MP-60 COMPUTER SYSTEMS

MPX/OS Input/Output Subsystem

EXTERNAL REFERENCE SPECIFICATION

Prepared by
Applications Systems Development Group
Systems Development Department

TABLE OF CONTENTS

1.0	INTRODUCTION	1-1
2.0	APPLICABLE DOCUMENTS	2-1
3.0	FEATURE DESCRIPTIONS	
3.1	DEVICE AND FILE MANAGEMENT	3-1
3.2	EVENT NOTIFICATION	3-2
3.3	DATA TRANSFER	3-20
3.4	DEVICE CONTROL	3-28
		3-33
4.0	PRODUCT-LEVEL DESCRIPTION	4-1
4.1	PUBLICATIONS AFFECTED	4-1
4.2	EQUIPMENT CONFIGURATION	4-1
4.3	INTERFACES TO OTHER SOFTWARE PRODUCTS	4-2
4.4	RESTRICTIONS AND LIMITATIONS	4-2
4.5	RELIABILITY, AVAILABLITY AND MAINTAINABILITY REQUIREMENTS	4-2
5.0	GLOSSARY	5-1
Appendix A	ESR AND DEVICE CROSS REFERENCE CHART	A-1
Appendix B	HARDWARE/DEVICE CODES	B-1
Appendix C	VALID HARDWARE TYPES	C-1
Appendix D	MPX/OS ERROR RECOVERY PROCEDURES	D-1

1.0 INTRODUCTION

The MPX Operating System, MPX/OS, Input/Output Subsystem includes all components required to perform the MPX/OS input and output Executive Service Requests (ESRs). These ESRs allow the MPX/OS user to perform physical data transfers, device control, and status checking. The ESRs processed by the MPX/OS I/O Subsystem are functionally divided into four (4) groups as follows:

Device and File Management
Event notification
Data Transfers
Device Control

The first group controls the access to all I/O devices and resolves device conflicts by delivering these resources to the requesting tasks on a priority basis. In order to generalize and facilitate a task's interface with the MPX/OS I/O Subsystem, a logical connection between the device and a task is created. This logical connection is called the logical unit (lu), and is a number between one and sixty-three (1-63). Using either the file or device assignment features, the user requests the MPX/OS I/O Subsystem to assign a device to a specified logical unit. Once the logical unit is assigned, the user may use the features in the other groups. This External Reference Specification describes the features contained in all four groups.

In the MPX/OS, device availability is divided into system and user. System available devices are normally shared by multiple users; therefore management and control must be performed by the MPX/OS. The user available devices are normally unit record equipment but may be any device not reserved for the system. Once a user available device is scheduled and assigned a job exclusively manages and controls the device. A system available device accessed by a user is refered to as a Logical Device, while a system available device accessed by the system or a user available device accessed by a user is refered to as a Physical Device. The MPX/OS Logical Devices supported by the I/O Subsystem are:

Dummy (a device with all ESRs avaliable but does nothing)
Files
Message Queues
Interactive Terminal
Remote Batch Terminals
Communications Network

The MPX/OS Physical Devices are installation dependent but normally include all peripheral equipment.

2.0 APPLICABLE DOCUMENTS

The following list of Control Data manuals document the MP-60 Emulation, the MPX Operating System and the MPX Product Set:

17329110	MP-32 Computer System MP-60 Emulation Reference Manual
17329100	MP-60 Computer System MPX/OS Reference Manual
10992100	MP-60 Computer System MPX/OS Operator's Guide
11110000	MP-60 Computer System Installation Handbook
14291700	MP-60 Computer System Program Command Console Reference Manual
14063800	MP-60 Computer System Utility Reference Manual
14062200	MP-60 Computer System PRELIB Reference Manual
14061300	MP-60 Computer System COMPASS Reference Manual
14061100	MP-60 Computer System FORTRAN Reference Manual
17328900	MASS/MPSIM Reference Manual

3.3 FEATURE DESCRIPTIONS

The MPX Operating System Input/Output Subsystem features are invoked by the execution of an MP-60 MON instruction. The MPX/OS routines, which are invoked by MON instruction, are called Executive Service Requests (ESRs). This section divides the MPX/OS I/O ESRs into four (4) groups. These groups and their associated ESRs are as follows:

Device and File Management
- ALLOCATE - reserve mass storage space.
- CLOSE ---- clear logical unit assignment.
- DEVICE --- assign logical unit to device.
- EXPAND --- increase mass storage space.
- MODIFY --- change mass storage attributes.
- OPEN ----- assign logical unit to file.
- RELEASE -- reduce mass storage space.
- SAVEQ ---- alter mass storage attributes.

Event Notification
- BUSY - check logical unit busy.
- UST -- obtain unit status.
- MUST - wait on multiple events.
- UTYP - obtain dynamic unit status.

Data Transfer
- FORMAT - initialize disk track.
- READLU - read from logical unit.
- READDS - alternate read from logical unit.
- WRITLU - write to logical unit.
- WRITDS - alternate write to logical unit.

Device Control
- BKSP --- backspace unit.
- CLEAR -- clear unit.
- ERASE -- erase tape segment.
- FUNC --- Function unit.
- REWD --- rewind unit.
- SELECT - select operating mode.
- SEOF --- search for end of file.
- UINT --- unsolicited interrupt.
- ULOC --- locate record on unit.
- UNLD --- unload unit.
- WEOF --- write end of file on unit.
- DIAG --- run diagnotstic test on unit.

The following sections describe the ESRs contained within each of the above groups.

3.1 DEVICE AND FILE MANAGEMENT

3.1.1 Abstract

The Device and File Management feature is invoked by control cards and Executive Service Requests (ESRs). The control cards and ESR are used to manage I/O resource allocation. The DEVICE and OPEN ESRs are used to establish the linkage between a task and the device, while the CLOSE ESR removes this linkage. The other Device and File Management ESRs allow the user to create, maintain, and remove files in the mass storage system.

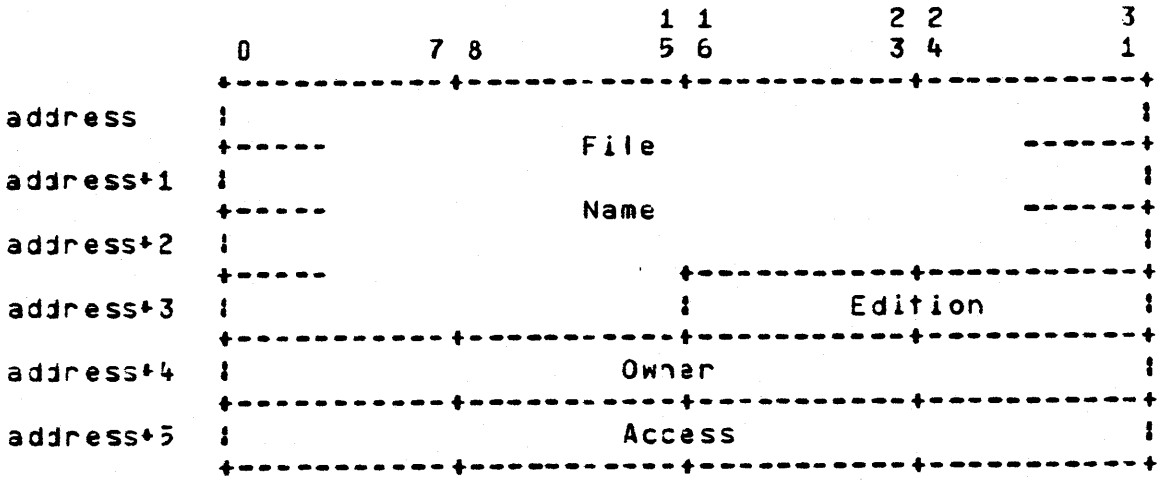
3.1.2 Description

The user invokes the Device and File Management features with the following ESRs:

ALLOCATE - reserve mass storage space.
CLOSE ---- clear logical unit assignment.
DEVICE --- assign logical unit to device.
EXPAND --- increase mass storage space.
MODIFY --- change mass storage attributes.
OPEN ----- assign logical unit to file.
RELEASE -- reduce mass storage space.
SAVEQ ---- alter mass storage attributes.

The Device and File Management ESRs used to manage the files on mass storage maintains a file directory on the system resident pack (SYSTEM01). This file directory contains information about the file necessary to uniquely identify and the locate the file in the mass storage system. The user must create an communications area in memory before invoking a file management ESR. The first six words of this area contain the information necessary to locate the file in the file directory. These first six words are refered to as the File Identification area and are described in Figure 3-1. The appropriate Device and File Management control card parameters are left-justified blank filled within the File Identification Area.

FILE IDENTIFICATION AREA



Where:

The File Name is fourteen (14) characters long.

The Edition is two characters long.

The Owner is four characters long.

The Access is a four character privacy access key.

The first five words must uniquely identify the file within the file system.

Figure 3-1

The ALLOCATE control card and ESR reserve space in the mass storage system and builds an entry in the file directory. Once a file is sucessfully created, the file will remain allocated until released (see RELEASE).

The CLOSE control card and ESR clear the logical unit assignment by removing the linkage between the task and the file or device. When the linkage for a file is removed, the CLOSE ESR updates information contained in the file directory and returns the File Identification in the caller's parameter area (PARM). All linkages established by a job are removed automatically when the job terminates.

The DEVICE and EQUIP control card, and the DEVICE ESR assign the specified logical unit to a Logical or Physical Device by establishing a linkage between a task and the device. The logical unit must be a number between 1 and 63. The EQUIP control card and the DEVICE ESR also allow the assignment of a logical unit to a previously assigned logical unit.

The EXPAND control card and EXPANDQ ESR are used to increase the mass storage space reserved for a file. Before the EXPAND control card or ESR can be invoked the user must have established a linkage to the file with the OPEN control card or ESR.

The MODIFY control card and ESR are used to alter the information contained in the file directory. This control card and ESR cannot be performed on an assigned file. If the file is assigned to this job then an error indicator is returned. If the file is assigned to another job then the caller may request waiting until the file is closed by the other job.

The OPEN control card and ESR assign the specified logical unit to a file; thus establishing a linkage between a task and the device. The logical unit must be a number between 1 and 63. The OPEN control card and ESR also allow the assignment of the file in an exclusive access mode; therefore the caller can request waiting until exclusive access may be established or a previous exclusive access is cleared by the CLOSE control card or ESR.

The RELEASE control card and ESR are used to remove some or all of the space reserved for a file. This contol card and ESR cannot be performed on a file which is assigned and the caller may request waiting until previous file assignments are cleared.

The SAVEPF control card and SAVEQ ESR are used to change the attributes of an assigned file. Before the control card or ESR can be invoked the user must have established a linkage with the OPEN control card or ESR.

3.1.3 Interfaces

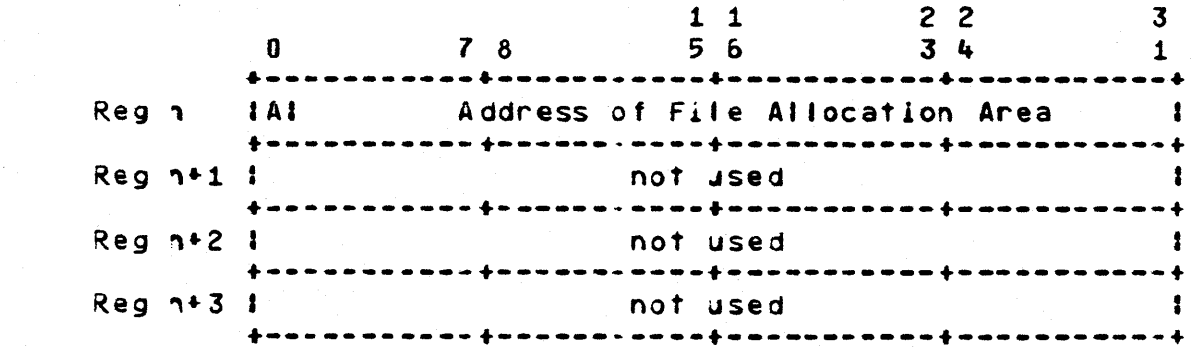
The format of the ALLOCATE control card call is as follows:

```
*ALLOCATE(fn,own,ed,ak,bsize,nblks,NS/S,use,dt,did1,...,did8)
```

where:

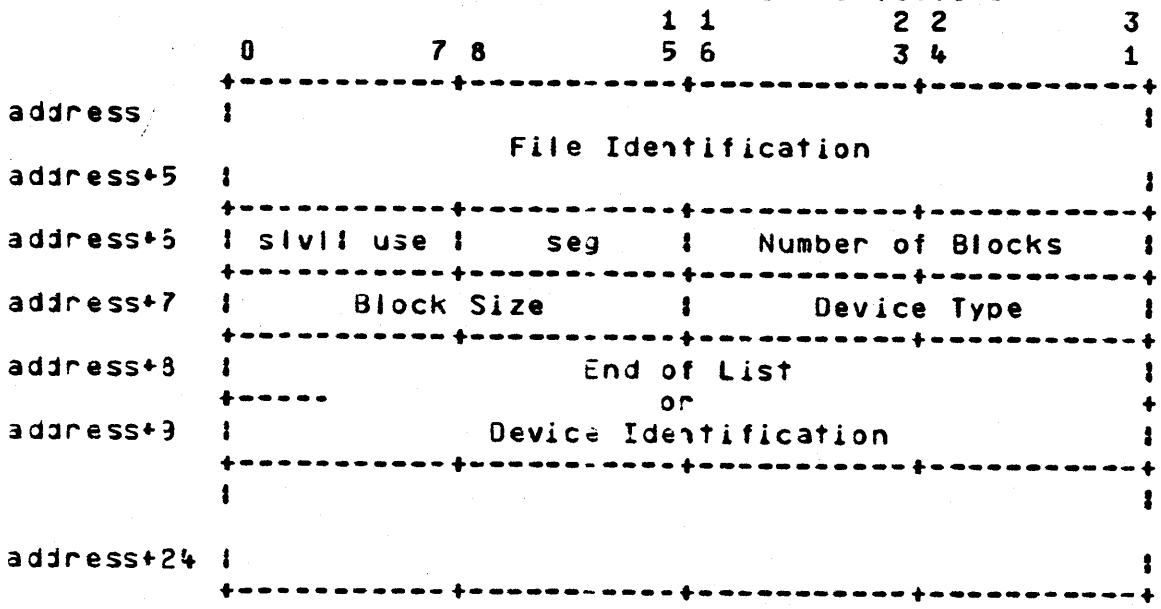
fn	is the file name (0-14 characters).
own	is the owner (0-4 characters).
ed	is the edition (0-2 characters).
ak	is the privacy access key (0-4 characters).
bsize	is the logical record size (1-4096).
nblks	is the number of blocks to reserve (1-65535).
NS/S	segmentation flag (NS--Segmentation not allowed).
use	is the allowable file usage.
dt	is the device type number.
did1	is the packname, upto eigth packnames are allowed.

The ALLOCATE ESR format is as follows:



A = 0 thread request if File Manager Busy.
= 1 return if File Manager Busy.

