontal Queue Array
             |----------------------------------------|
             |Header|     |     | HQ Head|     |      |
             |Word  |     |     |   "Z"  |     |      |
             |--------------------|-------------------|
                                  | A
|---------------------------|  |
|                              |
|   Command Queue Header "A"   |   Command Queue Header "B"
|   |------------------------| |   |------------------------|
|-> | Queue Control Word     | |   | Queue Control Word     |
    |------------------------| |   |------------------------|
    | Head IOCB Link         | |   | Head IOCB Link         |
    |------------------------| |   |------------------------|
    | Tail IOCB Link         | |   | Tail IOCB Link         |
    |------------------------| |   |------------------------|
    | HQ Head Pointer      -->+<-- HQ Head Pointer          |
    |------------------------|     |------------------------|
    | HQ Link              ----->| HQ Link=NULL             |
    |------------------------|     |------------------------|
```

HQ = Horizontal queue

Figure 9. Multiple Command Queues Linked to a Horizontal Queue Array

When the MLIP finishes processing a command/unit queue (that is, when it cannot initiate any more commands from that command/unit queue), it checks the horizontal queue referenced by the command/unit queue header to see if other command/unit queues are waiting. If so, the MLIP unlinks and begins processing the first command/unit queue in the horizontal queue. In this fashion, the MLIP traverses the entire horizontal queue.

RESULT QUEUE

Result queues are lists of completed IOCBs. During system initialization, PHYSICALIO builds a result queue array for each processor in A 3, A 9, B 5900, and B 6900 systems. In A 3K and A 10 systems, only one result queue array is initialized. A result queue array contains a header word (which includes a Queue Length field and the code 4'10CF' as the Result Queue Mark) and a one-word Result Queue Head for each of several result queues. The exact number of result queues is determined by the number of classes of results that are to be queued separately. Typically, PHYSICALIO establishes seven result queues, one for each of the following result classes:

1. Normal I/Os
2. Internal PHYSICALIO ("kernel") I/Os
3. Peripheral status I/Os
4. Error IOCB I/Os
5. Datacomm I/Os
6. Intersystem control I/Os
7. Disk and pack I/Os

Before passing an IOCB to the MLIP, the MCP must indicate, in the Result Queue Head Pointer (Word 8) of the IOCB, which result queue the IOCB is to be queued into on completion. When the operation is complete, the MLIP sets the I/O Finish Time (Word 14) of the IOCB to the time of day, stores its MLIP result into the MLIP State and Result (Word 12), and stores the DLP result in the location specified by the DLP I/O Result Pointer (Word 5). The MLIP then links the IOCB at the head of the result queue specified (that is, the result queue is last-in, first-out) and decides whether or not to cause an I/O Finish interrupt (see "I/O Finish Interrupt").

Because several MLIPs can update the same result queue, the Next IOCB Link (Word 9) of the first IOCB is checked before the result queue is updated. The MCP checks the Next IOCB Link (Word 9) for its value. If the value is 1, the IOCB is not processed.

In the situation depicted by Figure 10, IOCBs 1 and 2 of command queue "A" have completed, IOCB 1 first; both are linked into result queue "L." Command queue header "A" is queued in horizontal queue "Z," waiting for the DLP to become available so that IOCB 3 can be initiated. When

PHYSICALIO/MLIP Interface

IOCB 3 completes, it will be linked into result queue "M." In A 3K and A 10 systems, a unit queue can be used in place of the command queue in this example.

```
                      Command Queue Header "A"
                     |---------------------------|
                     | Queue Control Word        |
                     |---------------------------|
              |------| Head IOCB Link            |
              |      |---------------------------|
              |      | Tail IOCB Link             ---------------------|
              |      |---------------------------|                     |
              |      | HQ Head Pointer-->"Z"     |                     |
              |      |---------------------------|                     |
              |      | HQ Link --> CQ Header "B" |                     |
              |      |---------------------------|                     |
              |                                                        |
              |---------------------------------|                      |
                                                |                      |
      IOCB 1                    IOCB 2          |        IOCB 3        |
|---------------|        |---------------|      |  |---------------|   |
|               |<--|    |               |<--|  |-->|            <--|
|---------------|   |    |---------------|   |     |---------------|
| CQ Header "A" |   |    | CQ Header "A" |   |     | CQ Header "A" |
|---------------|   |    |---------------|   |     |---------------|
| RQ Head   "L" |   |    | RQ Head   "L" |   |     | RQ Head   "M" |
|---------------|   |    |---------------|   |     |---------------|
| Next=NULL     |   |---- Next Link      |   |     | Next=NULL     |
|---------------|        |---------------|   |     |---------------|
|               |        |               |   |     |               |
|---------------|        |---------------|   |     |---------------|
                                             |
                         |-------------------|
                         |
      |------------------|-------------------------------------|
      |Header|      |RQ Head |RQ Head |     |     |     |      |
      |Word  |      |"L"     |"M"     |     |     |     |      |
      |------------------------------------------------------|
                        Result Queue Array
```

CQ = Command queue
HQ = Horizontal queue
RQ = Result queue

Figure 10. Multiple IOCBs Linked to a Result Queue Array

I/O FINISH INTERRUPT

When the MLIP has finished processing an IOCB and the IOCB has been linked into the appropriate result queue, the MLIP must decide whether or not to cause an I/O Finish interrupt to notify the processor. The MLIP will cause an I/O Finish interrupt for any one of the following conditions:

1. The MCP has requested an interrupt by setting the Cause I/O Finish Interrupt field in the MLIP Control Word (Word 0) of the completing IOCB.

2. The Exception field in the MLIP State and Result (Word 12) of the completing IOCB is TRUE.

3. The command/unit queue of the finishing IOCB is empty.

MLIP ERROR HANDLING

In addition to IOCBs that the MCP allocates to perform I/Os, a small number of IOCBs are allocated to provide the MLIP with a mechanism to report errors that are not relevant to the IOCB currently in progress, or for which there is too much information to be reported in the IOCB.

These "Error IOCBs" are initiated during system initialization, but are not completed until they are needed to report one of the following errors:

- An invalid queue structure (for example, bad queue mark).
- An invalid queue reference.
- A bad descriptor link received from a DLP (see the "MLIP/UIO Interface" section).
- A memory error detected by the MLIP.

Error IOCBs have their own command/unit queue and result queue. Because it is important that the Error IOCB command/unit queue not be suspended, the MCP and the MLIP take special precautions when handling Error IOCBs.

In order that the Exception field does not get set to TRUE, the following must happen: the MCP must issue all Error IOCBs with the Attention field set to FALSE; the MLIP must not set the DLP Error, MLIP/MLI Error, MLIP/Hardware Error, or Completed After Queue Suspended fields to TRUE. In order that the queue does not get suspended because the MLIP Suspend All Queues field is TRUE, the MCP must issue all Error IOCBs with the Ignore Suspend All Queues field set to TRUE.

The MCP issues all Error IOCBs with the Cause I/O Finish interrupt field set to TRUE. This action is required because the MLIP is not permitted to set any exception fields, and there is no other way for an interrupt to be generated.

See also

IOCB. 33
MLIP/UIO Interface. 71

7 MLIP

The Message-Level Interface Processor (MLIP) is a hardware module with the following purposes:

- To accept I/O operations from the Master Control Program (MCP).
- To queue the I/O operations.
- To route the I/O operations to their destination Data Link Processors (DLPs).
