PRIME ENGINEERING HANDBOOK

for

SERIES 50 ARCHITECTURE

PE-T 500

Revision 2

Donald M. Koch

Copyright (c) 1988
Prime Computer Inc.
Natick, MA 01760
All rights reserved

PRIME RESTRICTED
This is the third release of the Prime Engineering Handbook, a document produced and maintained by Prime Computer Research and Development. Comments should be addressed to:

Donald Koch  
Prime Computer Research and Development  
MS 10C-13  
500 Old Connecticut Path  
Framingham, MA 01701  
Internet mail: aardvark@marvin.prime.com

Published by Prime Computer, Inc.  
October 1988  
The information contained in this handbook is subject to change without notice. Prime Computer Incorporated assumes no responsibility for errors that may appear in this document. This handbook is intended for the use of Prime employees only.

Copyright (c) 1985, 1988 by Prime Computer, Inc. All Rights Reserved.
Table of Contents

1. INTRODUCTION 1-1
  1.0.1. Acknowledgments 1-1
  1.0.2. Corrections and updates 1-1

2. COMMANDS 2-1
  2.1. Command Syntax 2-1
    2.1.1. ABBREVIATIONS 2-1
    2.1.2. COMMAND LINE VARIABLES 2-1
    2.1.3. OPTIONAL PARAMETERS 2-1
    2.1.4. ALTERNATIVE OPERAND SPECIFICATION, DEFAULTS 2-1
    2.1.5. Repeated Operands 2-1
  2.2. Filenames and Name Generation 2-1
  2.3. Filename Suffix Convention 2-3
  2.4. Command Resume Order 2-4
  2.5. Command Procedure Language (CPL) 2-4
    2.5.1. CPL Directives 2-4
  2.6. Command Functions 2-6
    2.6.1. Logical, Arithmetic, and Relational Functions 2-6
    2.6.2. String Functions 2-7
    2.6.3. File System Functions 2-8
    2.6.4. Miscellaneous Functions 2-8
  2.7. Command Descriptions 2-10
    2.7.1. Standard compiler options 2-96

3. ARCHITECTURE 3-1
  3.1. Argument Pointer (AP) 3-1
  3.2. Cache entries 3-1
  3.3. Checks 3-1
    3.3.1. Check header 3-2
  3.4. Concealed Stack/Queue 3-2
  3.5. Diagnostic Status Word (DSW) 3-2
    3.5.1. DSWSTAT 3-3
      3.5.1.1. 6350, 6550 3-3
      3.5.1.2. 9750, 9950, 9955 3-4
      3.5.1.3. 2250, 2550, 9650 3-5
      3.5.1.4. All other 50 series 3-6
    3.5.2. DSWPARITY 3-7
      3.5.2.1. 6350, 6550 3-7
      3.5.2.2. 9750, 9950, 9955 3-8
      3.5.2.3. 2550, 9650 3-10
      3.5.2.4. 750, 850 3-10
    3.5.3. DSWRMA 3-11
      3.5.3.1. 6350, 6550 3-11
      3.5.3.2. 9955 3-11
      3.5.3.3. 9750, 9950 3-11
      3.5.3.4. All other 50 series 3-11
    3.5.4. DSWPB 3-11
  3.6. Descriptor Table Address Register (DTAR) 3-11
  3.7. Entry Control Block (ECB) 3-12
  3.8. Faults 3-12
    3.8.1. Fault table entry 3-12
  3.9. Floating Point formats 3-13
3.9.1. Memory formats 3-13
3.9.2. Register formats 3-13
3.10. Indirect Pointers (IP) 3-14
3.11. KEYS, MODALS 3-15
3.12. Modals 3-16
3.13. Page maps 3-17
3.13.1. HMAP, LMAP 3-17
3.14. MMAP entry 3-18
3.15. Process Control Block (PCB) 3-18
3.16. Queue Control Block (QCB) 3-19
3.17. READY LIST 3-19
3.18. Registers 3-20
3.19. RSAV format 3-24
3.20. Segment descriptor word (SDW) 3-26
3.21. Semaphores 3-26
3.22. Stack frame 3-27
3.23. Stack Headers 3-27
3.24. STLDB 3-28

4. PRIMOS
4.1. ABORT FLAGS 4-1
4.2. EPF formats 4-1
4.3. FIGCOM 4-3
4.4. LOCKS, LCKCOM 4-4
4.5. PTUSEG 4-4
4.6. PUDCOM 4-4
4.7. Shared Segments 4-6
4.8. Semaphore allocation 4-8
4.9. Software interrupt flags 4-9
4.10. Software Stack Frame 4-10
4.11. SVC Interlude 4-11
4.12. UPCOM 4-11