The format of the File Allocation Area is as follows:



An example of the ALLOCATE ESR calling sequence is as follows:

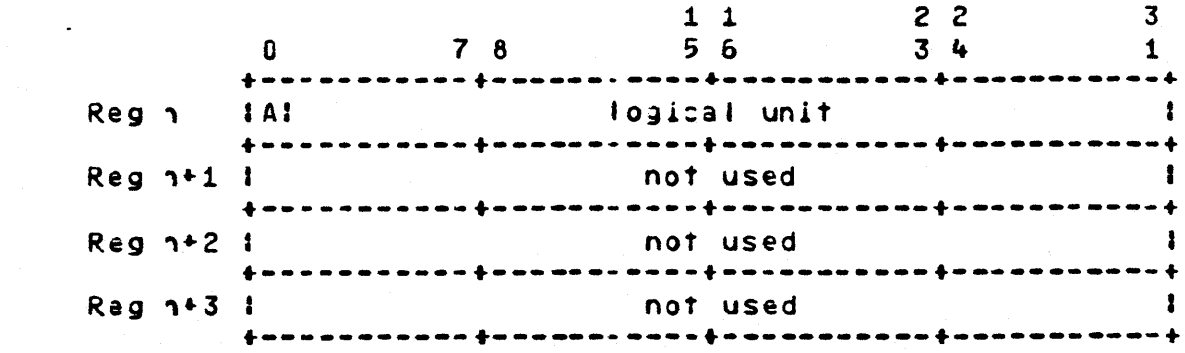
```
        EXT     ALLOCATE        Externally defined symbol
        LDA,R0  FAA             File Allocation Area address
        MON,R6  ALLOCATE        Monitor request


FAA     BSS     0
        TEXTC   14,SCRATCH      File Name
        TEXTC   2,01            Edition
        TEXTC   4,DDPG          Owner
        TEXTC   4,DNSS          Privacy access key
        VFD     4/0,4/0,8/0,16/100
        VFD     16/480,16/5
        TEXTC   8,SYSTEM01
        GEN     -1
```

The CLOSE control card format is as follows:

```
*CLOSE(lu1,...,lun)
```

Where the lu is the logical unit assigned to the file or device.

The CLOSE ESR format is as follows:



A = 0 tohread request if File Manager Busy.
= 1 return if File Manager Busy.

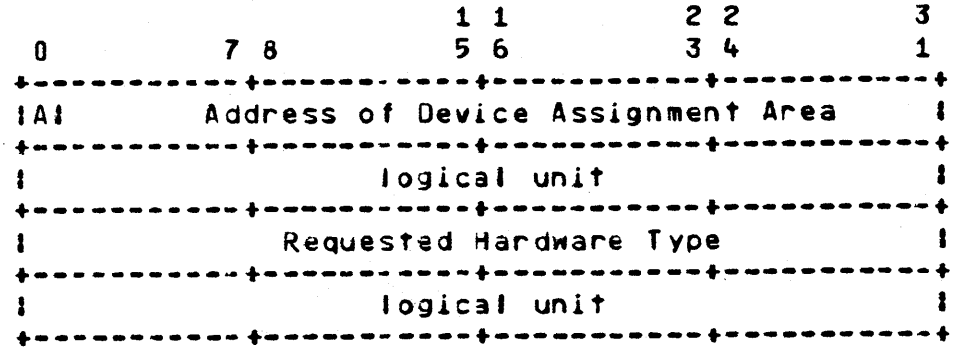
An example of the calling sequence is as follows:

```
        EXT     CLOSE           Externally defined symbol
        LDI,R0  15              Logical unit number
        MON,R0  CLOSE           Monitor request
```

The format of the EQUIP and DEVICE control cards are as follows:

```
*EQUIP(lu=d,...)
*EQUIP(lu1=lu2,...)
*DEVICE(lu=dev,p1,..,pn)
```

The DEVICE ESR format is as follows:



A = 0 thread request if File Manager busy.
= 1 return if File Manager busy.

An example of the calling sequence is as follows:

```
        EXT     DEVICE          Externally defined symbol
        LDA,R0  DAA             Device Assignment Area address
        LDA,R1  15              Logical unit number
        LDA,R2  2               Magnetic Tape
        MON,R0  DEVICE          Monitor request


DAA     BSS     0
        GEN     -1              Any unit.
```

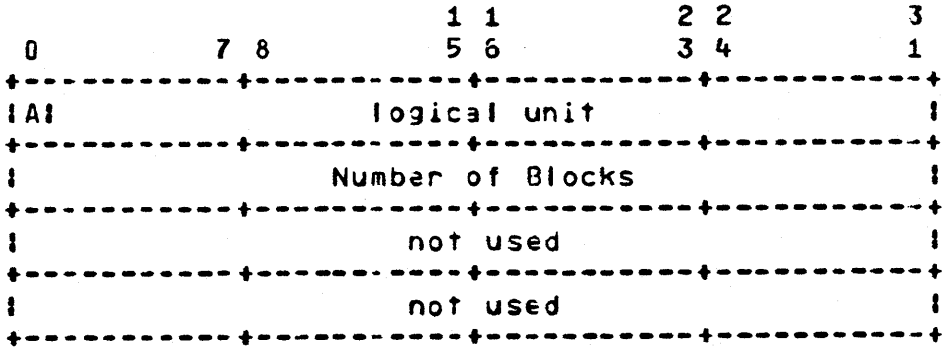
The format of the EXPAND Control Card is as follows:

```
*EXPAND(lu,NOBLKS)
```

where:

lu is the logical unit.
NOBLKS is the number of blocks added to the file.

The EXPAND ESR format is as follows:



A = 0 tnread request if File Manager busy.
= 1 return if File Manager busy.

The Number of Blocks is the number of logical blocks the file is to increase.

An example of the calling sequence is as follows:

```
EXT       EXPANDQ        Externally defined symbol
LDA,R0    15             logical unit number
LDA,R1    48             Number of blocks to add
MON,R0    EXPANDQ        Monitor request
```

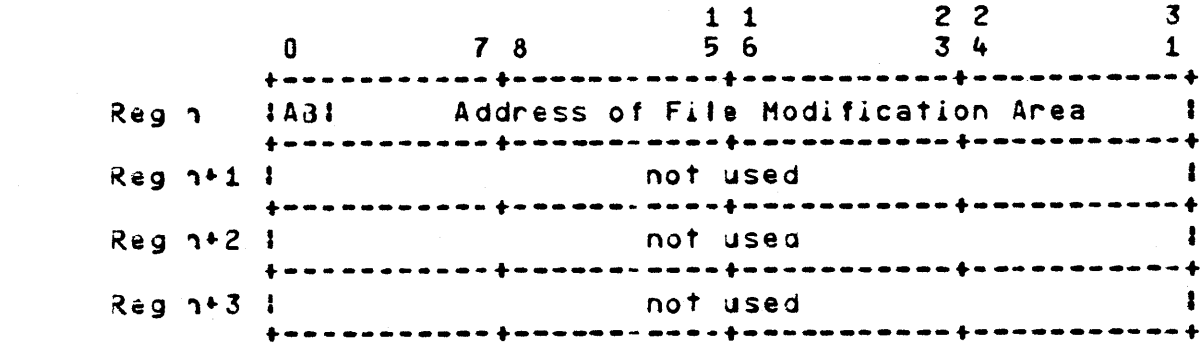
The format of the MODIFY control card is as follows:

```
*MODIFY(ofn,oown,oed,oak,nfn,nown,ned,nak,nblks,
        S/NS,use,dt,did1,...,did8)
```

where:

ofn	is the old the File Name string.
oown	is the old owner string.
oed	is the old edition string.
oak	is the old privacy access key.
nfn	is the new File Name string.
nown	is the new Owner Name string.
ned	is the new edition string.
nak	is the new privacy access key.
nblks	is the number of blocks to be added to the file.
NS/S	is the segmentation flag.
use	is the allowable file usage.
dt	is the device type number.
did1	is the device identification.

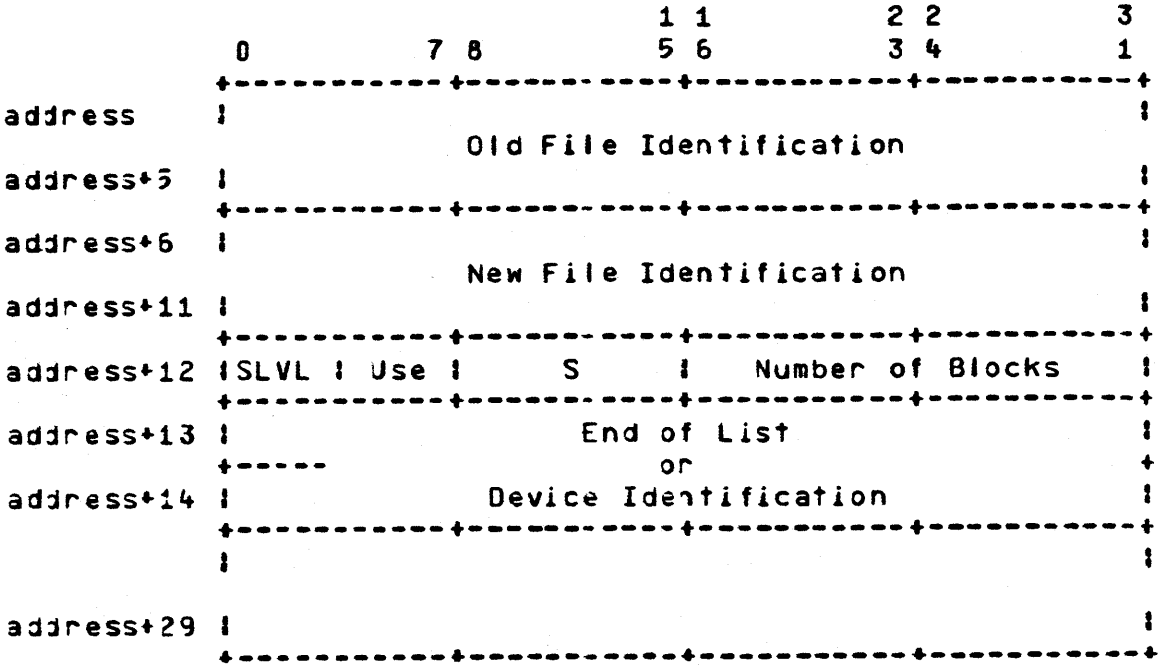
The MODIFY ESR format is as follows:



A = 0 thread request if File Manager busy.
= 1 return if File Manager busy.

B = 0 thread request if file is opened.
= 1 return if file is opened.

The format of the File Modification Area is as follows:



An example of the calling sequence is as follows:

```
          EXT       MODIFY          Externally defined symbol
          LDA,R0    FMA             File Modification Area address
          MON,R0    MODIFY          Monitor request


FMA       BSS       0
          TEXTC     14,TAPE10       Old File Name
          TEXTC     2,00                 Edition
          TEXTC     4,OOSC               Owner
          TEXTC     4,$$$$               Privacy access key
          TEXTC     14,SCRATCH      New File Name
          TEXTC     2,01                 Edition
          TEXTC     4,OOPG               Owner
          TEXTC     4,ONSS               Privacy access key
          VFD       4/0,4/0,8/0,16/100
          TEXTC     8,SYSTEM01
          GEN       -1
```

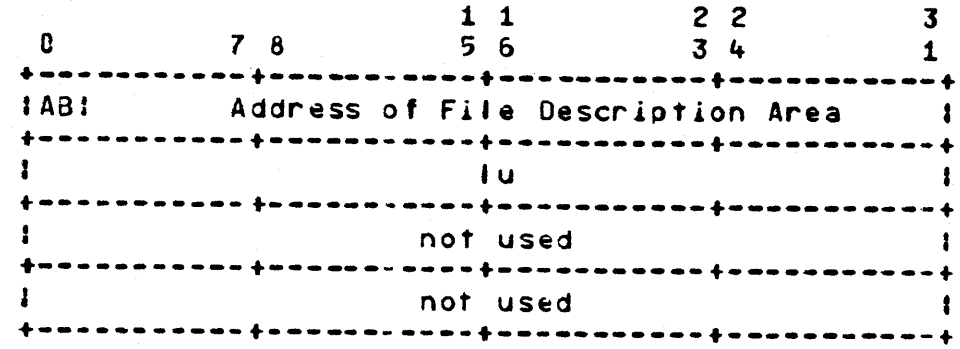
The format of the OPEN control card is as follows:

```
*OPEN(lu,fn,own,ed,ak,use,block)
```

where:

lu	is the logical unit number.
fn	is up to fourteen character string for the File Name.
own	is up to four character Owner NAME.
ed	is a two character edition.
ak	is up to four character privacy access key.
use	is the file usage (RW, read/write; W, write; R, read).
block	referenced block number for partial open.

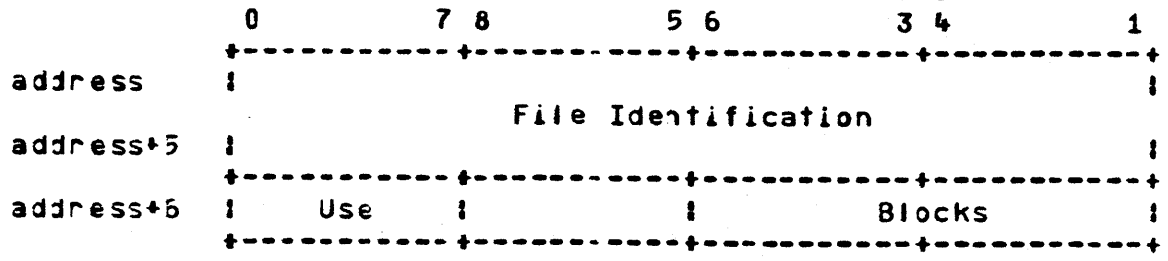
The OPEN ESR format is as follows:



A = 0 thread request if File Manager busy.
= 1 return if File Manager busy.

B = 0 thread request if file is opened.
= 1 return if file is opened.

The format of the File Description Area is as follows:



The Use field is a binary value which defines the use of the file:

=0 File to be used for read/write.
=1 File to be used for read only.
=2 File to be used for read/write, and set the highest block written to zero.

When the Block field is non-zero the file is opened where the value in the block field is the next block to be read/written.