- To handle the Message-Level Interface (MLI) dialog with the DLP.
- To return the result information to the MCP.

Many of these functions are described in other sections of this manual, particularly in the "PHYSICALIO/MLIP Interface" section and the "MLIP/UIO Interface" section.

MLIP COMMANDS

The MLIP also performs operations itself. These functions are, for the most part, related to queue management and DLP configuration management and, with two exceptions, do not involve communication with the Universal I/O (UIO) subsystem.

The MCP requests MLIP operations by setting the MLIP/DLP Command field in the MLIP Control Word (Word 0) of an Input/Output Control Block (IOCB) to TRUE (indicating MLIP command) and storing, in the first word pointed to by the DLP I/O Command Pointer (Word 4) in the IOCB, a code indicating a command. The following commands are available on all systems that support the MLIP I/O subsystem:

Wait for Error

This command indicates to the MLIP that the IOCB is to be used as an Error IOCB (see "MLIP Error Handling").

Discontinue Error IOCB

This command discontinues the currently active Error IOCB, if any. If there is no Error IOCB currently active, the MLIP will indicate that there is no Error IOCB to discontinue. This operation is used when changing software environments, such as when entering and exiting TAPEDUMP, to remove all outstanding Error IOCBs.

Set the Suspend All Queues Flag

This command sets the MLIP Suspend All Queues field, which suspends any active command/unit queues so that no more I/Os can be initiated (see "Command Queue" and "Unit Queue"). This command is issued by the MCP through the IOFAUCET procedure. The purpose of the command is to prevent I/Os from being initiated during a memory dump.

Reset the Suspend All Queues Flag

This command resets the MLIP Suspend All Queues field.

Activate Queue

This command resets the Suspended field in the Queue Control Word (Word 0) of the command/unit queue specified by the Command or Unit Queue Header Pointer (Word 2) in the IOCB, and to start processing the queue. This command is issued by the MCP to restart a command/unit queue after the exception condition that caused the queue to be suspended has been handled appropriately.

Deactivate Queue

This command sets the Suspended field in the Queue Control Word (Word 0) of the command/unit queue specified by the Command or Unit Queue Header Pointer (Word 2) in the IOCB. This command suspends the command/unit queue before TEST/WAIT operations and before operations that may alter the queue structure, such as BLASTUNIT, PATHRES, and UNITMOVER.

Return Queue

This command returns the IOCBs queued in the command/unit queue specified by the Command or Unit Queue Header Pointer (Word 2). The MLIP returns the queue by copying the Head IOCB Link (Word 1) from the Command Queue Header into the first word pointed to by the DLP I/O Result Pointer (Word 5) and by setting the Head IOCB Link (Word 1), Tail IOCB Link (Word 2), and Inactive Count field of the Queue Control Word (Word 0) to 0. The MLIP will not remove the command/unit queue from the horizontal queue if it is currently linked. This command is issued by BLASTUNIT, for example, when it must alter the structure of the command/unit queue.

Read MLIP Status

This command returns one word of status information in the first word of the area pointed to by the DLP I/O Result Pointer (Word 5) in the IOCB. This status information includes the system type, the MLIP firmware revision, the Host Return field (see "Host Identification"), and a bit vector identifying the MLI ports that are present. This information is requested by the software during peripheral initialization. On A 3K and A 10 systems, the MLIP also returns information in bit 16 of the DLP I/O Result Pointer (Word 5) as to whether or not disk seek optimization has been implemented.

Read DLP Status

This command connects the MLIP to the DLP specified in the DLP Address Word (Word 1) of the IOCB and reads its status, which is returned in the MLIP State and Result (Word 12) of the IOCB.

Clear DLP

This command issues an "MLI selective clear" command to the DLP specified in the DLP Address Word (Word 1). This operation is used by IOFAUCET to clear TEST/WAIT operations when beginning a nonfatal dump and by BLASTUNIT when it must ensure that the DLP is in a known state.

General Clear

This command issues an "MLIP master clear" command to all MLI ports. This command is issued by IOFAUCET when beginning a fatal dump, and by the soft Halt/Load procedure "FAKEHALTLOAD" to simulate the action of a Halt/Load.

MLIP COMMANDS AVAILABLE ON A 3K AND A 10 SYSTEMS

The following are additional MLIP commands available on the A 3K and A 10 systems.

Return Active IOCB

This command returns the active count of the unit queue and Last IOCB Initiated (Word 5) of the unit queue to the first and second words pointed to by the DLP I/O Result Pointer (Word 5) of the IOCB. This command is used by IODISASTER when handling hung DLPs.

Set I/O Table EMPDS

This command updates the EMP Destination Set (EMPDS) (Word 5) of the MLIP I/O table. The MCP uses this command after system initialization to update the destination set with all of the running E-Mode processors (EMPs).

Set I/O Table MLIPDS

This command updates the MLIP Destination Set (MLIPDS) (Word 6) of the MLIP I/O table. IOFAUCET uses this operator to update the MLIP I/O table with a mask of the running MLIPs during a memory dump.

Set Unit Queue Path Word

This command stores the second word pointed to by DLP I/O Command Pointer (Word 4) of the IOCB in the Path Table Pointer (Word 9) of the unit queue. The MCP uses this operation whenever a path has been selected for a unit.

Set Unit Queue I/O Optimization Word 2

This command stores the second word pointed to by DLP I/O Command Pointer (Word 4) of the IOCB in the I/O Optimization Word 2 (Word 8) of the unit queue. This operation is used by the MCP to maintain data necessary for the MLIP to perform disk seek optimization.

Decrement Unit Active Limit

This command decrements the active limit field of the Queue Control Word (Word 0) of the unit queue. This operation is used by the MCP when maintenance tests are being run.

Increment Unit Active Limit

This command increments the active limit field of the Queue Control Word (Word 0) of the unit queue. This operation is used by the MCP when maintenance tests are being run.

See also
Command Queue 42
Host Identification 73
MLIP Error Handling 63
Unit Queue. 48

8 MLIP/UIO INTERFACE

The Message-Level Interface Processor (MLIP) communicates with the Universal I/O (UIO) subsystem through the Message-Level Interface (MLI), a protocol-driven interface that allows the MLIP and the Data Link Processors (DLPs) to process independently between data transfer bursts. When the MLIP has an I/O operation to transfer to a particular DLP, it connects to the DLP and transfers the DLP command and a "descriptor link" to the DLP. The descriptor link contains two items:

1. The Host Return field, required for the return routing of information from the DLP to the MLIP (see "Host Identification").

2. A tag that uniquely identifies a particular I/O operation.

After the DLP command and descriptor link have been transferred, the MLIP can disconnect and perform other operations.

When a DLP is ready to transfer data, it reconnects to the MLIP and transfers the descriptor link for the I/O operation requiring the data transfer. The MLIP uses this descriptor link to find the appropriate Input/Output Control Block (IOCB) and reconstructs its state by accessing the intermediate results it previously stored in the IOCB. One or more blocks of data are then transferred, and the MLIP disconnects.

When the DLP has finished the operation, it reconnects to the MLIP and transfers both the descriptor link and the DLP result descriptor for the completed operation. It then drops the connection, and the MLIP finishes processing the IOCB.