5. File System
5.1. Diskrat Formats 5-1
5.1.1. 21 5-1
5.1.2. Rev 19 and 20 5-3
5.1.3. RAT specifier bits 5-3
5.2. Record Header Formats 5-4
5.2.1. Rektyp 5-5
5.2.2. DBS Record Headers 5-6
5.3. UFD Header and Entry Formats 5-7
5.3.1. UFD header formats 5-7
5.3.2. UFD Entry Formats 5-9
5.3.2.1. File entries 5-9
5.3.2.2. ACAT entries 5-11
5.3.2.3. DBS entries 5-12
5.3.2.4. File Information bits 5-13
5.3.3. Entry Control Word (ECW) 5-13
5.4. File system date format 5-14

6. SUBROUTINES
6.1. System routines - Supervisor Calls 6-1
6.2. Spool library 6-66
6.3. Application Library 6-67
6.4. DBMS routines 6-71

7. INSTRUCTION SET 7-1

7.1. Instruction formats 7-1
  7.1.1. S, R, and V mode 7-1
  7.1.2. I mode 7-2

7.2. Machine Instructions 7-3

7.3. Instruction Set Grouped by Function 7-15
  7.3.1. Address Pointer Operations 7-15
  7.3.2. Branch Operations 7-16
  7.3.3. Control Operations 7-17
  7.3.4. Character String Operations 7-18
  7.3.5. Decimal Arithmetic 7-18
  7.3.6. Field Operations 7-18
  7.3.7. Floating-point Operations 7-18
  7.3.8. Floating-point Skip Operations 7-18
  7.3.9. Generic Operations 7-19
  7.3.10. Integrity Operations 7-20
  7.3.11. Input/Output Operations 7-21
  7.3.12. Logicize Operations 7-21
  7.3.13. Memory reference/General register to register 7-22
  7.3.14. Mode Operations 7-22
  7.3.15. Memory-reference Operations 7-23
  7.3.16. Programmed I/O Operations 7-26
  7.3.17. Quad Floating Point Operations 7-26
  7.3.18. Register AP Operations 7-27
  7.3.19. Register Generic Operations 7-27
  7.3.20. Shift Operations 7-28
  7.3.21. Skip Operations 7-28
  7.3.22. P300 Virtual Memory Operations 7-29

8. OPERATIONS 8-1

8.1. Front Panel Controls 8-1
8.2. Standard VCP Procedures 8-1
  8.2.1. Cold start 8-1
  8.2.2. Warm Start 8-2
  8.2.3. Tape Dump 8-2
8.3. Boot Device Settings 8-2
  8.3.1. Booting from SMDs 8-2
8.4. Formatting disks: MAKE 8-3
8.5. Disk maintenance: FIX_DISK 8-3
8.6. Adding & changing user configurations: EDIT_PROFILE 8-4
8.7. VCP Commands 8-4

9. Peripheral I/O 9-1

9.1. Addresses 9-1
9.2. AMLC 9-2
  9.2.1. OTA 01 – Set Line Configuration 9-2
  9.2.2. OTA 02 – Set Line Control 9-2
9.3. ASR 9-3
9.4. DISK CONTROLLERS 9-3
  9.4.1. Disk Channel Program Definitions 9-3
9.5. Disk Device Numbers (PDEV) 9-4
9.6. Disk Errors
   9.6.1. Diskette Controller
   9.6.2. Storage Module (4004 Controller)
9.7. DMx control words
   9.7.1. DMA
   9.7.2. DMC
   9.7.3. DMQ
   9.7.4. DMT
9.8. Magtape
   9.8.1. Command Bit Definitions
   9.8.2. Magtape Commands
   9.8.3. Magtape Status
9.9. PROGRAMMED I/O (PIO)
   9.9.1. OCP — Output Control Pulse
   9.9.2. SKS — Skip on Condition
   9.9.3. INA — Input to A-Register
   9.9.4. OTA — Output from A=Register
   9.9.5. Standard Functions
9.10. RS-232-C pin-outs

Appendix A. ASCII character set
Appendix B. Conversion tables
   B.1. Octal-Decimal Conversion Table
Appendix C. Powers of Two
Appendix D. IOAS$ usage
Appendix E. References
Index
1. INTRODUCTION
This handbook provides a summary of information useful for the development and maintenance of Prime 50 Series hardware and software systems. While this book contains information useful to a general user community, the information is presented in very condensed form. It is assumed that the reader has had prior contact with this material and, therefore, that detailed descriptions are unnecessary.