An example of the calling sequence is as follows:

```
        EXT       OPEN              Externally defined symbol
        LDA,R0    FDA               File Decription Area address
        LDA,R1    15                Logical unit number
        MON,R0    OPEN              Monitor request


FDA     BSS       0
        TEXTC     14,SCRATCH        File Name
        TEXTC     2,01              Edition
        TEXTC     4,DDPG            Owner
        TEXTC     4,DNSS            Privacy access key
        VFD       4/0,4/0,8/0,16/0
```

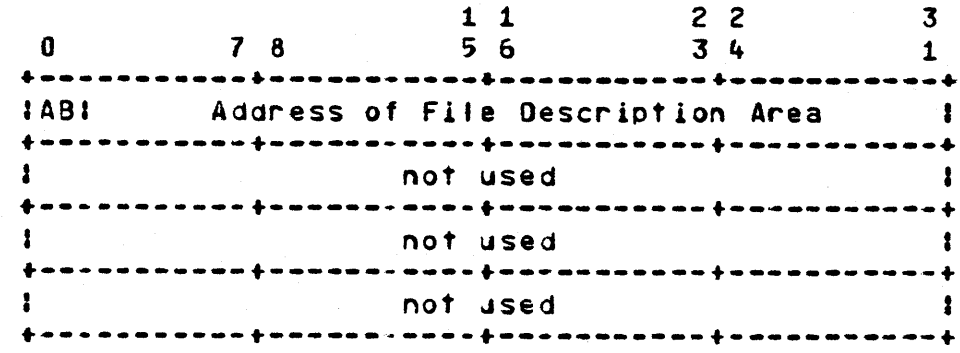
The format of the RELEASE control card is as follows:

```
*RELEASE(fn,own,ed,ak,nblks)
```

where:

fn	is up to fourteen character string for the File Name.
own	is up to four character Owner NAME.
ed	is a two character edition.
ak	is up to four character privacy access key.
nblks	is the number of blocks to release. A character of R means release unused. The character 0 means the entire file.

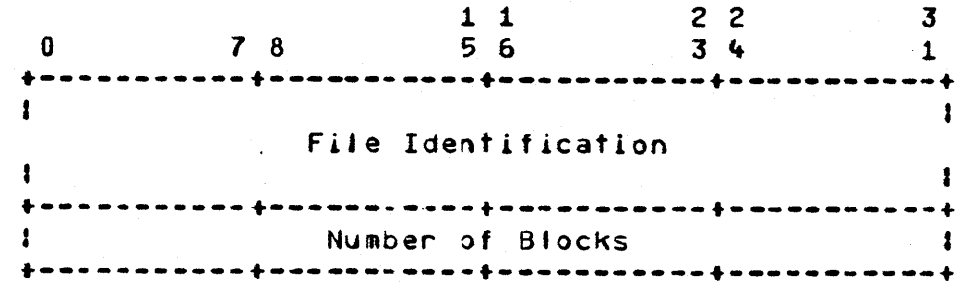
The RELEASE ESR format is as follows:



A = 0 thread request if File Manager busy.
= 1 return if File Manager busy.

B = 0 thread request if file is opened.
= 1 return if file is opened.

The format of the File Description Area is as follows:



The Number of Blocks is a binary number indicating the number of blocks to release. If the value is zero then the entire file is released, and if the value = -1 then the unused portion of the file is released.

An example of the calling sequence is as follows:

```
        EXT       RELEASE         Externally defined symbol
        LDA,R0    FDA             File Decription Area address
        MON,R0    RELEASE         Monitor request

FDA     BSS       0
        TEXTC     14,SCRATCH      File Name
        TEXTC     2,01            Edition
        TEXTC     4,DDPG          Owner
        TEXTC     4,DNSS          Privacy access key
        GEN       -1
```

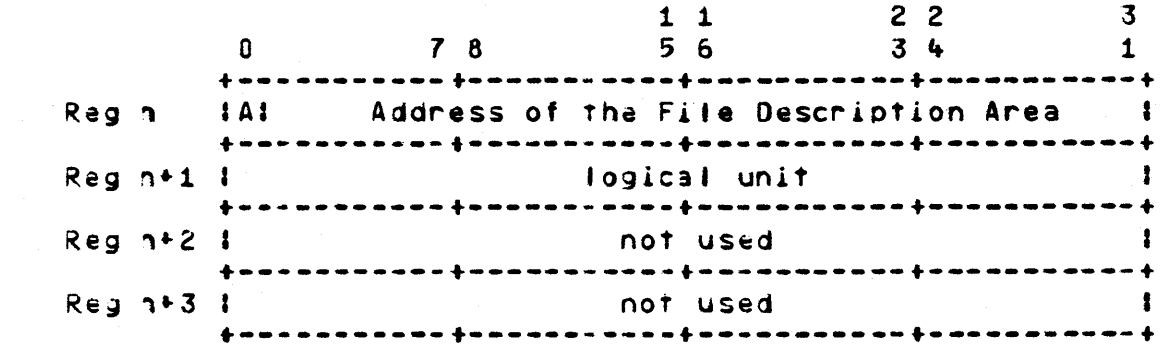
The format of the SAVEPF control card is as follows:

```
*SAVEPF(lu,fn,own,ed,ak,R)
```

where:

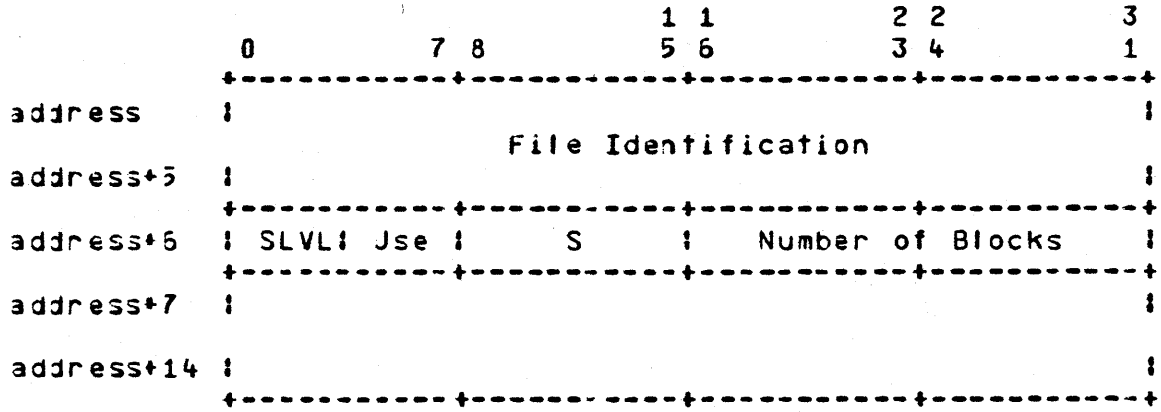
lu	is the logical unit number.
fn	is up to fourteen character string for the File Name.
own	is up to four character Owner NAME.
ed	is a two character edition.
ak	is up to four character privacy access key.
R	is the replacement flag.

The SAVEQ ESR format is as follows:



A = 0 thread request if File Manager busy.
= 1 return if File Manager busy.

The format of the File Description Area is as follows:



An example of the calling sequence is as follows:

```
          EXT       SAVEQ           Externally defined symbol
          LDA,R0    FDA             File Description Area address
          LDA,R1    15              Logical unit number
          MON,R0    SAVEQ           Monitor request

FDA       BSS       0
          TEXTC     14,SCRATCH      File Name
          TEXTC     2,01            Edition
          TEXTC     4,DDPG          Owner
          TEXTC     4,DNSS          Privacy access key
          GEN       -1
```

3.1.4 Aborts and Recovery

If the job termination is prevented from occurring while this ESR is in progress. If the system abnormally terminates then the worst case condition is the loss of usable disk space.

3.1.5 Errors

When a error in the call is detected, an error code is returned to the caller's parameter area, PARM. PARM must be defined as an external in the user's program. An error code of zero (0) specifies sucessful processing. The the File Manager error codes are:

1 File Manager Busy.
3 Incomplete parameter list.
4 No file name specified.
5 No block size specified.
6 No block count specified.
7 Illegal logical unit.
11 Label file read error.
12 File previously allocated.
13 Insufficient label file space.
14 Illegal device label.
15 Too many devices.
16 Insufficient contiguous space.
17 Insufficient space available on the specified devices.
18 File size exceeds system limits.
19 Block number exceed the number of blocks allocated.
20 File not allocated.
21 Operator cannot place devices on-line.
22 Device label read error.
23 Invalid logical unit.
24 Logical unit previously defined.
25 Incorract Read/Write permission.
26 File already opened with exclusive permission.
27 Insufficient table space.
28 Task has file open.
29 Illegal access key.
30 Too many device identifiers.
31 Label file cannot be closed.
32 Block size is 0.
33 Number of blocks is 0.
34 Segment count excceeds maximum.

3.1.6 Performance

These features require a number of disk references and should not be issued during time critical operations.

3.1.7 Installation Parameters

The default system device list is created during system initialization and reflects the entries in the mass storage tables which are installation dependent.

3.2 EVENT NOTIFICATION

3.2.1 Abstract

The Event Notification feature provides device and request status information, required by a user to manage an assigned logical unit.

3.2.2 Description

The Event Notification feature is invoked by the following ESRs:

BUSY - check logical unit busy.
UST -- obtain unit status.
MUST - wait on multiple events.
UTYP - obtain dynamic unit status.

These ESRs are legal on all devices and are described in the following paragraphs.

The BSY ESR returns the busy/not busy status of the specified logical unit. The status is not a function of a particular I/O request but rather of the logical unit itself. The requesting task is scheduled after the request is completed.

The UST ESR returns the status of the requesting task's last I/O request on the specified logical unit. If the I/O is still pending, the requesting task is placed in I/O wait until the I/O is completed.

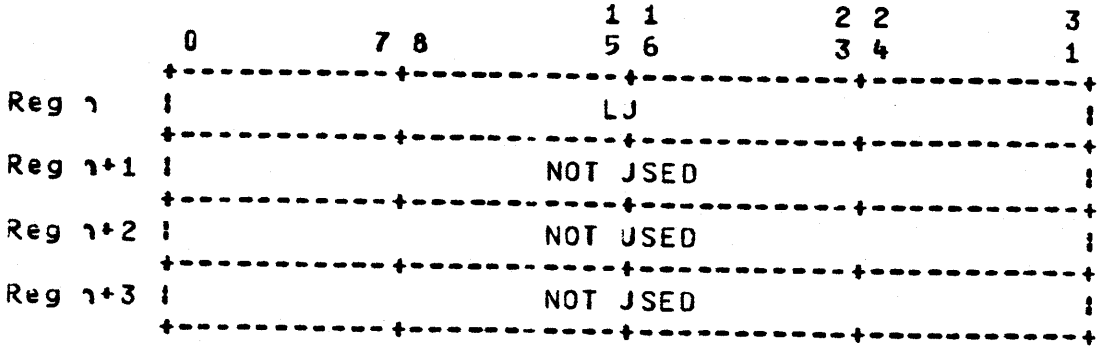
The MUST ESR allows the requesting task to be placed in a wait state pending the occurance of the specified event(s). The requesting task supplies a 128 bit mask where bits 1-63 correspond with the end of operation on logical units 1-63 respectively. The other bit (0, 64-127) correspond to the user defined events (DEFEVNTQ ESR).

The UTYP ESR returns the hardware type of the specified logical unit. If the hardware type is a disk file, additional file description information is also returned. The requesting task is scheduled for execution after the request is completed.

The STATUS ESR is called by a task to issue a status command to a specified logical unit. If the device is busy, the request is queued and the task is put into I/O wait. If the device is not busy, the current status is requested from the device and the task is put into I/O wait and not scheduled for execution until the operation is complete.

3.2.3 Interfaces

The BSY ESR format is as follows:

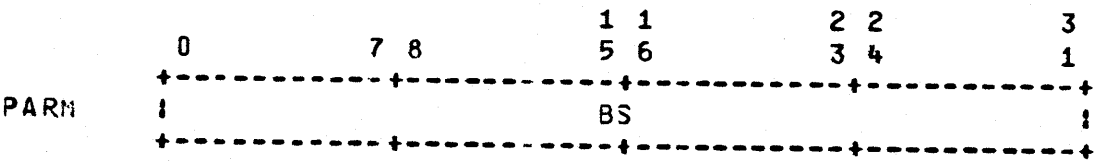


LJ Logical unit for which busy status is to be returned.

An example of a calling sequence is as follows:

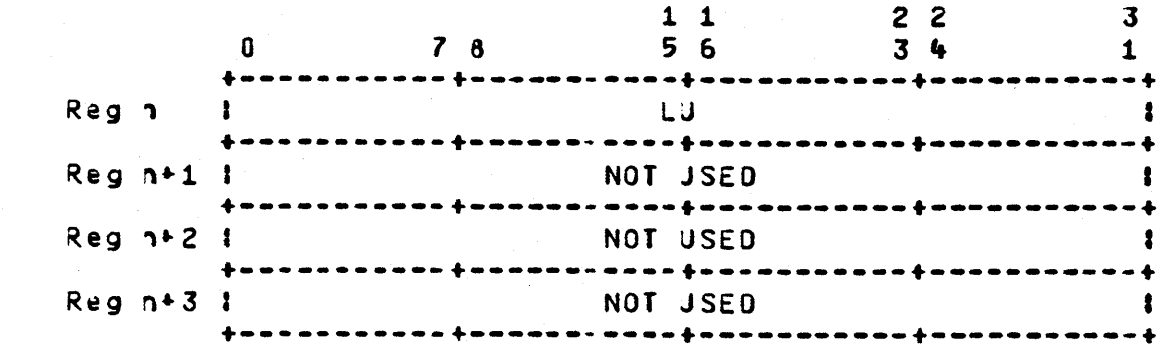
```
        EXT     BSY             Externally defined symbol
        LDI,R0  10              Logical unit number
        MON,R0  BSY             Monitor request
```

The parameters returned in PARM are as follows:



BS = Zero (0), unit is not busy.
= non-zero, unit is busy.

The UST ESR format is as follows:

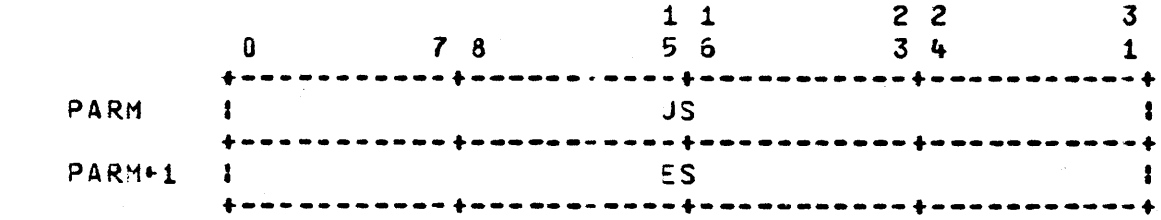


LU Logical unit for which status is to be returned.