See also
Host Identification 73

9 UIO SUBSYSTEM

To determine the configuration of the I/O subsystem, PHYSICALIO must be able to uniquely identify every base, Data Link Processor (DLP), and unit in the subsystem. To initiate an I/O to a specific unit, PHYSICALIO must be able to select and specify a path to that unit. Similarly, to return information to the system, the I/O subsystem must be able to uniquely identify each host.

This section describes the rules and conventions that must be followed to establish a valid Universal I/O (UIO) subsystem configuration.

HOST IDENTIFICATION

From the UIO subsystem's point of view, a "host" is any system component that communicates with a base on a Message-Level Interface (MLI). A host can be an A 3, A 3K, A 9, A 10, B 5900, or B 6900 processor; a Network Support Processor (NSP); or a B 6900 maintenance processor.

Each host connecting to a base has an identification number, called a Host Return field, that is used by the base to uniquely identify the host. The Host Return field must be in the range 0 through 7. The B 5900 and B 6900 processors generate Host Return fields that correspond to their processor IDs and, thus, are numbered from 1 through 4. The B 6900 maintenance processors use a Host Return field of 0, leaving the range 5 through 7 available for NSPs.

For the A 3K and A 10 systems, the Host Return field corresponds to the MLIP number (1 through 7).

A host connects to a base through a distribution card (DC), which is identified with a number that matches the Host Return field of the host it is connected to. There can be up to six DCs in one base, allowing up to six hosts to connect to the same base. Hosts that connect to DCs within the same base must have unique Host Return fields, but hosts that do not share bases need not have unique Host Return fields. A configuration that includes nonunique Host Return fields is illustrated in Figure 11, which shows two NSPs (both numbered 6) that do not connect to the same base.

```
 |-------------------|       |-------------------|
 |                   |       |                   |
 | B 5900 Processor 1|       | B 5900 Processor 2|
 |                   |       |                   |
 |===================|       |===================|
 |       MLIP        |       |       MLIP        |
 |-------------------|       |-------------------|
 |Port|Port|Port|Port|       |Port|Port|Port|Port|
   |    |    |    |            |    |    |    |
  MLI            MLI          MLI            MLI
   |              |            |              |
|--|--|        |--|--|         |           |--|--|
| DC 1|        | DC 1|         |           | DC 2|
|-----|        |-----|         |           |-----|
|     |        | DC 2----------|     |------- DC 6|
|-----|        |-----|               |     |-----|
| BCC |      |-- DC 6|               |     | BCC |
|-----|      | |-----|               |     |-----|
|     |      | | DC 7------------|   |     | PSM |
|-----|      | |-----|           |   |     |-----|
|     |      | | BCC |           |   |--MLI--NSP 6|
|-----|      | |-----|           |         |-----|
|NSP 6--MLI--| | PSM |           |-----MLI--NSP 7|
|-----|        |-----|                     |-----|
|     |        | LSP --datacomm lines      | LSP--datacomm lines
|-----|        |-----|                     |-----|
|     |        | LSP --datacomm lines      |     |
|-----|        |-----|                     |-----|
               | LSP --datacomm lines
               |-----|
               |     |
               |-----|
```

BCC = Base control card
DC = Distribution card
LSP = Line Support Processor
MLI = Message-Level Interface
MLIP = Message-level Interface Processor
NSP = Network Support Processor
PSM = Path Selection Module

Figure 11. UIO Subsystem Configuration with Multiple Hosts

Each base must include a base control card (BCC). The BCC controls access to the DLPs by multiple hosts and also provides base identification information to the hosts upon request.

Bases that contain more than one distribution card must include a Path Selection Module (PSM), which handles priority resolution and routing of messages being returned to the hosts.

BASE IDENTIFICATION

It is important that each base be uniquely identifiable so that a base accessed through multiple paths can be recognized as a single base. A base's identification number is derived from its maintenance identification number because each base must already have a unique identification number for maintenance purposes and because it is desirable for a processor and its maintenance processor to display the same identification number when referring to the same base.

Every B 5900 or B 6900 processor has an associated maintenance processor. The maintenance processor is connected to a maintenance test bus, which allows maintenance routines to communicate with each base through a maintenance card (MC). Figure 12 represents a sample B 5900 processor configuration, including the maintenance processor.

PHYSICAL I/O OVERVIEW

```
|--------------------|
|                    |--------------|
| B 5900 Processor 1| Maintenance |
|                    |  Processor  --diskette
|====================|----|---:----|
|        MLIP        |         :
|--------------------|    |    :
|Port|Port|Port|Port|         :
   |    |    |    |       |    :
  MLI            MLI           :
   |              |       |    :
|--|--|        |--|--|         :
| DC 1|        | DC 1|    |    :
|-----|        |-----|         :
|     |        |     |    |    :
|-----|        |-----|         :
| BCC |        | BCC |    |    :
|-----|        |-----|         :
|     |        |     |    |    :
|-----|        |-----|         :
|TP-DLP--TP    |ODT-DLP---|    :
|-----|        |-----|         :
|CR-DLP--CR    |MT-DLP--MEC    :
|-----|        |-----|         :
|HT-DLP--DPDC  |     |   Maintenance
|-----|        |-----|   Test Bus 1
|MC2/1|        |MC1/1|         :
|--:--|        |--:--|         :
   :..............:............:
```

BCC = Base control card	MEC = Master Electronic Control
CR = Card reader	MLI = Message-Level Interface
DC = Distribution card	MLIP = Message-Level Interface Processor
DLP = Data Link Processor	MT = Magnetic tape
DPDC = Disk Pack Drive Controller	ODT = Operator Display Terminal
HT = Host Transfer	TP = Train printer
MC = Maintenance card	

Figure 12. B 5900 Processor and Maintenance Processor Configuration

Every base must be connected to a maintenance test bus. Each maintenance test bus number must be unique within the entire system, and on a given maintenance test bus, each base address must be unique. Thus, each base can be uniquely identified by its maintenance test bus number and address.

UIO Subsystem

In the configuration above, the maintenance test bus numbered 1 connects to two bases, which have been assigned the maintenance test bus addresses 1 and 2. This is shown by the notations "1/1," indicating "address 1/bus 1," and "2/1," indicating "address 2/bus 1." Note that the maintenance processor is shown connected to an Operator Display Terminal-DLP (ODT-DLP). This connection is required only if the maintenance processor's display terminal is also to function as an ODT. The maintenance processor configuration for a B 6900 system is substantially different from this B 5900 processor configuration (see "B 6900 Maintenance Processor Configuration").

Each base includes a BCC, which will respond to a Test ID operation by returning a 16-bit, field-strappable base identification: 8 bits representing the maintenance test bus address of the base, 4 bits representing the maintenance bus number, and 4 bits that are currently unused and must be 0.

See also

B 6900 Maintenance Processor Configuration. 85

PHYSICAL I/O OVERVIEW

DLP AND UNIT IDENTIFICATION

The DLP ID is a 16-bit number with two components:

1. An 8-bit, field-strappable base unit number

2. An 8-bit DLP type identification

The DLP type identification specifies the type of the DLP (for example, a Magnetic Tape DLP (MT-DLP) or Card Reader DLP (CR-DLP)). The base unit number identifies the units connected to that DLP. Physical unit numbers are generated by assigning to the first unit on the DLP a unit number equal to the base unit number. Each subsequent unit (or potential unit) is assigned the next higher number, as shown in Figure 13 of an MT-DLP.