Some of the information contained herein pertains only to the latest revision of PRIMOS. This information will be updated on a regular basis as new revisions are released. (Refer to the cover page for the revision currently reflected in this version of the handbook.)

NOTICE

Some of the information contained within this document is not released. Unless stated otherwise, or verified by reviewing a published Tech. Publs. document, none of the information in this manual should be disclosed. Prime confidential information will be shown in shaded print or will be marked with the NR notation; non-shaded print does not necessarily indicate release of a function.

1.0.1. Acknowledgments
I would like to thank the many reviewers, some of whom, due to my wonderful memory for names, I have undoubtedly forgotten. Those among the remembered are: Dick Snyder (who funded this clam bake), Ewan Milne, Marilyn Hammond (who kept fueling the architecture chapter), Don Slutz, Dave Hornbaker, Dave Peterson, John P. Jones, Kent Fielden, Chris Allen, Peter Borner, Martin Phillips, Martin Doughty, Peter Hassall, Patrick O\'Kane, Doug Rand, Cathy Phipps, C. James Cook and Jerry Kazin. The cover design was done by John Gustin.

1.0.2. Corrections and updates
Please send mail indicating corrections or updates to aardvark@marvin.prime.com.
2. COMMANDS

2.1. Command Syntax
The command descriptions in this manual use the following syntax:

2.1.1. ABBREVIATIONS
Uppercase letters represent abbreviations for commands and options. (When actually typing the
command or option, either uppercase or lowercase can be used.) For example:

    COMOutput

specifies the COMOUTPUT command.

2.1.2. COMMAND LINE VARIABLES
Italics indicate a variable for which specific information is to be substituted; for example:

    filename

should be replaced with a valid filename.

2.1.3. OPTIONAL PARAMETERS
Brackets enclose optional parameters for a command; for example:

    $Hutck [ALL]

2.1.4. ALTERNATIVE OPERAND SPECIFICATION, DEFAULTS
When an operand has more than one possible specification, choices are enclosed in braces ({ }),
brackets ([ ]), if optional, and separated by vertical bars (|). A default option, if any, is
underscored; for example:

    OPRpri { 1 | 0 }

The OPRPRI command accepts a single parameter of 1 or 0. If none is specified, the default
parameter is 0.

2.1.5. Repeated Operands
Ellipsis indicate an operand that may be repeated one or more times; for example:

    Close funit ...

The CLOSE command accepts one or more file unit specifications (separated by blanks or
commas).

2.2. Wildcards and Name Generation
Some commands accept wildcard names. Names are divided into components by periods. A
wildcard name is a filename that contains one or more of the following characters:

    @ matches zero or more characters in the corresponding component.

    @@ matches zero or more characters including periods.
matches any single character in the corresponding component except periods.

selects the subset of objects whose names do NOT match the wildcard name. If used, the "^^" must be the first character in the wildcard name.

Name generation from wildcarded and non-wildcarded names may be done by utilizing one or more of the following:

- Copy the corresponding component.
- Copies one or more components.
- Excludes a single component.
- Excludes one or more components.

literal-string
Replace component with literal-string.

+literal-string
Adds the component given by literal-string.

In addition to the options for the command, the following may be added to control wildcard action:

-FILE
SAM, DAM or CAM files.

-DIRECTORY
Directories.

-SEGMENT_DIRECTORY
Segment directories (SAM or DAM).

-ACCESS_CATEGORY
Access categories.

-RBF
Recovery based files.

-Modified_AFTER date.time
Objects last modified after date.time.

-Modified_Before date.time
Objects last modified before date.time.

-ACCESS After date.time
Accessed after date.time.

-ACCESS Before date.time
Accessed before date.time.

-BACKUP_AFTER date.time
Backed up after date.time.