An example of a calling sequence is as follows:

```
        EXT     UST             Externally defined symbol
        LUI,R0  10              Logical unit number
        MON,R0  UST             Monitor request
```

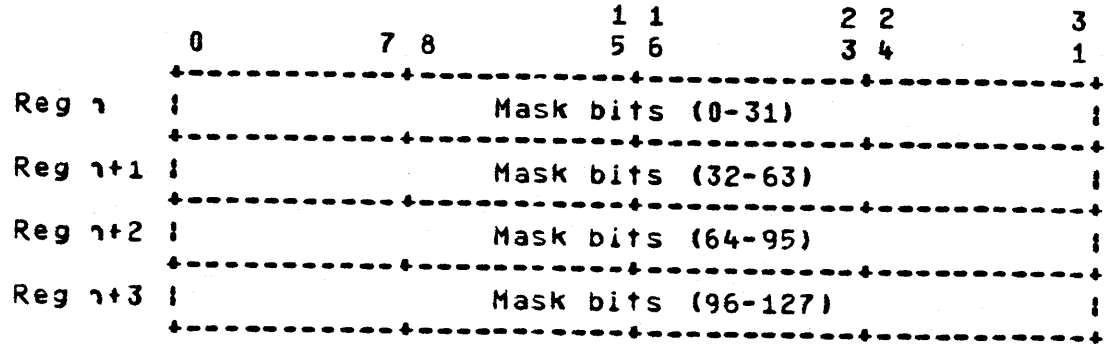
The parameters returned in PARM are as follows:



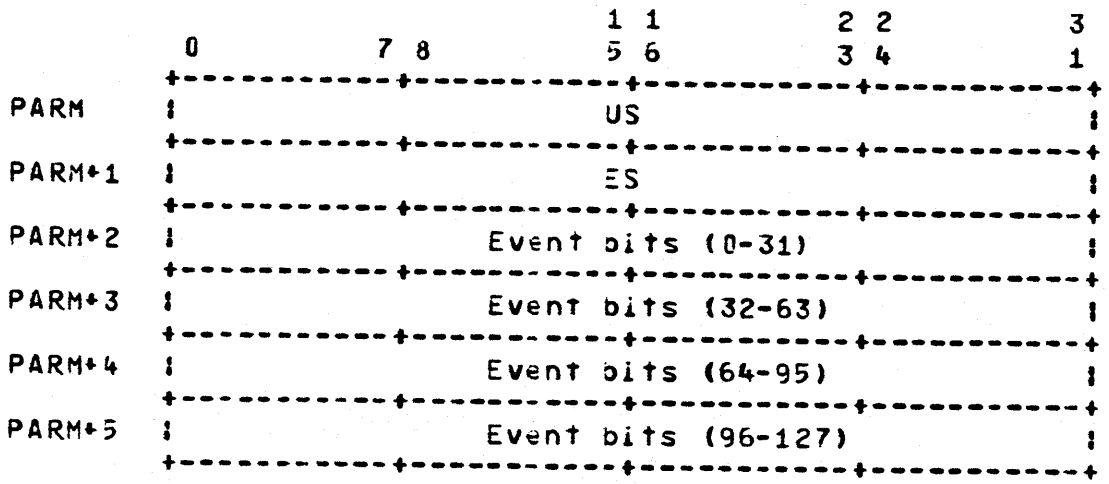
US Normal unit status as described in Appendix B.

ES Expanded status as described in Appendix B.

The MUST ESR format is as follows:



The parameters returned in PARM are as follows:



US Unit status as described in Appendix B.

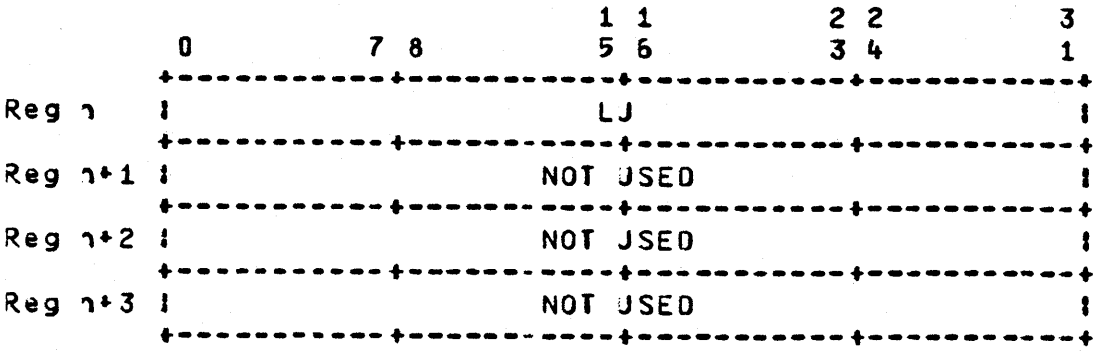
ES Expanded status as described in Appendix B.

The Event bits (0-127) are set to a one for each event which has occurred. The US and ES parameters are for the lowest numbered logical unit with an event bit set.

An example of a calling sequence is as follows:

```
EXT       MUST          Externally defined symbol
LDD,R0    MASK1         Event Mask Bits
LDD,R2    MASK3         Event Mask Bits
MON,R0    MUST          Monitor Request
```

The UTYP ESR format is as follows:

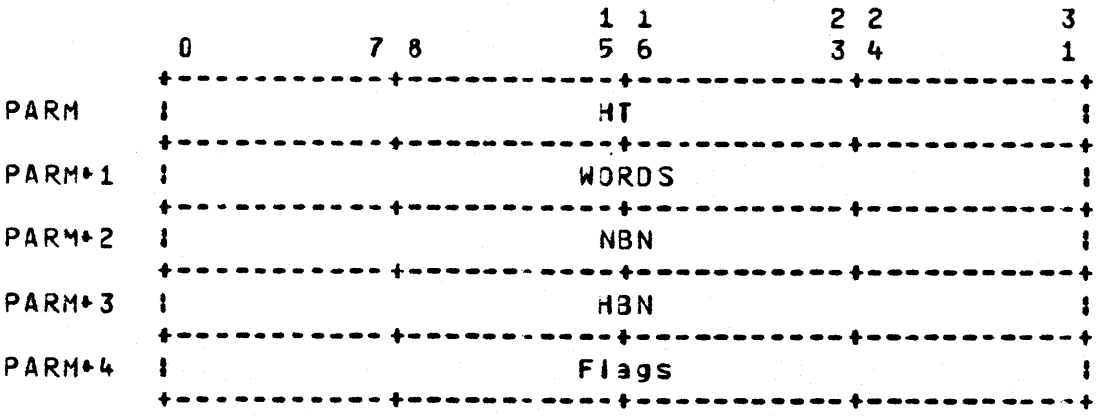


LU Logical unit for which hardware type is to be returned.

An example of a calling sequence is as follows:

```
        EXT       JTYP          Externally defined symbol
        LDI,R0    10            Logical unit number
        MON,R0    UTYP          Monitor request
```

The parameters returned in PARM are as follows:



HT Hardware type, valid hardware types are defined in Appendix C.

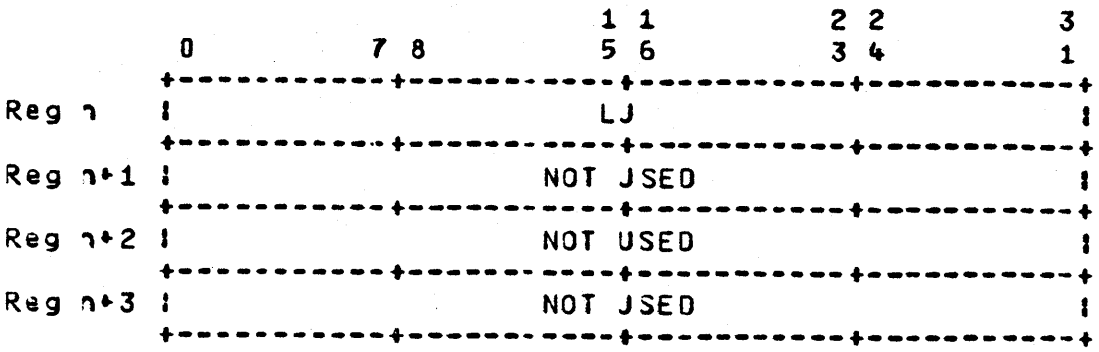
WORDS Number of words per block, returned for HT = 1 only

NBN Number of next block to be read or written (current block number), returned for HT = 1 only

HBN Highest block number written (end-of-file), returned for HT = 1 only

Flags
- Bit 0, if set the device is a disk file
- Bit 1, if set the device is a magnetic tape
- Bit 2, if set the device is a blocked device
- Bit 3, if set the device is an input-only device
- Bit 4, if set the device is an output-only device
- Bit 5, if set the device is an ASCII-only output device
- Bit 6, if set the device is an interactive terminal
- Bit 7, if set the device is a remote batch terminal
- Bit 8, if set the device is a communication network

The STATJS ESR format is as follows:

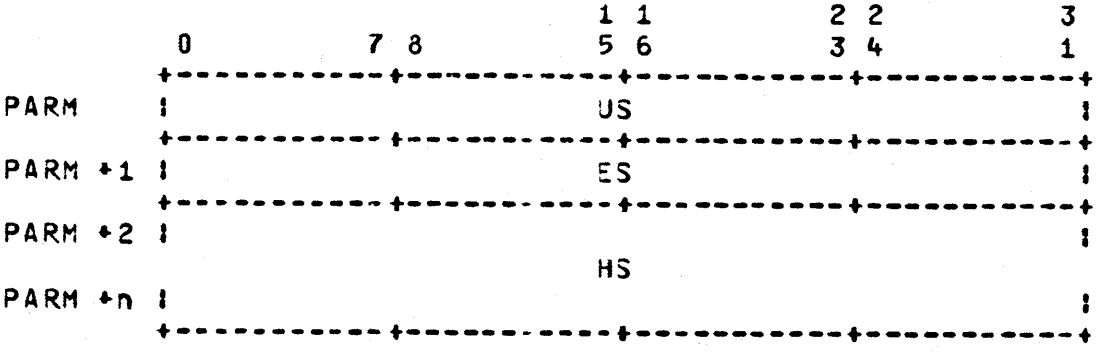


LJ Logical unit

An example of a calling sequence is as follows:

```
        EXT       STATUS          Externally defined symbol
        LDI,R0    10              Logical unit number
        MON,R0    STATUS          Monitor request
```

The parameters returned in PARM are as follows:



JS Normal unit status as described in Appendix B.

ES Expanded status, as described in Appendix B.

HS Hardware status as described in Appendix B.

3.2.4 Aborts and Recovery

The following two abort conditions are possible with these ESRs:

TYPE	ABORT CODE	
1	3	Logical unit not assigned to a device or file.
1	8	Logical unit value not between 1 and 63.

3.2.5 Errors

Not applicable.

3.2.6 Performance

The BSY, UST, MUST, and STATUS ESRs are used to synchronize I/O request processing with task processing and could result in the task being placed in an I/O wait status. Use of the BSY ESR may degrade system performance and the MUST ESR could be used more efficiently in place of the BSY ESR.

3.2.7 Installation Parameters

The values returned in PARM by the UTYP ESR may be modified during system generation. In addition, the peripheral equipment configuration is defined during system generation.

3.3 DATA TRANSFER

3.3.1 Abstract

The ability to transfer data between a task's memory buffer and a device is supplied by this feature.

3.3.2 Description

The Data Transfer feature is invoked by the following ESRs:

FORMAT - initialize disk track.
READLU - read from logical unit.
READDS - alternate read from logical unit.
WRITLU - write to logical unit.
WRITDS - alternate write to logical unit.

These ESR are legal on the devices as specified in Appendix A. The following paragraphs describe the Data Transfer ESRs.

The FORMAT ESR writes the track addresses and timing marks necessary for subsequent data storage on a disk pack. The requesting task is scheduled for execution after the request is initiated, and must issue a BSY, JST, or MUST ESR to determine when the request is completed.

The READDS ESR initiates a data transfer from a specified logical unit to a buffer residing in the requesting task's memory. The requesting task is scheduled for execution after the request is initiated, and must issue a BSY, UST, or MUST ESR to determine when the request is completed.

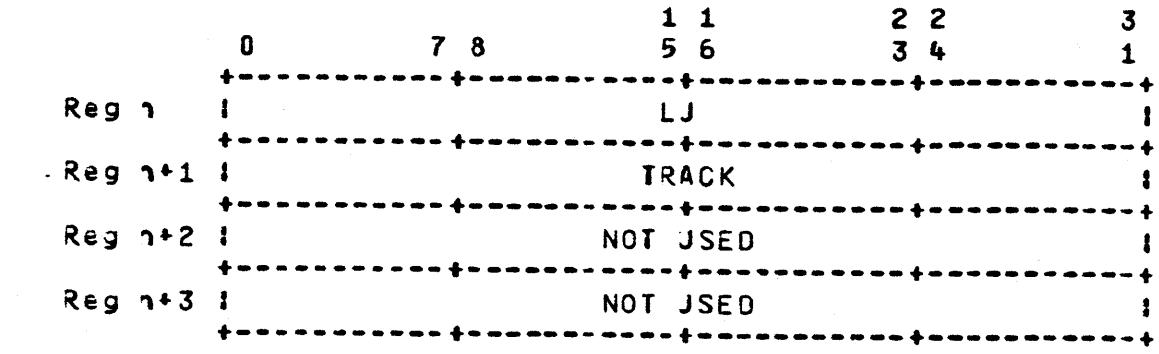
The READLU ESR initiates a data transfer from a specified logical unit to a buffer residing in the requesting task's memory. The requesting task is scheduled for execution after the request is initiated, and must issue a BSY, UST, or MUST ESR to determine when the request is completed.

The WRITDS ESR initiates a data transfer to a specified logical unit from a buffer residing in the requesting task's memory. The requesting task is scheduled for execution after the request is initiated, and must issue a BSY, JST, or MUST ESR to determine when the operation is completed.

The WRITLU ESR initiates a data transfer to a specified logical unit from a buffer residing in the requesting task's memory. The requesting task is scheduled for execution after the request is initiated, and must issue a BSY, UST, or MUST ESR to determine when the operation is completed.

3.3.3 Interfaces

The FORMAT ESR format is as follows:



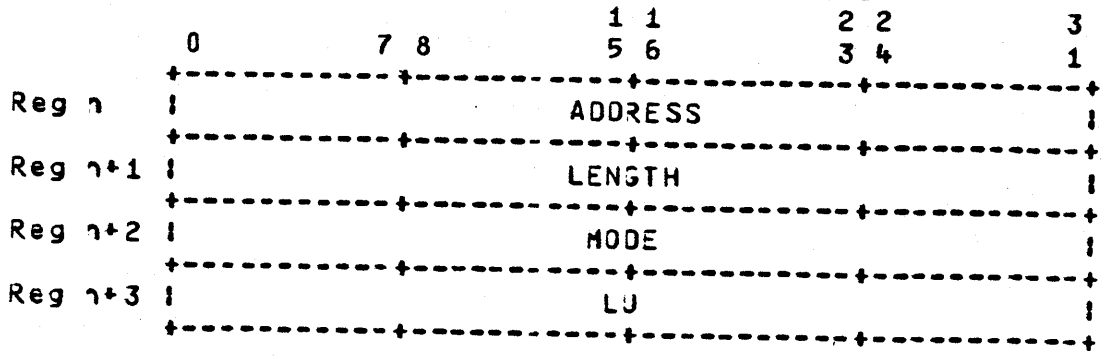
LJ Logical unit

TRACK Track number to format

An example of a calling sequence is as follows:

```
        EXT       FORMAT          Externally defined symbol
        LDI,R4    47              Logical unit number
        LDI,R5    500             Track number
        MON,R4    FORMAT          Monitor request
```

The READDS and READLU ESR formats are as follows:



ADDRESS The address of the first element (word or byte) of the buffer is determined by the MODE parameter.