```
             |- MT Unit 48
             |- MT Unit 49
|-----|      |-
             |- MT Unit 51
|     |      |-
             |-
|-----|      |- MT Unit 54
|MT-DLP--MEC-|-
| 48  |      |-
|-----|      |-
             |-
|     |      |- MT Unit 59
             |-
|-----|      |-
             |-
             |- MT Unit 63
```

MEC = Master Electronic Control
MT Unit = Magnetic tape unit
MT-DLP = Magnetic Tape Data Link Processor

Figure 13. Physical Unit Numbers Assigned According to Base Unit Number

There must be a one-to-one correspondence between units and unit numbers; that is, multiple units cannot be assigned the same unit number, and multiple unit numbers cannot be assigned to the same unit. These rules imply the following restrictions on the assignment of DLP base unit numbers:

UIO Subsystem

1. Base unit numbers for DLPs that connect to different units must be unique.

2. Base unit numbers must be assigned such that there is no possibility for the range of potential unit numbers to overlap. For example, in a configuration that included the MT-DLP shown in Figure 13, a DLP could not be assigned a base unit number of 62, because the range of possible unit numbers for units connected to MT-DLP 48 includes unit 62.

3. The range of unit numbers that must be considered "occupied" for a given DLP depends on the type of DLP.

4. All DLPs that connect to the same units through an exchange must have the same base unit number, so that the unit numbers generated for the units will be the same through all paths.

The unit number 0 is not allowed to be assigned.

PHYSICAL I/O OVERVIEW

Table 6 shows the types of DLPs currently supported and the number of potential units that can be configured behind DLPs of each type.

Table 6. Supported DLPs and Maximum Units per DLP

DLP	Peripheral	No. of Units
CR-DLP	B9115/6/7 Card Reader	1
TP-DLP	400-/750-/1100-/1500-lpm Train Printer	1
TP2-DLP	B9246-X Model Printers	1
TP3-DLP	B924 Model Drum Printer	1
PT-DLP	B9246-X/B924 Model Printers	1
	B9498 Streamer Tape	4
MT-DLP	PE Tape	16
NRZ-DLP	9-Track NRZ Magnetic Tape	16
GCR-DLP	GCR Magnetic Tape	16
CP-DLP	Card Punch	1
5NDF-DLP	5N Disk file	16
HT-DLP	(Host Transfer)	16
	225/235/206/207/659/677 Disk Pack	
ODT-DLP	Operator Display Terminal	3
LSP-DLP	Datacomm lines	1*
NSP-DLP	Datacomm network (LSPs)	1*

* Defined to be 1 for this purpose. LSPs and individual datacomm lines are selected by other means.

CP = Card Punch
CR = Card Reader
DLP = Data Link Processor
GCR = Group Coded Recording
HT = Host Transfer
lpm = Lines per minute
LSP = Line Support Processor
MT = Magnetic tape
NSP = Network Support Processor
NRZ = NonReturn to Zero
ODT = Operator Display Terminal
PE = Phase-Encoding
PT = Printer tape
TP = Train printer
5NDF = 5N disk file

PATH SPECIFICATION

When requesting that an I/O be performed, PHYSICALIO selects a particular path by including the following information in the IOCB for the I/O:

1. The MLIP number

2. The MLI port number

3. The Line Expansion Module (LEM) port number (described below)

4. The DLP address (described below)

5. The relative unit number of the unit (that is, the DLP-relative offset of the unit)

A unique base is identified by the combination of MLIP number, MLI port number, and LEM port number. An LEM provides the capability to expand one MLI into several, allowing multiple bases to connect to one MLI port.

The DLP address is a number that identifies the DLP within the selected base. The DLP address is also used by the PSM to resolve conflicts when multiple DLPs are ready to transmit data at the same time; DLPs with higher addresses have higher priority.

For A 3, A 9, B 5900, and B 6900 systems, Table 7 indicates the current valid values for selection numbers.

PHYSICAL I/O OVERVIEW

Table 7. Valid Selection Numbers to Identify I/O Path for A 3, A 9, B 5900, and B 6900 Systems

	B 5900	B 6900	A 3	A 9
MLIP Number *	1-4	1-4	1-7	1-7
MLI Port Number	0-3	0-7	0-2	0-7
LEM Port Number **	0-7	0-7	0-7	0-7
DLP Address	0-7	0-7	0-7	0-7
Relative Unit Number ***	0-15	0-15	0-15	0-15

* Matches the processor ID.
** A LEM can have either 4 or 7 ports; the port numbers are in the range 0 through 7.
*** Depends on DLP type; 15 is maximum for any DLP.

For A 3K and A 10 systems, Table 8 indicates the current valid values for selection numbers.

Table 8. Valid Selection Numbers to Identify I/O Path for A 3K and A 10 Systems

	A 3K	A 10
MLIP Number	1-7	1-7
MLI Port Number	0-2	0-7
LEM Port Number *	0-7	0-7
DLP Address	0-7	0-7
Relative Unit Number **	0-15	0-15

* A LEM can have either 4 or 7 ports; the port numbers are in the range 0 through 7.
** Depends on DLP type; 15 is maximum for any DLP.

UIO Subsystem

Figure 14 shows a disk unit reached by the path specified by MLIP 1, MLI port 0, LEM port 3, DLP address 2, relative unit 3. The disk unit is identified with [*] at the end of the figure.

```
                    |--------------------|
                    |                    |
                    |      Processor     |
                    |                    |
                    |====================|
                    |       MLIP 1       |
                    |--------------------|
                    |Port|Port|Port|Port|
                      |    |    |    |
                     MLI
                      |
|----------------|-----------------|
|               LEM                |
|----------------------------------|
|Port|Port|Port|Port|Port|Port|Port|
  |    |    |    |    |    |    |
                MLI
             |--|---|
             | DC   |
             |------|
             | BCC  |
             |------|
           0:|TP-DLP---Train Printer
             |------|
           1:|CR-DLP---Card Reader
             |------|
           2:|HT-DLP----|-----------------|
             |------|   |      DPDC       |
             | MC   |   |-----------------|
             |------|   |    Exchange     |
                        |-|-|-|-|-|-|-|-|-|
                          | | | | | | | |
                          0 1 2 3 4 ... 15
                                [*]
```

```
BCC  = Base control card            LEM  = Line Expansion Module
CR   = Card reader                  MC   = Maintenance card
DC   = Distribution card            MLI  = Message-Level Interface
DLP  = Data Link Processor          MLIP = Message-Level Interface
DPDC = Disk Pack Drive Controller          Processor
HT   = Host Transfer                TP   = Train printer
```

Figure 14. I/O Path from Processor to Disk Unit

PHYSICAL I/O OVERVIEW

In addition to the path specifications described above, the IOCB contains the processor ID of the processor that initiated the I/O. For A 3, A 9, B 5900, and B 6900 systems, when an IOCB is unlinked from a result queue by PHYSICALIO, this processor ID is used to determine which processor is to finish processing the I/O. For A 3K and A 10 systems, any processor can finish processing any I/O.

See also
Processing of I/Os. 21

B 6900 MAINTENANCE PROCESSOR CONFIGURATION

A sample B 5900 processor configuration including a maintenance processor was illustrated in Figure 12. Although the maintenance test bus connects to the bases in the same manner on the B 6900 system as it does on the B 5900 system, the B 6900 maintenance processor requires access to some peripherals that are not required by the B 5900 maintenance processor, necessitating some additional configuration restrictions for the B 6900 system.