-BACKUP_Before date.time
Backed up before date.time.
2.3. Filename Suffix Convention

| BASIC       | BASICV source file (BASICV). |
| BIN         | binary file.                  |
| C,CC        | C source file (CC, CI).       |
| CBL         | CBL source file (CBL).        |
| COBOL       | COBOL source file (COBOL).    |
| COMI        | command input file.           |
| COMO        | command output file.          |
| CPL         | CPL file (CPL, RESUME, JOB, PHANTOM). |
| DOC         | document text (output) file (SCRIBE). |
| FTN         | FORTRAN source file or insert file (FTN). |
| F77         | FORTRAN 77 source or insert file (F77). |
| INS.xxx     | insert file for given language. |
| LIST        | listing file.                 |
| MAP         | load map file.                |
| MOD         | Modula-2 source file (MODULA). |
| MSS         | document source file (SCRIBE). |
| PASCAL      | PASCAL source or insert file (PASCAL). |
| PLP         | PLP source or insert file (PLP). |
| PL1         | PL1 source or insert file (PL1). |
| PL1G        | PL1G source or insert file (PL1G). |
| PMA         | PMA source or insert file (PMA). |
| QUIC        | QUIC (QMS) output file (SCRIBE). |
RPx  Replacements of EPFs (COPY, BIND).
RPG  RPG source file (RPG, VRPG).
RUN  EPF runfile (BIND, RESUME).
RUNI Runoff input file (RUNOFF).
RUNO Runoff output file (RUNOFF).
SAVE R-mode runfile (RESUME).
SEG  segmented runfile (SEG).
SPSS SPSS input file (SPSS).
SPL  SPL source or insert file (SPL).
SYM  Modula-2 symbol table file (MODULA).
VRPG VRPG source file (VRPG).

2.4. Command Resume Order
Resumeable commands end with one of the suffixes: .RUN, .CPL, and .SAVE; or with no suffix at all. If more than one of these is found, the order of preference by which they are executed are: .RUN, .SAVE, .CPL, none.

2.5. Command Procedure Language (CPL)
To invoke CPL, type:

    Resume pathname[.CPL]
    or
    CPL pathname

CPL allows one statement per line. A statement is either a CPL directive or a PRIMOS command. A CPL directive has the form:

    &directive_name arguments

where arguments are expressions, CPL directives, or PRIMOS commands. Commands may be continued onto a second line by appending a tilde (~) to the end of the line.

2.5.1. CPL Directives

&ARGS [name[.type [default]] | name: -ctl_list,...]
    argument specification and validation. Arguments can be positional or control arguments. They can also be assigned types and default values.

&CALL routine_name
    invoke a routine defined by a &ROUTINE. Routine returns when &RETURN executes.

&CHECK expr &ROUTINE handler
    invoke handler if expr is true.
&DATA stmt
    compute input for a subsystem call at runtime. Format:
    &DATA stmt
    data_1
    ...
    data_n
    &END

&DEBUG opt_list
    enable and disable debugging facilities. Options are:
    &OFF
        turn off all debugging options.
    &NO_EXECUTE, &EX
        suppress execution of PRIMOS commands but interpret CPL directives.
    &EXECUTE, &EX
        enable execution of PRIMOS commands (default).
    &ECHO [ALL | COM | DIR]
        ALL echo PRIMOS commands and CPL directives (dft); COM echo PRIMOS commands; DIR echo CPL directives.
    &NO_ECHO [ALL | COM | DIR]
        ALL cancel all echoing (dft); COM cancel echoing of PRIMOS commands; DIR cancel echoing of CPL directives.
    &WATCH [var_1,...,var_{16}]
        add var_f to the list of watched variables. If no list is specified, all variables are watched.
    &NO_WATCH [var_1,...,var_{16}]
        remove var_f from the list of watched variables.

&DO [iteration]
    consider all statements between the &DO and &END as a single statement. Format:
    &DO [iteration]
        stmt_1
        ...
        stmt_n
    &END

    iteration is: [var := start [ &TO expr ] [ &BY expr ] [ &LIST list ] [ &ITEMS items ] ] [ &WHILE cond ] [ &UNTIL cond ] . Note: 1024 character limit for list.

&EXPAND switch [&USING processor_name]
    enable and disable statement expansion. switch can be ON or OFF.

&GOTO label
    transfer control to label.

&IF cond &THEN true_stmt &ELSE false_stmt
    conditional test.

&LABEL label_name
    define label that identifies the next statement.

&ON condition &ROUTINE handler_label
    define handler for condition.

&RESULT expr
    return the value of a user defined function.
&RETURN [severity] [MESSAGE text]  
return severity code to the invoker.

&REVERT condition  
cancel the handler for condition.

&ROUTINE routine_name  
identify following code as an internal routine.

&SELECT expr  
evaluate &SELECT expr, compare to &WHEN expr, and execute appropriate stmt.