LENGTH The number of words (bytes) to be transfered. LENGTH values may be from 0 to 4096 words (0 to 16,384 bytes). A value of zero (0) is treated as the maximum LENGTH value of 4096 words (16,384 bytes).

MODE The data Transmission Mode (format) code:

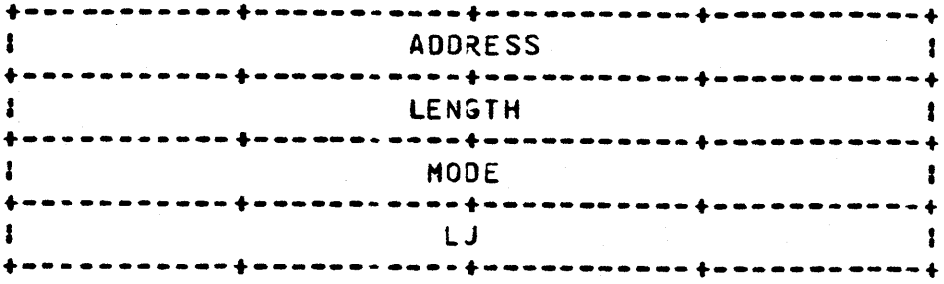
0 ASCII record, word format
16 ASCII record, byte format
32 Binary record, word format

LU Logical unit to be read.

An example of the calling sequence is as follows:

```
EXT       READDS      Externally defined symbol
LDCA,R0   BUFA        Byte address of buffer
LDI,R1    48          Number of bytes
LDI,R2    16          ASCII records in byte format
LDI,R3    15          Logical unit number
MON,R0    READDS      Monitor request
```

The WRITDS and WRITLU ESR formats are as follows:



ADDRESS The address of the first element (word or byte) of the buffer is determined by the MODE parameter.

LENGTH The number of words (bytes) to be transfered. LENGTH values may be from 0 to 4096 words (0 to 16,384 bytes). A value of zero (0) is treated as the maximum LENGTH value of 4096 words (16,384 bytes).

MODE The data Transmission Mode (format) code:

0 ASCII record, word format
16 ASCII record, byte format
32 Binary record, word format

LJ Logical unit to be read.

An example of the calling sequence is as follows:

```
        EXT     WRITDS      Externally defined symbol
        LDCA,R0 BUFA        Byte address of buffer
        LDI,R1  48          Number of bytes
        LDI,R2  16          ASCII records in byte format
        LDI,R3  15          Logical unit number
        MON,R0  WRITDS      Monitor request
```

3.3.4 Aborts and Recovery

The following abort conditions are possible with these ESRs:

ABORT TYPE	ABORT CODE	
1	1	Operator rejected request to ready a unit.
1	2	Buffer size larger than 4096 words.
1	3	Logical unit unassigned.
1	4	Attempt to write on read-only file.
1	5	An input was attempted into a read-only page.
1	6	Hardware reject.
1	7	An input or output was attempted upon a protected page.
1	8	Illegal logical unit number.
1	9	Command is not legal for assigned device.

3.3.5 Errors

Hardware and data transmission error recovery procedures are error code and device dependent. The MPX/OS error recovery procedures are described in Appendix D.

3.3.6 Performance

Since the data transfer features are scheduled for device manager processing in priority order, a lower priority request could wait for a higher priority request to complete.

3.3.7 Installation Parameters

Not applicable.

3.4 DEVICE CONTROL

3.4.1 Abstract

The ability to control a device is provided by the Device Control features.

3.4.2 Description

The Device Control Feature is invoked by the following ESRs:

BKSP --- backspace unit.
CLEAR -- clear unit.
ERASE -- erase tape segment.
FUNC --- Function unit.
REWD --- rewind unit.
SELECT - select operating mode.
SEOF --- search for end of file.
UINT --- unsolicited interrupt.
ULOC --- locate record on unit.
UNLD --- unload unit.
WEOF --- write end of file on unit.
DIAG --- run diagnotstic test on unit.

These ESRs are legal on the devices as specified in Appendix A, and the following paragraphs describe these ESRs.

The BKSP ESR positions the logical unit before the preceeding physical record or block unless the logical unit is already at the beginning of the tape or file. The requesting task is scheduled for execution after the request is initiated and must issue a BSY, UST, or MUST ESR to determine when the request is completed.

The CLEAR ESR is called by a task to clear out the current I/O operation for a specified device. The command or data transfer is terminated at the point of receipt of the clear command. The task is put into I/O wait and not scheduled for execution until an end-of-operation interrupt is recieved . If the device is not busy, the clear request results in no action and the task is scheduled for immediate execution.

The ERASE ESR erases approximately 6 inches of magnetic tape in an effort to bypass faulty material. The requesting task is scheduled for execution after the request is initiated, and must issue a BSY, UST, or MUST ESR to determine when the request is completed.

The FUNC ESR is used to send device dependant functions to a logical unit. The requesting task is scheduled for execution after the request is initiated, and must issue a BSY, UST, or MUST ESR to determine when the request is completed.

The REWD ESR repositions a logical unit to the beginning-of-tape (BOT) for magnetic tape devices and to the first block for mass storage devices. The requesting task is scheduled for execution after the request is initiated, and must issue a BSY, UST, or MUST ESR to determine when the request is completed.

The SELECT ESR is used to set the operating modes for a logical unit. The available mode selections for each device are specified in Appendix B.

The SEOF ESR initiates a search operation (forward or backward) on a specified logical unit for the next file marker, initial point of the file for backward searches, or end of the file for forward searches. The requesting task is scheduled for execution after the request is initiated, and must issue a BSY, UST, or MUST ESR to determine when the request is completed.

The UINT ESR allows the requesting task to be notified when an unsolicited interrupt occurs on a logical unit. The unsolicited interrupt could be used to signal the occurance of a malfunction.

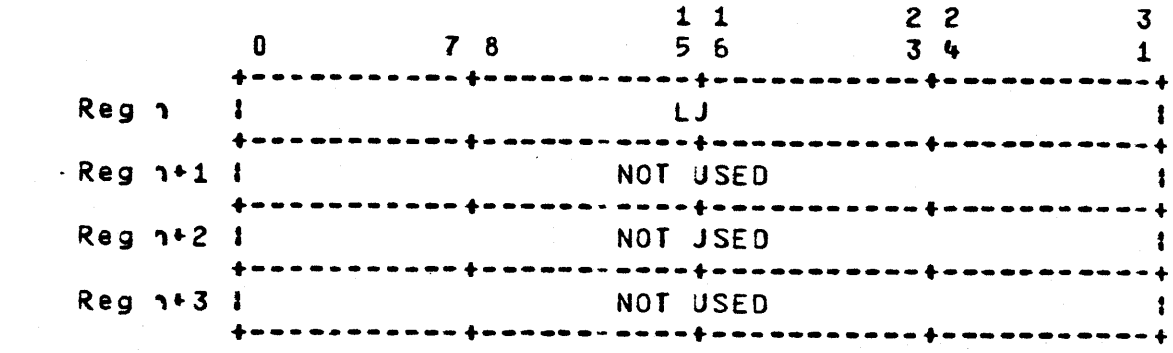
The ULOC ESR sets the next block number of a logical unit to a requested block. If the requested block number is greater than the allocated area, the next block number is set to the last block written +1. The requesting task is scheduled for execution after the request is completed.

The UNLD ESR rewinds and makes not ready a magnetic tape reel. The requesting task is scheduled for execution after the request is initiated, and must issue a BSY, UST, or MUST ESR to determine when the request is completed. Following an unload of a tape the tape drive is still assigned to the job and the logical unit.

The WEOF ESR causes an end-of-file mark to be written on the specified logical unit (magnetic tape), or sets the last block written value to the current block number (disk file). The requesting task is scheduled for execution after the request is initiated and must issue a BSY, UST, or MUST ESR to determine when the request is completed.

3.4.3 Interfaces

The BKSP ESR format is as follows:

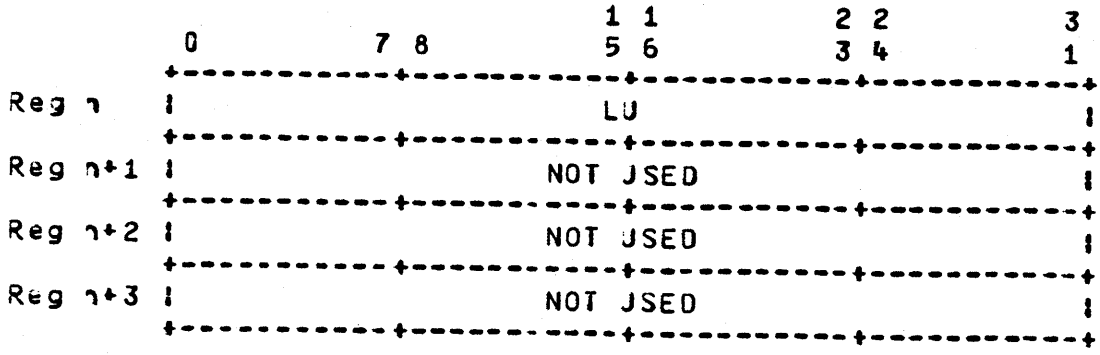


LJ Logical unit to be backspaced.

An example of a calling sequence is as follows:

```
        EXT     BKSP            Externally defined symbol
        LDI,R0  10              Logical unit number
        MON,R0  BKSP            Monitor request
```

The CLEAR ESR format is as follows:

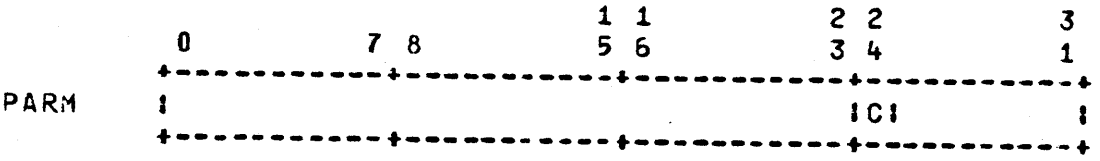


LU Logical unit

An example of a calling sequence is as follows:

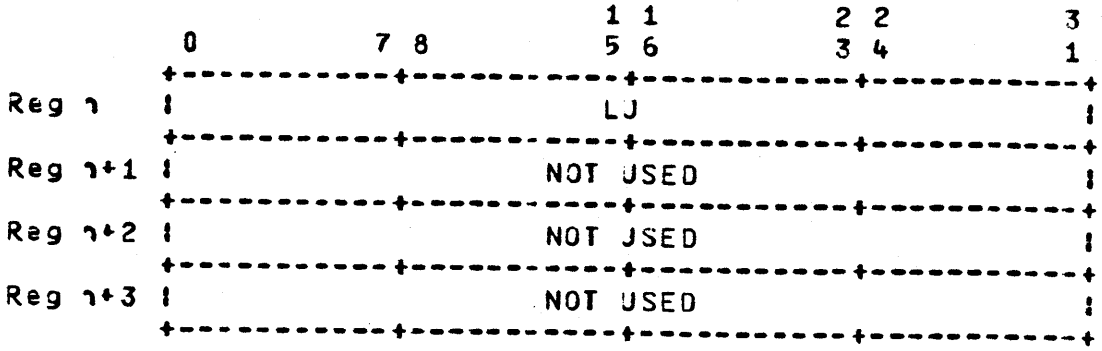
```
        EXT       CLEAR          Externally defined symbol
        LDI,R0    20             Logical unit number
        MON,R0    CLEAR          Monitor request
```

The parameters returned in PARM are as follows:



C Clear status, if set an operation in progress was cleared, otherwise the device was not busy.

The ERASE ESR format is as follows:

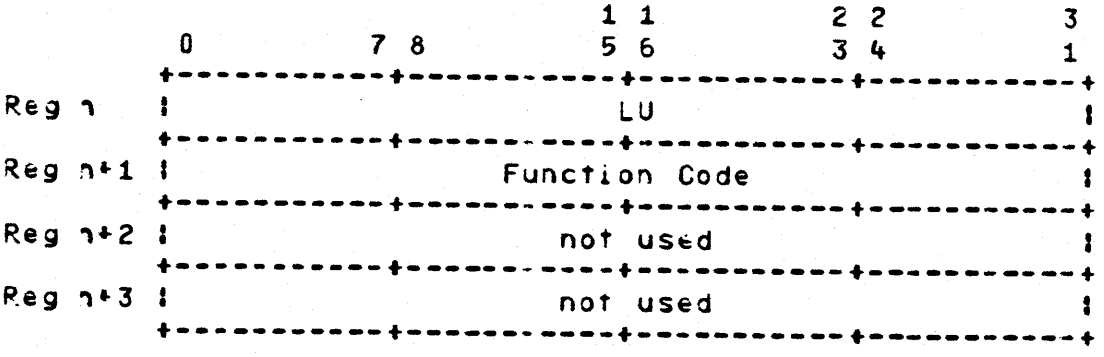


LU Logical unit to be erased.

An example of a calling sequence is as follows:

```
        EXT     ERASE           Externally defined symbol
        LDI,R0  10              Logical unit number
        MON,R0  ERASE           Monitor request
```

The FUNC ESR format is as follows:



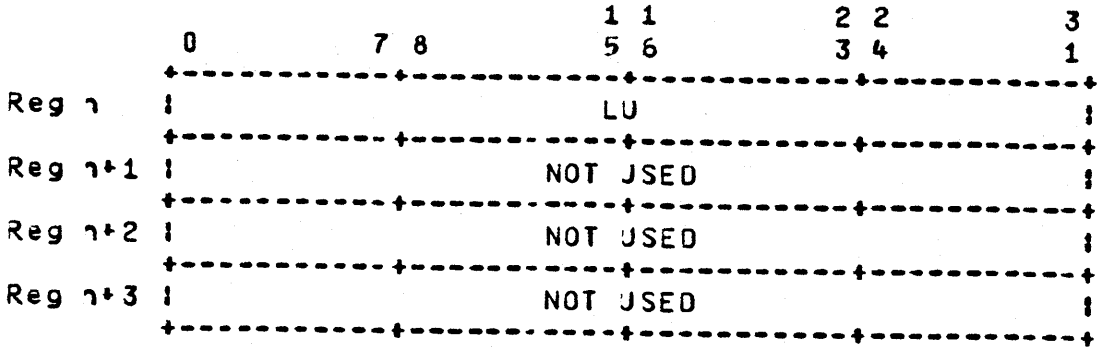
LU Logical unit

Function Codes as defined in Appendix B.