Because the B 6900 maintenance processor does not include a terminal for interacting with the system operator, the maintenance processor must have access to an ODT through an ODT-DLP. In addition, the maintenance processor requires access to a magnetic tape unit through an MT-DLP to load various programs and data (this information is accessed from a diskette on the B 5900 system). These DLPs must be located in the same base; this base is referred to as the "maintenance base" and is shown in Figure 15.

```
    |--------------------|
    |                    |-------------|
    | B 6900 Processor 1| Maintenance |
    |                    |  Processor  |
    |====================|----|---:----|
    |       MLIP         |    |   :
    |--------------------|    |   :
    |Port|Port|Port|Port|     |   :
      |    |    |     |       |   :
     MLI             MLI     MLI  :
      |               |       |   :
  |---|---|       |---|---|   |   :
  |  DC 1 |       |  DC 1 |   |   :
  |-------|       |-------|   |   :
  |       |       |  DC 0 ---|    :
  |-------|       |-------|       :
  |  BCC  |       |  BCC  | Maintenance
  |-------|       |-------| Test Bus 1
  |       |       |  PSM  |       :
  |-------|       |-------|       :
  |TP-DLP--TP     |ODT-DLP--ODT   :
  |-------|       |-------|       :
  |HT-DLP--DPDC   |MT-DLP---MEC   :
  |-------|       |-------|       :
  |       |       |       |       :
  |-------|       |-------|       :
  | MC2/1 |       | MC1/1 |       :
  |---:---|       |---:---|       :
      :...............:...........:
```

BCC = Base control card	MC = Maintenance card
DC = Distribution card	MLI = Message-Level Interface
DLP = Data Link Processor	ODT = Operator Display Terminal
DPDC = Disk Pack Drive Controller	PSM = Path Selection Module
HT = Host Transfer	TP = Train printer

Figure 15. B 6900 Processor and Maintenance Processor Configuration

Because all B 6900 maintenance processors generate a Host Return field of 0, a base can be the maintenance base for only one maintenance processor (because there can be only one distribution card 0). The maintenance test bus that connects to a maintenance base must originate in the maintenance processor associated with that base.

The maintenance base must be visible to (that is, shared with) only that processor associated with the maintenance processor. To simplify the path specification, there must not be a LEM on the MLI between the maintenance processor and its maintenance base.

10 SYSTEM INITIALIZATION

For the Master Control Program (MCP) on a Universal I/O (UIO) subsystem to begin system initialization, there must be valid bootstrap code. On A 3, A 9, B 5900, and B 6900 systems, this bootstrap code is resident in the low addresses of memory. On A 3K and A 10 systems, the bootstrap code is loaded into memory from disk by the maintenance processor. The steps of the system initialization process, described below, begin at bootstrap code initialization.

1. The bootstrap code can begin executing because of a Halt/Load, a Change MCP (CM), a fatal tape dump, a reconfiguration command, or some other software cause. The bootstrap's main task is to initiate the GETITGOING procedure. The bootstrap has sufficient path information to perform I/Os to the Halt/Load unit, from which it reads the code for GETITGOING.

2. GETITGOING reads in the necessary MCP data structures and "save" code. It also reads in and invokes all of the initialization procedures that select alternative modules based on the type of machine. These modules include PHYSICALIO (whose initialization procedure selects the PHYSICALIOHDP alternative for UIO subsystems), datacomm, maintenance, configuration control, and processor control. GETITGOING then calls PRIMARYINITIALIZE.

3. PRIMARYINITIALIZE builds a more permanent stack for the MCP and moves to it, entering SECONDARYINITIALIZE in the process.

4. SECONDARYINITIALIZE is the first procedure that recognizes the potential existence of other tightly-coupled processors. If there are multiple processors, one processor determines that it is the "leader" in the initialization process and coordinates the initialization of the other processors. The substeps of SECONDARYINITIALIZE describe the single-processor system; the multiple-processor system follows the same basic pattern, with the added complication of having to synchronize the leader and follower processors.

   After establishing its processor and memory environment, SECONDARYINITIALIZE calls PERIPHERALINITIALIZE.

   a. PERIPHERALINITIALIZE allocates space for I/O tables such as UNIT, UNITCONTROL, UNITMAP, and UNITSTATUS. It then calls PERIPHERALCONFIGURATOR with a parameter indicating that this call is from PERIPHERALINITIALIZE (PERIPHERALCONFIGURATOR is called at other times, such as when adding an outboard base and when freeing or acquiring a Data Link Processor (DLP)). PERIPHERALCONFIGURATOR then calls CONFIGPERIPHERALINITIALIZE.

b. The main task of CONFIGPERIPHERALINITIALIZE is to loop through all possible Message-Level Interface (MLI) ports and Line Expansion Module (LEM) ports, issuing a TEST UNIT operation to each potential base (see the "UIO Subsystem" section for configuration requirements). When a base is found, CONFIGPERIPHERALINITIALIZE calls ADDABASE.

c. ADDABASE performs a TEST ID operation on each potential DLP in the base, storing all of the resulting information in a base entry template that is to be merged into the configuration table. ADDABASE then calls MERGEBASEENTRY.

d. MERGEBASEENTRY determines whether or not there is already information in the configuration table about this base (for example, whether another processor has found it first or whether the base is described in the soft configuration file). If so, the base and DLP information derived by ADDABASE is merged in with the existing information (which may result in errors if the overlapping information is incompatible). If the base has not been seen before, a new base entry is added to the configuration table and ADDANENTRYTOOTHERTABLES is called once for each DLP in the base.

e. ADDANENTRYTOOTHERTABLES determines whether this DLP connects to units that have already been seen through another path or whether this DLP connects to new units. If the units have already been seen, ADDAPATH is called to add the current path to the existing list of paths under the "path node" for those units. If the DLP connects to new units, a new path node is entered in the path table, the current path is added to the path list, and each potential unit behind the DLP is assigned a physical and logical unit number and is added to the various physical and logical unit tables. If the units are of a type that cannot be configured behind a peripheral exchange (that is, if the units are "nonexchangeable") and if the units are of a type that requires peripheral status monitoring, STARTUNIT is called to start peripheral status monitoring for the units (see the "Peripheral Status" section).

After PERIPHERALINITIALIZE has been completed, SECONDARYINITIALIZE initiates the independent runners (including ETERNALIR and STARTSYSTEM), finishes the remainder of its processor synchronization code, and begins normal process switching.

5. STARTSYSTEM is left to finish the high-level initialization of the system, including setting up the information required for managing disk files. STARTSYSTEM calls ACQUIREUNITS to verify that there are no unit-ownership conflicts among

loosely-coupled systems; if a conflict is detected, the initializing system will free the unit. In addition, ACQUIREUNITS attempts to acquire any exchangeable units that may belong to this group in a loosely-coupled environment. Once these units (if any) have been acquired, STARTSYSTEM calls PERIPHERALCONFIGURATOR (through SETUNITINFO) with a parameter indicating that peripheral status monitoring is to be initiated for all exchangeable units (see the "Peripheral Status" section).

See also

11 PERIPHERAL STATUS

Monitoring a unit's peripheral ready/not-ready status is performed by issuing special TEST operations to each unit. There are four types of TEST operations involved:

1. TEST UNIT

2. TEST/WAIT FOR READY

3. TEST/WAIT FOR NOT READY

4. TEST/WAIT FOR TRANSMIT (valid only for Operator Display Terminals (ODTs))

A TEST UNIT operation will finish "immediately" (that is, without waiting), returning identification and status information for the unit to which it was directed. In contrast, a TEST/WAIT operation to a unit will not finish until the unit's status matches the status specified (for example, becomes "ready," in the case of TEST/WAIT FOR READY); the I/O Finish interrupt for a TEST/WAIT operation notifies the system that the unit requires attention, as its status may have changed.

Most types of devices are monitored by alternating TEST/WAIT FOR READY operations with TEST/WAIT FOR NOT READY operations. However, the peripheral status monitoring process must take into account a few device dependencies. First, peripheral status is not monitored for devices for which it is not meaningful, such as Network Support Processors (NSPs) and Line Support Processors (LSPs). Second, ODTs are monitored with TEST/WAIT FOR TRANSMIT operations, instead of TEST/WAIT FOR READY and TEST/WAIT FOR NOT READY operations, because of the nature of the device.

The maintenance of valid peripheral status information depends on the issuing of an initial TEST/WAIT operation when the unit first becomes visible to PHYSICALIO. During system initialization, peripheral status monitoring is initiated for nonexchangeable devices by ADDANENTRYTOOTHERTABLES and, later, for exchangeable devices by STARTSYSTEM. When a unit is taken from PHYSICALIO's control (through TAKEUNIT), monitoring of peripheral status is stopped; when the unit is returned, GIVEUNIT reinitiates peripheral status monitoring if requested by the appropriate parameter value. Other processes by which units may become visible, such as through the ODT commands ACQUIRE and UR-, also include initiation of peripheral status monitoring.

Once the first TEST/WAIT operation has been issued, the process is self-sustaining (a new TEST/WAIT operation being issued whenever the current one completes) until monitoring is canceled for some reason (for example, by the ODT commands UR and FREE). The following sequence of events occurs whenever a TEST/WAIT operation has completed:

1. IOFINISH68 or IOFINISH_ASIO is called by HARDWAREINTERRUPT to handle the I/O Finish interrupt. When IOFINISH68 or IOFINISH_ASIO determines that the I/O that finished is a peripheral status monitoring operation, IOSTATUS is called.

2. If the completed TEST operation is a TEST/WAIT FOR TRANSMIT operation, IOSTATUS initiates KEYIN to handle the operator input and to initiate a new TEST/WAIT FOR TRANSMIT operation. If the TEST operation was a TEST/WAIT FOR READY or TEST/WAIT FOR NOT READY operation, IOSTATUS determines whether or not the device has actually changed logical or physical status; if so, IOSTATUS updates the UNITSTATUS table to reflect the new status of the device, queues a message for PHYSICALPERIPHERALSTATUS indicating that the unit's status has changed, and initiates a new TEST/WAIT operation to wait for the opposite status condition.

Note

> If an exception occurs, such as a TEST/WAIT FOR READY operation finishing with a NOT READY peripheral status, the unit is marked logically NOT READY and peripheral status monitoring for that unit ceases.

PHYSICALPERIPHERALSTATUS is called by ETERNALIR either when the MCP requests that it be called due to a status change on a unit, or approximately once every second. In other words, PHYSICALPERIPHERALSTATUS is called either on demand or at regular intervals. PHYSICALPERIPHERALSTATUS reads each message in its queue, taking one or more of the following actions as appropriate:

1. If the message indicates that peripheral status monitoring is to start for a unit (for example, if GIVEUNIT queues such a message), DOSTATUSIO is called to initiate a TEST/WAIT operation of the type specified in the message.

2. If peripheral status monitoring is to stop for a unit (for example, if TAKEUNIT queues such a message), DOSTATUSIO is called to cancel the current TEST/WAIT operation on the unit. Because the command queue for the unit is marked "Suspended" when a TEST/WAIT operation is initiated, DOSTATUSIO initiates a cancel or discontinue operation with an IOCB with the Immediate field in the MLIP Control Word (Word 0) set to TRUE, causing the MLIP to initiate the cancel operation regardless of the fact that the command queue is suspended. When the TEST/WAIT

operation has been canceled, DOSTATUSIO initiates an MLIP Activate Queue command to reset the Suspended field for the command queue.

3. If the message indicates that a peripheral's status has changed, LOGICALPERIPHERALSTATUS is called. LOGICALPERIPHERALSTATUS updates the status for the unit in the UNIT table and takes other action as appropriate based on the unit's type and its current status. For example, if a magnetic tape unit changes status from not ready to ready, READALABEL is initiated.

12 DATA COMMUNICATIONS

Data communications operations are handled by three special-purpose Data Link Processors (DLPs): the Network Support Processor (NSP), the Line Support Processor (LSP), and the Data Communications Data Link Processor (DC-DLP). This section describes these DLPs and the functions provided by the DCCONTROL and PHYSICALIO modules of the Master Control Program (MCP) to support data communications.

NSPs, LSPs, AND DC-DLPs

NSPs and LSPs are programmable DLPs that provide data communications services. Both are programmed using the Network Definition Language II (NDLII). LSPs run the Adaptor Control portion of the NDLII code, and NSPs run the Editors and Line Control portions of the NDLII code.

Each DLP also runs an Executive program. The NSP Executive maintains a dialog with an MCP process called DCCONTROL and with the LSPs it controls. The LSP Executive supports the dialog with the NSP. DC-DLPs are NSPs that have built-in LSPs. DC-DLPs are not programmable with NDLII and only run specific protocols. The MCP communicates with DC-DLPs in the same manner as do NSPs.

For its configuration tables, PHYSICALIO considers NSPs, LSPs, and DC-DLPs to be single-unit DLPs. Message routing to a particular LSP, line, and station is performed by the NSPs, LSPs, and DC-DLPs themselves. Although all LSPs must be in bases that are visible to their controlling NSPs, they need not be in bases that are visible to the host. NSPs act as "outboard hosts," providing an indirect communication path between the LSPs and the host (see the "UIO Subsystem" section for a configuration diagram that includes NSPs and LSPs). Because LSPs may be "invisible" to the host during system initialization, PHYSICALIO must be able to add LSPs to its tables as datacomm is being initialized.

See also
UIO Subsystem 73

PHYSICAL I/O OVERVIEW

DATACOMM I/O REQUESTS

A typical datacomm I/O request from a user program follows a path schematically similar to the following:

```
program -> DCCONTROL -> PHYSICALIO -> UIO subsystem
```

The datacomm I/O request (for example, a read or a write) is passed to the MCP DCCONTROL process, which calls a PHYSICALIO interface procedure to initiate the actual I/O operation. From PHYSICALIO to the destination terminal, the I/O can follow the following paths:

```
PHYSICALIO -> MLIP -> NSP -> LSP -> terminal
                      |            |
                      |-> DC-DLP -->|
```

The I/O is passed to the Message-Level Interface Program (MLIP), which routes the request to the NSP. The NSP then sends a request through a Message-Level Interface (MLI) to the LSP that controls the line connected to the terminal.

This transmission process involves many levels of communication. For example, at a basic level, the NSP communicates with the LSP over an MLI; at a higher level, the NSP communicates control information and data to the LSP through the messages transmitted over the MLI. Similarly, DCCONTROL initiates normal I/O operations through PHYSICALIO to the NSP; at a higher level, the data transferred by these I/O operations is used to maintain a request/response dialog between DCCONTROL and the NSP.