An example of a calling sequence is as follows:

```
        EXT     FUNC            Externally defined symbol
        LDA,R0  ADDR            Buffer starting address
        LDI,R1  2               Buffer length
        LDI,R2  1               Function code
        LDI,R3  10              Logical unit number
        MON,R0  FUNC            Monitor request
```

The REWD ESR format is as follows:

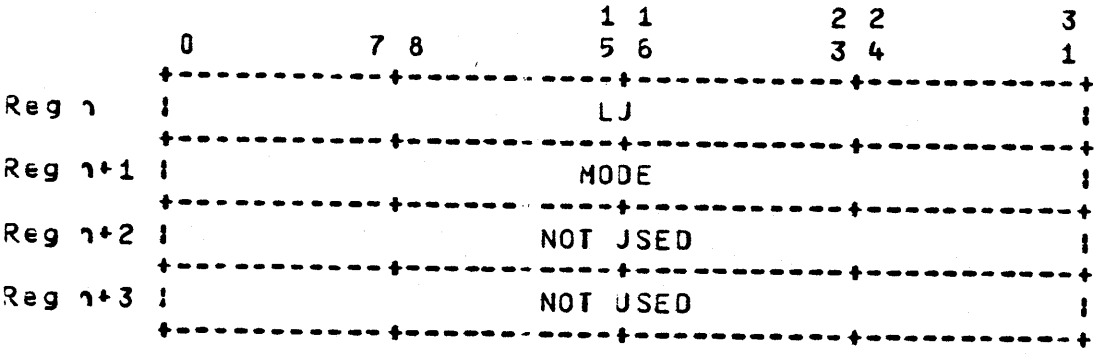


LU Logical unit to be positioned.

An example of a calling sequence is as follows:

```
        EXT     REWD            Externally defined symbol
        LDI,RB  21              Logical unit number
        MON,RB  REWD            Monitor request
```

The SELECT ESR format is as follows:



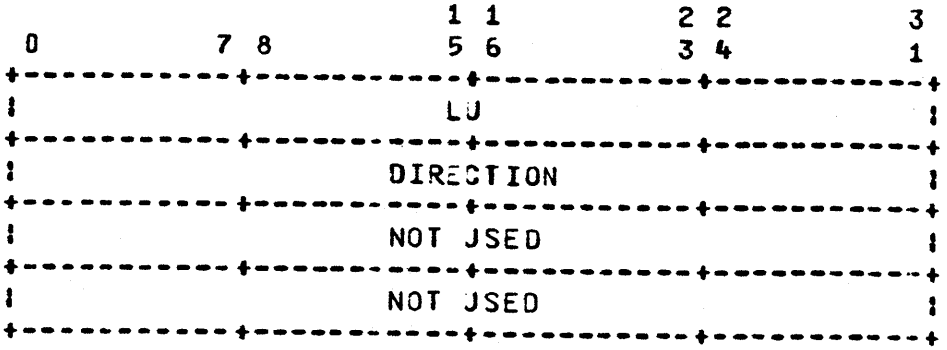
LU Logical unit

MODE The modes for specific devices are listed in Appendix B.

An example of a calling sequence is as follows:

```
        EXT     SELECT          Externally defined symbol
        LDI,R0  12              Logical unit number
        LDI,R1  1               Select mode
        MON,R0  SELECT          Monitor request
```

The SEOF ESR format is as follows:



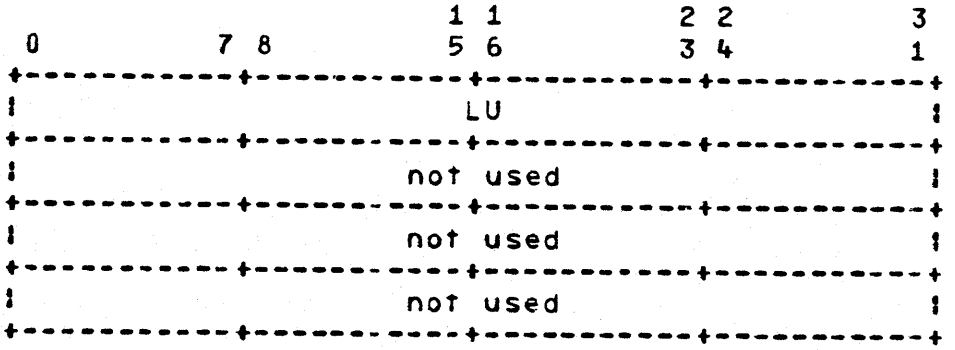
LU Logical unit to be searched for End-of-File mark.

DIRECTION Direction of the search
= 0, search forward
= 1, search backward

An example of a calling sequence is as follows:

```
        EXT     SEOF            Externally defined symbol
        LDI,R0  10              Logical unit number
        LDI,R1  0               Search forward
        MON,R0  SEOF            Monitor request
```

The JINT ESR format is as follows:

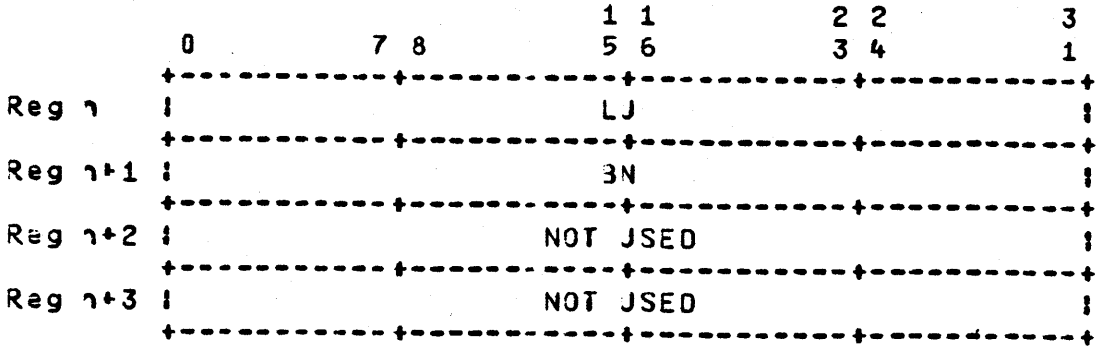


LU Logical Unit assigned to the monitored device.

An example of the calling sequence is as follows:

```
        EXT     JINT            Externally defined symbol
        LDI,R0  15              Logical unit number
        MON,R0  JINT            Monitor request
```

The ULOC ESR format is as follows:



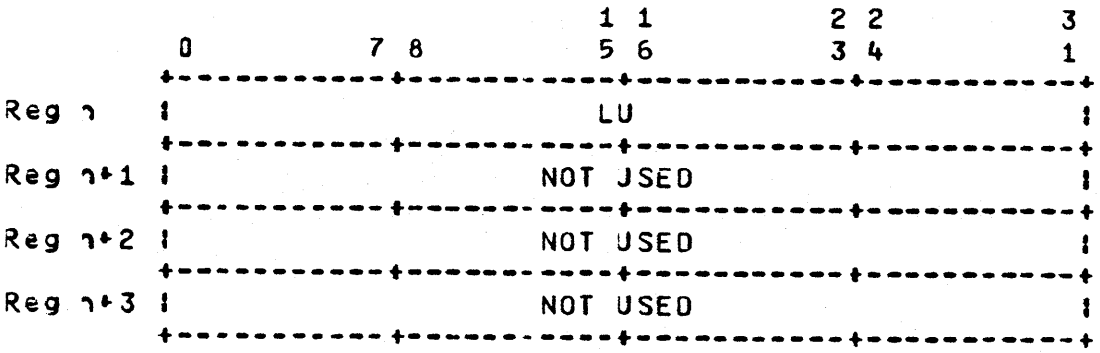
LU Logical unit to be positioned.

BN Block number to which the unit is to be set. This value will be the next block read or written. If the value = -1 then the last block written +1 will will be used.

An example of a calling sequence is as follows:

```
        EXT     JLOC            Externally defined symbol
        LDI,R3  10              Logical unit number
        LDI,R4  24              Block number
        MON,R3  JLOC            Monitor request
```

The UNLD ESR format is as follows:

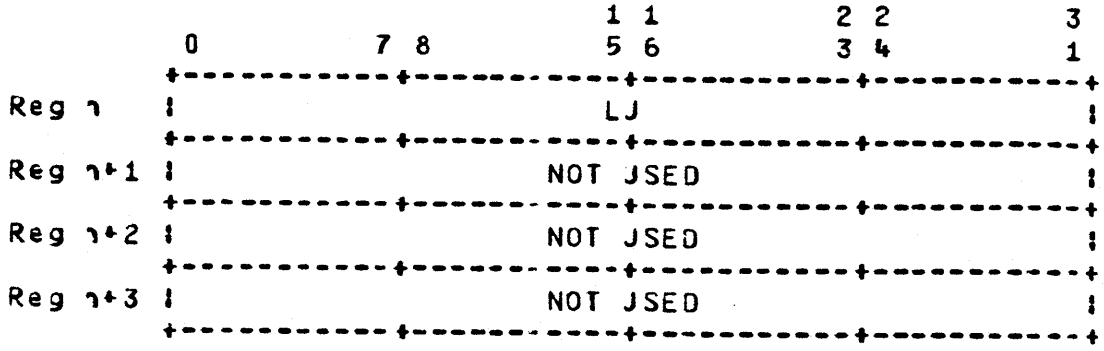


LU Logical unit to be unloaded.

An example of a calling sequence is as follows:

```
        EXT     UNLD            Externally defined symbol
        LDI,R0  10              Logical unit number
        MON,R0  UNLD            Monitor request
```

The WEOF ESR format is as follows:

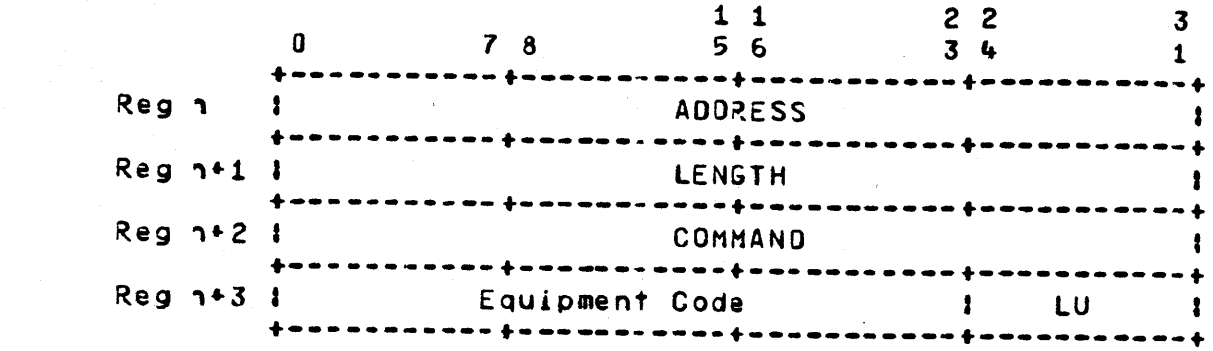


LJ Logical unit End-of-File mark is to be written upon.

An example of a calling sequence is as follows:

```
        EXT     WEOF            Externally defined symbol
        LDI,R0  10              Logical unit number
        MON,R0  WEOF            Monitor request
```

The DIAG ESR format is as follows:



ADDRESS Buffer first word address

LENGTH Buffer length

COMMAND Commands are device dependent and are listed for the legal devices in Appendix C.

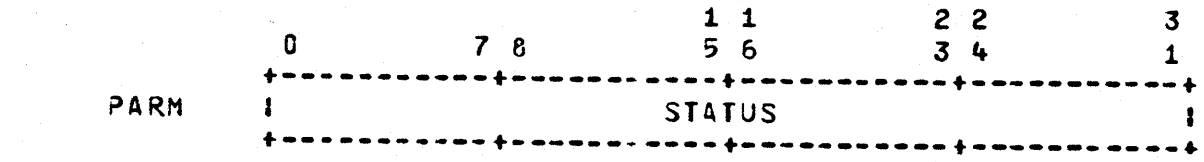
Equipment code is the hardware equipment identification.

LU Logical unit

An example of a calling sequence is as follows:

```
        EXT     DIAG            Externally defined symbol
        LDA,R0  BFWA            Buffer first word address
        LDI,R1  48              Buffer length
        LDI,R2  CMND            Command
        LD,R3   EQUP            Equipment code
        LDI,R3  20,R3           Logical unit number
        MON,R0  DIAG            Monitor request
```

The parameters returned in PARM are as follows:



STATUS
0 Diagnostic started
1 Device does not exist
2 Device assigned
3 LU assigned
4 No table space available

3.4.4 Aborts and Recovery

The following abort conditions are possible with these ESRs:

ABORT TYPE	ABORT CODE	
1	1	Operator rejected request to ready a unit .
1	2	Buffer size larger than 4096 words.
1	3	Logical unit unassigned.
1	4	Attempt to write on read-only file.
1	5	An input was attempted into a read-only page.
1	6	Hardware reject.
1	7	An input or output was attempted upon a protected page.
1	8	Illegal logical unit number.
1	9	Command is not legal for assigned device.

3.4.5 Errors

Hardware and data transmission error recovery procedures are error code and device dependent. The MPX/OS error recovery procedures are described in Appendix D.

3.4.6 Performance

Since the data transfer features are scheduled for device manager processing in priority order, a lower priority request could wait for a higher priority request to complete.

3.4.7 Installation Parameters

Not applicable.

4.0 PRODUCT-LEVEL DESCRIPTION

4.1 PUBLICATIONS AFFECTED

Not all manuals in the following list of Control Data documents describing the MP-60 Emulation, the MPX Operating System and the MPX Product Set are affected by this External Reference Specification.

41618300	MP-32 Computer System MP-60 Emulation Reference Manual
10817300	MP-60 Computer System MPX/OS Reference Manual
10992100	MP-60 Computer System MPX/OS Operator's Guide
11110000	MP-60 Computer System Installation Handbook
14291700	MP-60 Computer System Program Command Console Reference Manual
	MP-60 Computer System Text Editor Reference Manual
14063800	MP-60 Computer System Utility Reference Manual
14062200	MP-60 Computer System PRELIB Reference Manual
14061300	MP-60 Computer System COMPASS Reference Manual
14061100	MP-60 Computer System FORTRAN Reference Manual

4.2 EQUIPMENT CONFIGURATION

Equipment configurations required to support the MPX Operating System Input/Output Subsystem are as follows:

	MINIMUM	TARGET	MAXIMUM
Number of CPUs	1	1	8
Program states	6	6	48
Memory (32-bit words)	98,304	131,072	4,194,304
Card reader	1	1	1
Line printer	1	1	2
Display console	1	1	1
Interactive terminals	0	48	256
Mass storage (megabytes)	10	400	1500
Magnetic tapes	0	2	8

4.3 INTERFACES TO OTHER SOFTWARE PRODUCTS

The MPX Operating System Product Set members use the Executive Service Requests described in section 3. Their exists an exentsive interface between the MPX/OS I/O Subsystem and the Interactive Terminal Subsystem (ITS). This interface is described in the preliminary I/O Subsystem Internal Maintenance Specification.

4.4 RESTRICTIONS AND LIMITATIONS

The MPX Operating System Input/Output Subsystem is intended for the MP-32 architecture, but the design will be flexible enough to operate on the MPP. The MPX/OS I/O Subsystem, on either machine, should support software developed using the existing Input/Output modules.

4.5 RELIABILITY, AVAILABLITY AND MAINTAINABILITY REQUIREMENTS

The I/O Subsystem is an integral part of the MPX Operating System and meets the reliability, availability, and maintainability requirements of the MPX Operating System.