DATACOMM INITIALIZATION

There are several conditions that will cause datacomm to be initialized, including the entering of certain Operator Display Terminal (ODT) commands and the setting of some system options. When datacomm is to be initialized for an NSP or DC-DLP, the MCP process DCCONTROL is initiated. (There is a separate activation of DCCONTROL for each NSP; for simplicity, only a single activation is described here.) DCCONTROL calls DCINITIAL to begin the initialization. DCINITIAL performs the following steps:

1. DCINITIAL first calls procedures in the DATACOMSUPPORT library to build the datacomm tables.

2. DCINITIAL then calls the PHYSICALIO procedure INITIALIZEDCLCP to load the NSP firmware.

3. Once the NSP firmware has been loaded, ADDANOUTBOARDBASE is called; it performs TEST operations through the NSP to the base containing its LSPs so that the LSP base can be added to the configuration tables.

4. Next, DCINITIAL allocates two Input/Output Control Blocks (IOCBs): one is used to transmit the NSP requests required to establish the network configuration and to load the NDLII code from the DATACOMINFO file, and the other is used to receive the NSP's acknowledgments of the requests. These I/Os are initiated through the PHYSICALIO interface procedure INITIATECHARIO.

DCCONTROL then calls SCIOINITIALIZE to set up the I/O structures for normal operation.

During normal operation, each NSP has three allocated command queues. One of these queues is for read operations. The other two queues are used for write operations, one for normal output and one for control. Each queue has an active limit of one.

SCIOINITIALIZE allocates a large number of IOCBs and input buffers of various sizes for read operations; these IOCBs are then initiated, through INITIATECHARIO, into the input command queue. Additional IOCBs are allocated for output operations, but are not initiated until required.

PHYSICAL I/O OVERVIEW

Prior to beginning normal operation, DCCONTROL calls DCPCONTROLLER to process any spontaneous input messages that were received during the initialization phase.

NORMAL DATACOMM OPERATION

A typical request/response cycle requires two physical I/Os: a write to send the NSP request and a read to receive the NSP response. For example, to perform a write to a terminal, DCCONTROL would initiate an IOCB with the following information in the I/O buffer:

NSP Request Information (for example, WRITE)	Identification Tag	NDLII Program Control Information	Data for the Terminal

When the NSP has received the request through the MLI, it sends a DLP result to the MLIP; the MLIP stores the result information, and FINISHOFFIO causes the event specified in the IOCB, which notifies DCCONTROL only that the NSP has received the request. The NSP then transfers the data for the terminal to the LSP in a message that also contains LSP control information. When the data has been written to the terminal, the LSP returns a result to the NSP. The NSP then formats a result message and sends it to the system. Multiple results can be transferred to the I/O buffer. The data returned includes the following information:

NSP Result Information	Identification Tag	NDLII Program Result Information

When this I/O completes, DCCONTROL uses the identification tag to associate the response with its corresponding request and calls DCPCONTROLLER to return the result information to the requesting program.

In steady-state operation, DCCONTROL uses the "POBOX" mechanism to wait for one of several conditions to happen. The POBOX mechanism allows a process to assign an identifying "signature" to each of an arbitrary number of events, called "MEMOs." The process then monitors its POBOX until one of two conditions occurs: a MEMO is received, indicating that the event associated with that MEMO has happened, or a specified timeout period elapses. MEMOs are queued, if necessary, and are returned to the process in chronological order of their happening. Based on the signature, the process takes the appropriate action for each MEMO. DCCONTROL assigns MEMOs for the following events:

PHYSICAL I/O OVERVIEW

- Completion of a write operation
- Completion of a read operation
- Queuing of an I/O request from a program
- The DCCONTROL stack's EXCEPTIONEVENT

When a write operation completes, DCCONTROL notes that the I/O was completed and deallocates the data buffer but takes little additional action until the read of the NSP result completes.

When a read operation is complete, DCCONTROL examines each of the messages in the input buffer. If a message indicates that a requested operation is now complete, DCCONTROL calls DCPCONTROLLER to pass the result information to the requesting program. At this point, DCCONTROL can initiate writes to the NSP for previously requested actions that were queued until the proper conditions existed.

When an I/O request is queued, DCCONTROL interprets the request and initiates the appropriate I/O(s) or queues the request itself for later action.

When DCCONTROL's EXCEPTIONEVENT is caused, DCCONTROL analyzes its TASKVALUE to determine what action to take. If an NSP dump is requested, DCCONTROL calls the PHYSICALIO procedure DUMPDCDLP. If termination of datacomm was requested, DCCONTROL calls SCIOSHUTDOWN, (as described in "Datacomm Termination"), and then terminates.

When no MEMOs are received, DCCONTROL unconditionally waits 140 milliseconds and then processes any queued MEMOs. If there are no MEMOs queued, DCCONTROL waits ten seconds for a MEMO to be queued. If a MEMO is not queued in ten seconds, a null request message is sent. After the third consecutive null request message is sent with no MEMO received, DCCONTROL displays a message indicating that the NSP is not responding.

See also
Datacomm Termination. 101

DATACOMM TERMINATION

Termination of a datacomm process can result from any one of several operator input messages. When DCCONTROL detects that it has been terminated, it calls SCIOSHUTDOWN. SCIOSHUTDOWN calls BLASTUNIT to cancel all outstanding I/Os and returns canceled results to the requesting program(s). DCCONTROL then calls DCTERMINATE, which gives PHYSICALIO control (through GIVEUNIT) of the NSP and its LSPs, returns program requests that were queued waiting for initiation, and deallocates all message areas. If the last DCCONTROL process is terminating, DCTERMINATE also deallocates all datacomm tables.

GLOSSARY

base

A hardware unit in the Input/Output (I/O) subsystem that contains the components necessary for handling the routing of messages between the Data Link Processors (DLPs) and the host processor through the Message-Level Interface Processor (MLIP). The base itself contains at least one DLP.

base control card (BCC)

A base module that identifies a base and controls access to its Data Link Processors (DLPs) and to the base itself.

BCC

See "base control card."

Card Reader Data Link Processor (CR-DLP)

A processor that serves as the controller of a card reader and provides the I/O interface between the system and the card reader.

Change MCP (CM)

An Operator Display Terminal (ODT) command that specifies a particular Master Control Program (MCP) code file on an optionally specified disk pack family as the MCP code file to be used at the next Halt/Load.

CM

See "Change MCP."

Communicate with Universal I/O (CUIO)

An operator that passes the address of an I/O Control Block (IOCB) to the Message-Level Interface Processor (MLIP) for initiation.

CR-DLP

See "Card Reader Data Link Processor."

CUIO

See "Communicate with Universal I/O."

DATACOMINFO file

A file that contains a complete description of the datacomm configuration, including algorithms, editors, and translate tables. This is the file that the Interactive Datacomm Configurator (IDC) modifies and from which the Master Control Program (MCP) initializes the data communications subsystem.

Data Communications Data Link Processor (DC-DLP)

A data communications processor that combines the functions of the Network Support Processor (NSP) and Line Support Processor (LSP) into one physical Data Link Processor (DLP) and supports up to four lines of communication.

Data Link Processor (DLP)

A processor that serves as the controller of one or more peripheral devices or data communications lines and provides the interface between the system and the peripherals and lines.

Data Link Processor Address Word (DLPAW)

Contains several fields used by the Message-Level Interface Processor (MLIP) to address a Data Link Processor (DLP).

DC

See "distribution card."

Glossary

DC--DLP

See "Data Communications Data Link Processor."

Disk Pack Drive Controller (DPDC)

An interface between the Data Link Processor (DLP) and disk pack units.

distribution card (DC)

A base module that acts as an interface between a host and a base.

DLP

See "Data Link Processor."

DLPAW

See "Data Link Processor Address Word."

DPDC

See "Disk Pack Drive Controller."

EMP

See "E-Mode Processor."

EMP Destination Set (EMPDS)

Contains the destination set of E-Mode processors that are signaled when an I/O has finished.

EMPDS

See "EMP Destination Set (EMPDS)."

PHYSICAL I/O OVERVIEW

E-Mode Processor (EMP)

The central processing unit in Burroughs E-Mode architecture.

FIB

See "File Information Block."

File Information Block (FIB)

Contains information about the user I/O that is passed to the PHYSICALIO module.

GCR-DLP

See "Group Coded Recording Data Link Processor."

Group Coded Recording Data Link Processor (GCR-DLP)

A processor that serves as the controller of a Group Coded Recording (GCR) magnetic tape drive and provides the I/O interface between the system and the GCR magnetic tape drive.

Halt/Load

A process that starts or restarts the Master Control Program (MCP).

HDU

See "Host Data Unit."

Host Data Unit (HDU)

The B 7900 or A 15 system host interface to the Input/Output (I/O) subsystem. An HDU is configured with three host-dependent ports, each of which supports two Message-Level Interface (MLI) cables. A B 7900 or A 15 host processor can have more than one HDU. The B 7900 and A 15 systems are called "HDU machines."

Glossary

Host Transfer Data Link Processor (HT-DLP)

A processor that communicates information to the Disk Pack Drive Controller (DPDC) that interfaces up to 16 disk drive units.

HT-DLP

See "Host Transfer Data Link Processor."

input/output (I/O)

An operation in which the system reads data from or writes data to a peripheral device such as a disk drive.

Input/Output Control Block (IOCB)

A data structure used for communication between the host system and the Message-Level Interface Processor (MLIP).

Input/Output Control Word (IOCW)

Area in the IOCB where information about the I/O to be performed resides.

IOCB

See "Input/Output Control Block."

IOCW

See "Input/Output Control Word."

I/O

See "input/output."

LEM

See "Line Expansion Module."

Line Expansion Module (LEM)

A hardware module that allows the attachment of several bases to a single Message-Level Interface Processor (MLIP) port.

Line Support Processor (LSP)

On Message-Level Interface Processor (MLIP) systems, the data communications subsystem processor that manages communication with the host and initiates processes that control the input of messages to and output of messages from data communications lines.

lpm

Acronym for "lines per minute" used when describing the printing speed of a train printer.

LSP

See "Line Support Processor."

Magnetic Tape Data Link Processor (MT-DLP)

A processor that serves as the controller of a magnetic tape drive and provides the I/O interface between the system and the tape drive.

Maintenance card (MC)

A base module that acts as an interface between a maintenance test bus and a base.

Master Control Program (MCP)

The operating system on A Series and B 5000/B 6000/B 7000 Series systems: the program that controls the operational environment of the system. This control includes memory management, job selection, peripheral management, system utilization, program segmentation, subroutine linkage, and error logging.

Glossary

Master Electronic Control (MEC)

An interface between the Magnetic Tape Data Link Processor (MT-DLP) and the magnetic tape (MT) units that the MT-DLP supports.

MC

See "Maintenance card."

MCP

See "Master Control Program."

MEC

See "Master Electronic Control."

Message-Level Interface (MLI)

The interface between the host system and the input/output (I/O) and data communications subsystems.

Message-Level Interface Processor (MLIP)

The input/output (I/O) processor associated with a central processor unit.

Message-Level Interface Processor Destination Set (MLIPDS)

Contains the destination set of MLIPs that can be signaled for I/O initiation.

MLI

See "Message-Level Interface."

MLIP

See "Message-Level Interface Processor."

MLIPDS

See "Message-Level Interface Processor Destination Set."

MT-DLP

See "Magnetic Tape Data Link Processor."

NDLII

See "Network Definition Language II."

Network Definition Language II (NDLII)

The Burroughs language used to describe the physical, logical, and functional characteristics of the data communications subsystem on Message-Level Interface Processor (MLIP)/Host Data Unit (HDU) systems.

Network Support Processor (NSP)

On Message-Level Interface Processor (MLIP) systems, the data communications subsystem processor that controls the MLIP.

New Programming (NEWP) language

A structured, high-level programming language used in developing some of Burroughs system software.

NEWP

See "New Programming (NEWP) language."

NonReturn to Zero Data Link Processor (NRZ-DLP)

A processor that serves as the controller of a NonReturn to Zero (NRZ) tape drive and provides the I/O interface between the system and the NRZ tape drive.

NRZ-DLP

See "NonReturn to Zero Data Link Processor."

NSP

See "Network Support Processor."

ODT

See "Operator Display Terminal."

ODT-DLP

See "Operator Display Terminal Data Link Processor."

Operator Display Terminal (ODT)

The system console device that allows the operator to enter commands directly to the operating system to perform various functions.

Operator Display Terminal Data Link Processor (ODT-DLP)

A processor that serves as the controller of an Operator Display Terminal and provides the I/O interface between the system and the Operator Display Terminal.

Path Select Module (PSM)

The state of the CANDE input queue after the original executing command has finished. At that time, a CANDE "?GO", "?WAIT", "?PURGE", "?TAKE", or "?ENTER" command can be entered to manipulate the queue, or the queued input can be discarded by entering any other CANDE command.

PE-DLP

See "Phase-Encoding Data Link Processor."

Phase-Encoding Data Link Processor (PE-DLP)

A processor that serves as the controller of a Phase-Encoding (PE) tape drive and provides the I/O interface between the system and the PE tape drive.

PSM

See "Path Select Module."

Signal Processing Element Set (SPES) operator

An operator that signals an MLIP for I/O initiation.

SPES operator

See "Signal Processing Element Set (SPES) operator."

TP-DLP

See "Train Printer Data Link Processor."

Train Printer Data Link Processor (TR-DLP)

A processor that serves as the controller of a train printer and provides the I/O interface between the system and the printer.

UIO

See "Universal Input/Output."

unit

A peripheral device such as a disk drive.

Glossary

Unit Reserved (UR)

An Operator Display Terminal (ODT) command used to reserve a unit identified by a device name and unit number. The "-" option restores the unit identified by the device name and unit number.

Universal Input/Output

The input/output (I/O) subsystem that manages all transfers of information between the operating system and peripheral devices.

UR

See "Unit Reserve (UR)."

UR-

See "Unit Reserve (UR)."

WATI operator

WATI is a mnemonic for "read machine identification." The WATI operator identifies the system in use and determines which I/O method is to be used.

Index

Index

Index