5.0 GLOSSARY

Abort	The premature termination of a process whenever an irrecoverable situation occurs.
Absolute	Refers to actual machine address.
ASCII	American Standard Code for Information Interchange. The standard code, using a coded character set consisting of 7-bit coded characters. See American National Standard X3.4-1977.
Assemble	The process by which an object (binary) module is created from the symbolic language program.
Asynchronous	Refers to a type of serial transmission in which bit synchronization is accomplished for each character.
Batch	Class of tasks which run on a time available basis.
BCLA	Buffered Communications Line Adapter is the controller interface to the RS-232C asynchronous communications lines.
Bit	A binary digit.
Block	A group of machine words or bytes. Usually a collection of one or more records used in I/O to reduce the number of physical operations.
Buffer	A portion of memory used to collect data in order to compensate for speed differences between the processor and peripheral devices.
Byte	Eight (8) contiguous bits.
CALL	The tranfer of control to a closed routine or task. A Executive Service Request, CALL, is used to activate a specific task.

Callee	The task initiated by the CALL Executive Service Request.
Caller	The task initiating another routine or task.
CLA	Communication Line Adapter, the hardware interface between the CPU and the communications line.
Compile	The process by which an assembly module is created from a problem solving language. A compiler usually generates several machine instructions from a single symbolic statement.
Common	An area of memory that may be shared between tasks or routines. Tasks may communicate through common areas.
Control Point Zero	CYBER System Control Point.
CPU	Central Processing Unit.
CRC	Cyclic Redundancy Check.
Data	An area of memory that may be prestored with information at load time and may be shared between routines of a task.
Dispatcher	An operating system routine that unthreads a task from the top of the ready list and places the task into execution.
ESR	Executive Service Request, name given to the MPX Operating System routines invoked by the MON instruction.
File	A collection of blocks and/or records, usually of related data. Each mass storage file has an entry in the system Label File.
Interrupt	A break in the normal processing flow usually caused by a hardware signal. Interrupts can be enabled or disabled and occur with an associated priority. Processes that are interrupted are later resumed at the point of interruption.

ITS	Interactive Terminal Subsystem.
IOC	Input/Output Controller.
Job	The sequential and/or parallel execution of tasks. Begins with a *JOB card and ends with a *EOJ card.
JCT	Job Control Table is an area of memory containing information controlling a given job.
Job Manager	A system task that processes the input stream of the job. The Job Manager is a set of reentrant routines shared by all user jobs.
Library	A collection of frequently used, checked-out programs maintained on an external device that can be loaded and executed separately or in conjuction with a user's program. Libraries must be arranged to minimize searching.
Linkage	The interconnection between routines or devices. The loader matches externals and entry points to establish program linkage. The Interactive Terminal Subsystem match job terminal requests with terminal user connection requests to establish port linkage.
Loader	A system task that is used to load, relocate, and link binary object modules.
LU	Logical Unit, a number representing the user connection to a device.
MPX/OS	The MPX Operating System.
MP-32	MP-60 CPU Emulation device.
MPR	MP-60 CPU Emulation device (Militarized, or Ruggidized).

Ordinal	The relative location of an entry in a table. The absolute location of an entry can be obtained by multiplying the ordinal by the number of machine words per entry and adding the starting address of the table.
Page	A 4096 word block of contiguous memory. Paging is a technique where a logical address is transformed via a set of page registers into a physical address.
PMAC	Programmable Multiple Access Controller.
Port	The communications line between the CPU and the user.
PPU	CDC CYBER/6000 Peripheral Processor Unit.
Process	A software program equivalent to the MPX/OS terms tasks and job.
Priority	A value assigned to an item in the system which facilitiates scheduling and processing within the operating system.
Queue	A list used to control the processing to be done.
Ready list	A prioritized list of tasks waiting for control of the CPU.
Relocatable	Refers to a program that has been prepared by a source language compiler or assembler to be loaded into any area of available memory.
Resident	The portion of the operating system which resides permanently in memory.
RETURN	A Executive Service Request that terminates a task and transfers control to the point in the caller where the call originated. A task may return with or without release of memory.
Schedule	The determination of which processing is required next.

Stack	A last-in, first-out queue.
Status	A stage or condition of an I/O request or a task.
Synchronous	Serial transmission in which characters are sent bitwise without start and stop bits.
System Initialize	Refers to the intial system load process where the resident is loaded, memory initialized, and the system tasks are started.
Task	An independent unit of work that can compete for the resources of the system. A task may call and be called by other tasks.
TCT	Task Control Table is an area of memory containing information used to control a task.
Terminal	The device connected to the user end of the PORT.
Terminate	The process of completing a job. A job may terminate normally or abnormally.
Thread	A linked list of elements, the contents of each thread cell contains the address of the next thread cell and so on until a thread cell contains a value indicating the end of the list.
Utility	A routine, procedure or program that supports the operation of a system.

Appendix A ESR AND DEVICE CROSS REFERENCE CHART

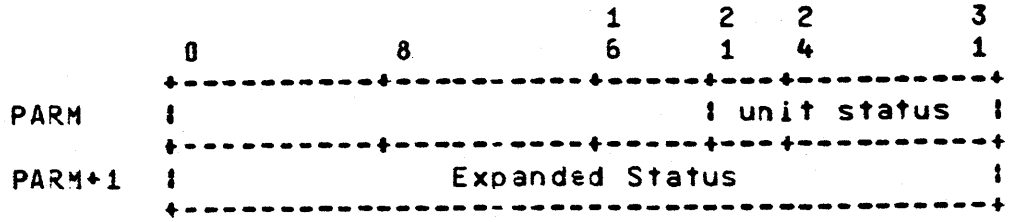
The following chart cross references ESRs and devices; ESRs not in the chart are legal on all devices.

	FORMAT	READLU	READDS	WRITLU	WRITDS	BKSP	ERASE	FUNC	REWD	SELECT	SEOF	ULOC	UNLD	WEOF	DIAG
Logical Devices															
Dummy	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
Message Queues	I					I	I		I		I		I		N
Files	I		I		I		I	I		I			I		N
Interactive	I		I		I	I	I	I	I		I	I	I	I	N
Remote Batch	I		I		I	I	I	I	I		I	I	I	I	N
Com. Network	I				I	I	I	I	I	I	I		I	I	N
Physical Devices															
1867 SMD/MMD						I	I	I	I	I	I		I	I	N
1833-5 FDD			I		I		I	I		I	I		I		N
1829 CARD READER	I		I	I	I	N	I	I	N		I	I	I	I	N
1827 PRINTER	I	I	I		I	N	I	I	N		I	I	I		N
1860-4 MAG TAPE	I		I		I			I		N		I			N
2558-3 COUPLER	I						I				I		I		N
LOCAL CRT	I	I	I	I	I	I	I		I	I	I	I	I	I	N
BCLA/MUX	I	I	I	I	I	I	I		I	I	I	I	I	I	N
MPCLA	I	I		I		I	I		I		I		I	I	N

I = ILLEGAL   N = NULL   blank = LEGAL

Appendix B HARDWARE/DEVICE CODES

The information returned in PARM for the UST, MUST, and STATUS ESRs is as follows:



The unit status for each device is as follows:

Dummy

No error bits or status used.

Files

31 Reject -- always set when an error occurs. When no other error bits are set and this bit is set then the device is busy.

30 Not Ready -- when this bit and bit 31 are set, the mass storage device is in a not ready condition.

29 End of File -- when this bit and bit 31 are set, an attempt to read beyond the highest block written was attempted.

28 Transmission Mode -- After a data transfer this bit is one.

27 End of Device -- When this bit and bit 31 are set a transfer beyond the highest available sector was attempted.

26 End of Allocated Blocks -- When this bit and bit 31 are set a write beyond the highest allocated block was attempted and the file requires expansion.

25 Data Error -- When this bit and bit 31 are set a data error was encountered.

24 Hardware Error -- When this bit and bit 31 are set a hardware condition was encountered.

23 Lost Data -- When this bit and bit 31 are set a lost data condition was encountered.

22 Address Error -- When this bit and bit 31 are set a sector address error was encountered.

21 Seek Error -- When this bit and bit 31 are set a seek error was encountered.

Queues

31	Reject -- always set when an error occurs. When no other error bits are set and this bit is set then the device is busy.
30	Not Ready -- when this bit and bit 31 are set, the mass storage device is in a not ready condition.
29	End of File -- when this bit and bit 31 are set, an attempt to read beyond the end of the message was attempted.
28	Transmission Mode -- After a data transfer this bit is one.
27	Not used. beyond the highest available sector was attempted.
26	End of Allocated Blocks -- when this bit and bit 31 are set a write with no space available in the message queue was attempted.
25	Data Error -- when this bit and bit 31 are set a data error was encountered.
24	Hardware Error -- when this bit and bit 31 are set a hardware condition was encountered.
23	Lost Data -- when this bit and bit 31 are set a lost data condition was encountered.
22	Address Error -- when this bit and bit 31 are set a sector address error was encountered.
21	Seek Error -- when this bit and bit 31 are set a seek error was encountered.

Interactive

31	Reject -- always set when an error occurs. When no other error bits are set and this bit is set then the device is busy.
30	Not Ready -- when this bit and bit 31 are set, the terminal connection was lost.
29	Not used.
28	Transmission Mode -- After a data transfer this bit is zero.
25-27	Not used.
24	Hardware Error -- when this bit and bit 31 are set a hardware condition was encountered.
23	Not used.
22	Break -- when this bit is set a BRAEK was detected.
21	Not used.

Remote Batch

31	Reject -- always set when an error occurs. When no other error bits are set and this bit is set then the device is busy.
30	Not Ready -- when this bit and bit 31 are set, the terminal connection was lost.
29	Not used.
28	Transmission Mode -- After a data transfer this bit is zero.
26-27	Not used.
25	Data Error -- when this bit and bit 31 are set a data error was encountered.
24	Hardware Error -- when this bit and bit 31 are set a hardware condition was encountered.
23	Lost Data -- when this bit and bit 31 are set a lost data condition was encountered.
22	Break -- when this bit is set BREAK was encountered.
21	Framing Error -- when this bit and bit 31 are set a framing error was encountered.

Communications Network

31	Reject -- always set when an error occurs. When no other error bits are set and this bit is set then the device is busy.
30	Not Ready -- when this bit and bit 31 are set, the terminal connection was lost.
29	Not used.
28	Transmission Mode -- After a data transfer this bit is zero.
26-27	Not used.
25	Data Error -- when this bit and bit 31 are set a data error was encountered.
24	Hardware Error -- when this bit and bit 31 are set a hardware condition was encountered.
23	Lost Data -- when this bit and bit 31 are set a lost data condition was encountered.
22	Break -- when this bit is set BREAK was encountered.
21	Framing Error -- when this bit and bit 31 are set a framing error was encountered.

1857 SMD/MMD

31	Reject -- always set when an error occurs. When no other error bits are set and this bit is set then the device is busy.
30	Not Ready -- when this bit and bit 31 are set, the mass storage device is in a not ready condition.
29	Not Used.
28	Transmission Mode -- After a data transfer this bit is one. After a UINT instruction only this bit is set when an unsolicited interrupt occurs.
27	Write Protect Fault -- when this bit and bit 31 are set a write to a protected unit was attempted.
26	Not used.
25	Data Error -- when this bit and bit 31 are set a data error was encountered.
24	Hardware Error -- when this bit and bit 31 are set a hardware condition was encountered.
23	Lost Data -- when this bit and bit 31 are set a lost data condition was encountered.
22	Address Error -- when this bit and bit 31 are set a sector address error was encountered.
21	Seek Error -- when this bit and bit 31 are set a seek error was encountered.

1833-5 FDD

31	Reject -- always set when an error occurs. When no other error bits are set and this bit is set then the device is busy.
30	Not Ready -- when this bit and bit 31 are set, the mass storage device is in a not ready condition.
29	Not used.
28	Transmission Mode -- After a data transfer this bit is one. After a UINT request only this bit is set to indicate an unsolicited interrupt occurance.
27	Write Protect Fault -- when this bit and bit 31 are set a write to a protected unit was attempted.
26	Not used.
25	Data Error -- when this bit and bit 31 are set a data error was encountered.
24	Hardware Error -- when this bit and bit 31 are set a hardware condition was encountered.
23	Lost Data -- when this bit and bit 31 are set a lost data condition was encountered.
22	Address Error -- when this bit and bit 31 are set a sector address error was encountered.
21	Seek Error -- when this bit and bit 31 are set a seek error was encountered.

1829 Reader

31	Reject -- always set when an error occurs. When no other error bits are set and this bit is set then the device is busy.
30	Not Ready -- when this bit and bit 31 are set, the mass storage device is in a not ready condition.
29	End of File -- when this bit and bit 31 are set, a card with a 7 and 8 punch in the first column was read.
28	Transmission Mode -- After a data transfer this bit is set to indicate the type of card (0 - ASCII, 1 - Binary).
27	Feed Failure -- when this bit and bit 31 are set a card failed to feed through the read station. The card reader also goes Not Ready.
26	Not used.
25	Data Error -- when this bit and bit 31 are set a data error was encountered.
24	Hardware Error -- when this bit and bit 31 are set a hardware condition was encountered.
23	Input Tray Empty -- when this bit and bit 31 are set the input hopper is empty, the card reader is Not Ready and no card was read.
21-22	Not used.

1827 Printer

31	Reject -- always set when an error occurs. When no other error bits are set and this bit is set then the device is busy.
30	Not Ready -- when this bit and bit 31 are set, the mass storage device is in a not ready condition.
29	Not used.
28	Transmission Mode -- After a data transfer this bit is zero.
26-27	Not used.
25	Data Error -- when this bit and bit 31 are set a data error was encountered.
24	Hardware Error -- when this bit and bit 31 are set a hardware condition was encountered.
22-23	Not used.
21	Paper Fault -- when this bit and bit 31 are set the printer is normally out of paper and Not Ready.

1860-4 Tape

31 Reject -- always set when an error occurs. When no other error bits are set and this bit is set then the device is busy.

30 Not Ready -- when this bit and bit 31 are set, the mass storage device is in a not ready condition.

29 End of File -- when this bit and bit 31 are set, read or write of a tape mark occured.

28 Transmission Mode -- After a data transfer this bit is one.

27 Write Protect Fault -- when this bit and bit 31 are set a write to a protected unit was attempted.

26 Not used.

25 Data Error -- when this bit and bit 31 are set a data error was encountered.

24 Hardware Error -- when this bit and bit 31 are set a hardware condition was encountered.

23 Lost Data -- when this bit and bit 31 are set a lost data condition was encountered.

22 Load Point -- when this bit is set the tape unit is resting at load point.

21 End of Tape -- when this bit and bit 31 are set the End of Tape foil was encountered.

2558-3 Coupler

31 Reject -- always set when an error occurs. When no other error bits are set and this bit is set then the device is busy.

30 Not Ready -- when this bit and bit 31 are set, the switch on the coupler card is switched in the off line position.

29 End of file -- when this bit and bit 31 are set, an end of file condition was sensed.

28 Transmission Mode -- After a data transfer this bit is set to indicate the mode of the data transfer (0 - ASCII, 1 - Binary).

27 End of Record -- when this bit and bit 31 are set an end of record was encountered.

26 Not used.

25 Data Error -- when this bit and bit 31 are set a data error was encountered.

24 Hardware Error -- when this bit and bit 31 are set a hardware condition was encountered.

23 Not used.

22 Not used.

21 Not used.

Local CRT

31 Reject -- always set when an error occurs. When no other error bits are set and this bit is set then the device is busy.

21-30 Not used -- the status information is contained in the Peripheral Information Table (PIT).

BCLA/MUX

31 Reject -- always set when an error occurs. When no other error bits are set and this bit is set then the device is busy.

21-30 Not used. The status information is contained in the PIT.

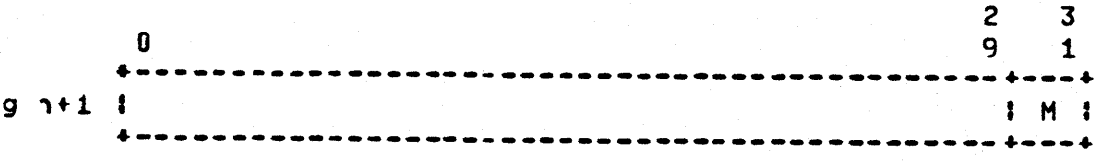
MPCLA

31 Reject -- always set when an error occurs. When no other error bits are set and this bit is set then the device is busy.

21-30 Not used. The status information is contained in the PIT.

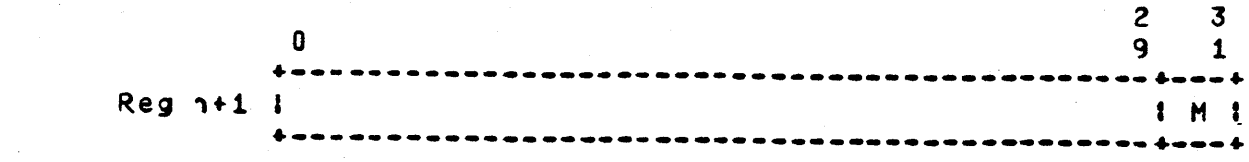
If no error conditions are set, then the Extended Status will contain the number of bytes not transfered; otherwise the Extended Status will contain further definiation of the error condition.

The MODE parameter in the SELECT ESR is defined for the 1829 card reader as follows:



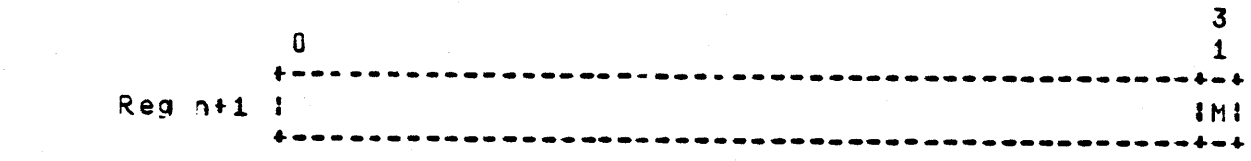
M 0 = Select 029 Hollerith conversion mode
1 = Select 026 Hollerith conversion mode
2 = Select ASD Hollerith conversion mode

The MODE parameter in the SELECT ESR is defined for the 1827 line printer as follows:



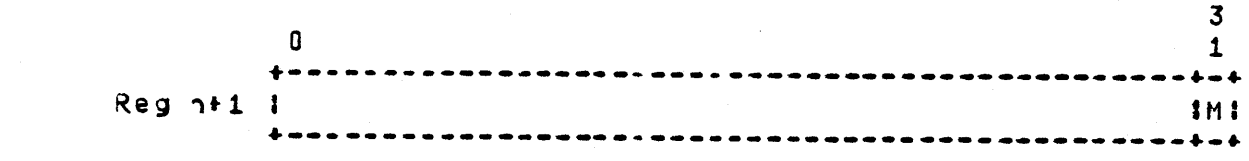
M = 0 = Select 96 ASCII character set.
1 = Select 64 ASCII character set.
2 = Fold 96 into 64 ASCII character set.

The MODE parameter in the SELECT ESR is defined for the 1860-4 magnetic tape as follows:



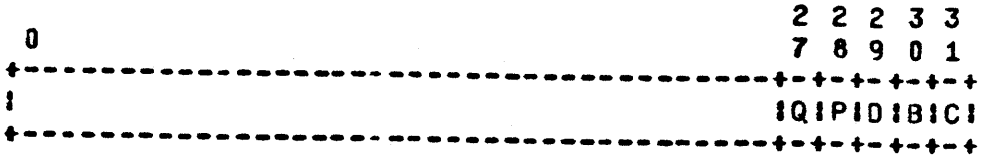
M = 0 = Select low recording density.
1 = Select high recording density.

The MODE parameter in the SELECT ESR is defined for the 2558-3 CYBER Channel Coupler as follows:



M = 0

The MODE parameter in the SELECT ESR is defined for Interactive Terminals and Remote Batch as follows:



C — 0 = 96 ASCII character set (default).
1 = 64 ASCII character set (96 folded into 64).

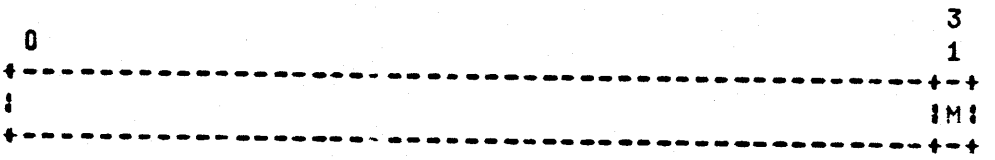
B — 0 = Line mode (default).
1 = Block mode (default).

D — 0 = Enable auto-prompt (default).
1 = Disable auto-prompt.

P — 0 = Use prompt already defined (default).
1 = Next WRITLU defines new prompt.

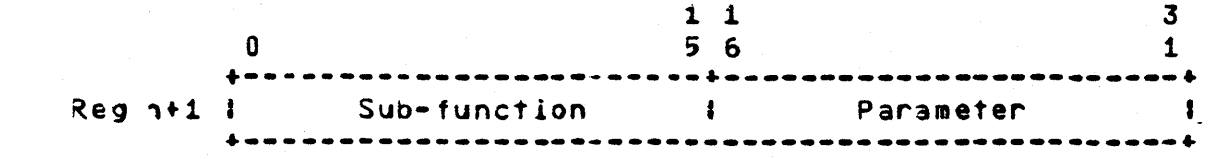
Q — 0 = Do not queue received data (default).
1 = Queue received data and status.

The CODE parameter in the FUNC ESR is defined for the 2558-3 CYBER Channel Coupler as follows:



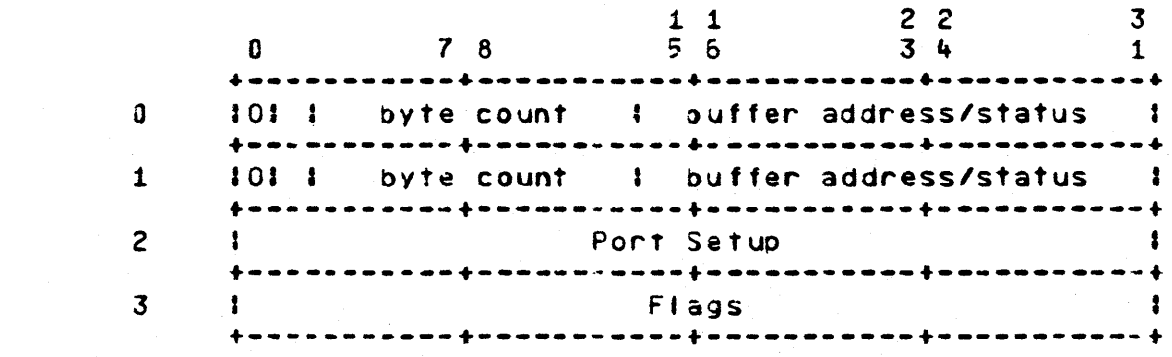
n — 0

The CODE parameter in the FUNC ESR is defined for the Local CRT, BCLA, and MPCLA as follows:



Sub-function
- 0 -- Define Peripheral Interface Table(PIT), the paramter specifies the first word address of the PIT.
- 1 -- Read address in PIT, the parameter specifies the port.
- 2 -- Write buffer address in PIT, the parameter specifies the port.
- 3 -- Port Setup in PIT, the parameter specifies the port.
- 4 -- Clear a port, the parameter specifies the port.

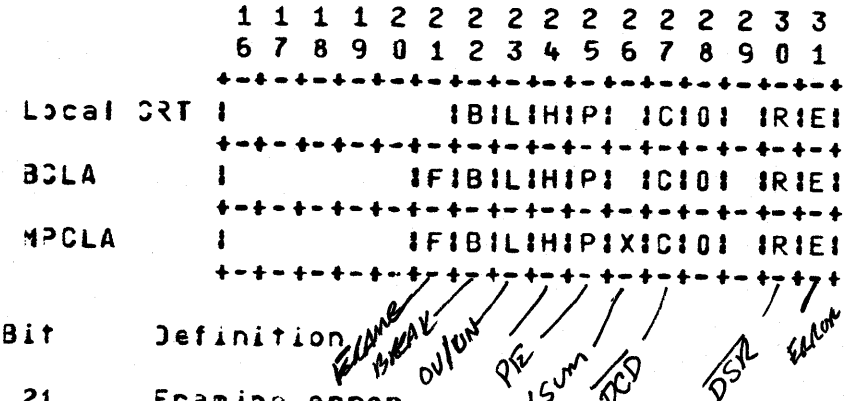
The Peripheral Interface Table (PIT) is used to communicated port (unit) information to the device manager. The PIT contains an entry for each port or unit and an entry has the following format:



Word	Bits	Definition
0	0	Control bit (0=system, 1=device manager).
	2-13	Number of bytes not used in received buffer.
	14-31	Receive buffer first byte address.
	16-31	Status (see below)

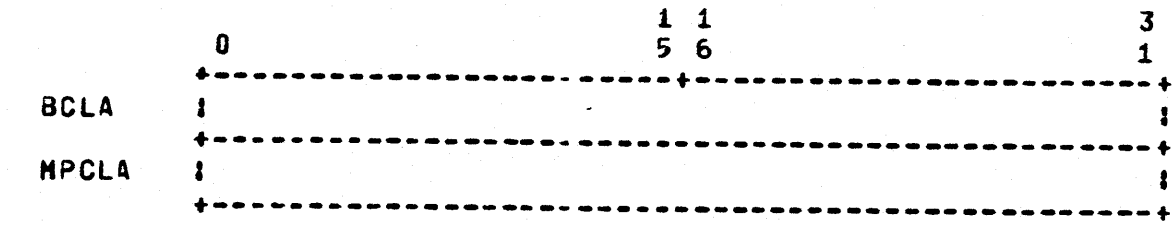
1	0	Control bit (0=system, 1=device manager).
	2-13	Number of bytes not transmitted.
	14-31	Transmit buffer first byte address.
	16-31	Status (see below).
2		Port/Unit setup information (see below).
3		Flags useable by device manager.

The status field in the PIT has the following definitions for the associated device:



Bit	Definition
21	Framing error.
22	Break detected.
23	Overrun/Underrun.
24	Hardware Error.
25	Parity Error.
26	Checksum error.
27	Carrier not on.
30	Data set not ready.
31	Error flag.

The port setup field in the PIT has the following definitions for the associated device:



APPENDIX C VALID HARDWARE TYPES

THE VALID HARDWARE TYPES WHICH CAN BE REQUESTED WITH THE EQUIP AND DEVICEQ ESRS AND THE HARDWARE TYPE FIELD RETURNED BY THE UTYP ESR ARE AS FOLLOWS:

HARDWARE TYPE CODE	MNEMONIC	TYPE	DEVICE IDENTIFIER
0	MEM	MEMORY	NA
1	DP	MASS STORAGE	PACK NAME
2	MT9	MAGNETIC TAPE (9-TRACK)	MT9___N_
3	CR	CARD READER	CARD__N_
4	CP	CARD PUNCH	PUNCH_N_
5	LP	LINE PRINTER	LINE__N_
6	CRT	OPERATOR CONSOLE	CRT___N_
7	TT	TELETYPE	TT____N_
8	CT	CARTRIDGE TAPE	CT____N_
9	PLT	PLOTTER	PLT___N_
10	FDD	FLEXIBLE DISK	FDD___N_
11	CCC	CYBER COUPLER	CCC___N_
12	MT7	MAGNETIC TAPE (7-TRACK)	MT7___N_
13	IT	INTERACTIVE TERMINAL	NA
14	RBT	REMOTE BATCH TERMINAL	NA
15	CN	COMMUNICATIONS NETWORK	NA
16	MQ	MESSAGE QUEUE	D/S NAME
17	MUX	MUX	MUX___N_
		BCLA	BCLA__N_
18	SMX	MPCLA (Z80-A)	MPCLA_NA
		MPCLA (Z80-B)	MPCLA_NB

Appendix D MPX/OS ERROR RECOVERY PROCEDURES

The MPX/OS error recovery procedures are dependent upon the device and error status codes returned by the device managers. The recovery techniques employed for each error status code and associated devices are described in the following paragraphs.

Busy - Bit 31 set and no other error bits set.

All Devices

The request is retired periodically and/or until the associated device manager generates an end of operation.

Not Ready - Bit 31 and 30 set, and not other error bits.

All Devices

The operator is notified and the request is retired every 5 seconds.

Data Error - Bit 31 and 25 set.
Lost Data - Bit 31 and 23 set.

Mass Storage

The request is retired three (3) times.

Tape

The following sequence is executed:

1) Backspace
2) Retry operation
3) Repeat 1&2 three times

Read	Write
4) Skip back 3 records	4) Backspace
5) Skip 2 records forward	5) Erase
6) Retry operation	6) Repeat 1-5 three times
7) Repeat 1-6 three times	

All others

Notify operator and await response of Accept (retry) or Reject (no-retry).

Hardware Error - Bit 31 and 24 set.

All Devices

The operator is notified and the error is considered irrecoverable.

Address Error - Bit 31 and 22 set.

Mass Storage Only.

The following sequence is executed:

1) Return to zero seek.
2) Retry operation.
3) Repeat 1-2 three (3) times.

Seek Error - Bit 31 and 21 set.

Mass Storage Only.

The operation is retried three (3) times.

Write Protect Fault - Bit 31 and 27 set.

Mass Storage and Tape only

The operator is notified and await response. If the operator Accepts then the operation is retried; otherwise the operation is considered irrecoverable.

Feed Failure - Bit 31 and 27 set.
Input Tray Empty - Bit 31 and 23 set.

Card Reader Only.

The operator is notified and await response. If the operator Accepts then the operation is retried; otherwise the error is considered irrecoverable.

Paper Fault - Bit 31 and 21 set.

Printer Only.

The operator is notified and await response. If the operator Accepts then the operation is retried; otherwise the error is considered irrecoverable.

End of Tape - Bit 31 and 21 set.

Magnetic Tape only.

The tape is unloaded and the operator is notified.