&SELECT expr  
&WHEN expr1 [ , ..., exprn ]  
stmt  
[ &WHEN expr1 [ , ..., exprn ]  
stmt  
...  
[ &OTHERWISE  
stmt ] ]  
&END

&SET_VAR var1 [ , ..., varn ] := value  
set one or more local or global variables.

&SEVERITY [ERROR | WARNING] [FAIL | IGNORE | ROUTINE label]  
specify the action to be taken when certain severity codes are produced.

&SIGNAL condition [NO_RETURN]  
raise the condition and search its handler.

&STOP [severity] [MESSAGE text]  
abort current CPL procedure and any routines it has invoked.

&TTY  
take input from terminal. Used within an &DATA block.

&TTY_CONTINUE  
take input from previous command stream. Used within an &DATA block.

2.6. Command Functions
In the following examples beginning and ending brackets are entered literally.

2.6.1. Logical, Arithmetic, and Relational Functions

[CALC infix_expr]  
evaluate expressions containing the following logical operators in the order indicated:

(highest):  ^ unary + unary -  
/ *  
+ -  
= ^= < > <= >=  
&

(lowest):  |
[HEX hex_string]
return a string representation of the decimal equivalent of hex_string.

[MOD dec_str, dec_str2]
return the string representation of the decimal equivalent of dec_str, modulo dec_str2.

[OCTAL oct_str]
return the string representation of the decimal equivalent of oct_str.

[TO_HEX dec_str]
return a string representation of the hexadecimal equivalent of dec_str.

[TO_OCTAL dec_str]
return a string representation of the octal equivalent of dec_str.

2.6.2. String Functions

[AFTER str find_str]
return in quotes the substring of str that occurs to the right of the leftmost occurrence of find_str in str.

[BEFORE str find_str]
return in quotes the substring of str that occurs to the left of the leftmost occurrence of find_str in str.

[INDEX str find_str]
return the position of the leftmost occurrence of find_str in str, else 0.

[LENGTH str]
return the number of characters in str.

[NUM str]
return TRUE if str is the true null string, else "" and FALSE.

[QUOTE str...]
add outer pair of quotes and double quotes already in str.

[SEARCH str, str]
returns position in str of first character contained in str2, otherwise 0.

[SUBST str, str2, str3]
replace all occurrences of str2 in str, with str3.

[SUBSTR str str_pos [num_chars]]
return in quotes the num_chars characters in str to the right of and including the character in position str_pos.

[TRANSLATE str [out_chars in_chars]]
return the string that is the result of replacing each character in str that appears in the ith position in in_chars with the ith character in out_chars.

[TRIM str [which_side] [trim_char]]
return in quotes the result of trimming a leading or trailing sequence from str. which_side can be -Right, -Left, or -Both.
2.6.3. File System Functions

[ATTRIB path option [-BRief]]
return information about path. option can be -TYPE, -DTM, -DTB, or -LENth (-L). -BRief suppresses some error messages.

[DIR path [-BRief]]
return in quotes the directory portion of path. -BRief suppresses some error messages.

[ENTRYNAME path]
return the entryname portion of path.

[EXISTS path [type] [-BRief]]
return TRUE if pathname path of type type exists, else FALSE. type can be -ANY, -FILE, -DIRectory, -SEGment_DIRectory, or -Access_CATegory. -BRief suppresses some error messages.

[GVPATH]
return the pathname of the active global variable file, if any, otherwise returns -OFF.

[OPEN_FILE path status_var -mode m]
open path on an available unit and return the unit number. m can be R, W, or WR.

[PATHNAME rel_path [-BRief]]
return in quotes the full pathname of rel_path. -BRief suppresses some error messages.

[READ_FILE unit status_var [-BRief]]
read a record from the file open on unit and return the quoted value of that record. -BRief suppresses some error messages.

[WILD wild1,wild2...wildn [ctl_arg...]] [-BRief]]
list the entrynames that match wild1 and ctl_arg. ctl_arg can be -BF date, -AF date, -FL, -DIRS, -SEGDIRS, and -SINGLE unit_var. -BRief suppresses some error messages.

[WRITE_FILE unit text]
strip text of one layer of quotes and write text on the file open on unit. Return 0 if successful; otherwise, nonzero.

2.6.4. Miscellaneous Functions

[ABBrev -EXPand text]
returns text with the abbreviations expanded.

[CND_INFO ctl_flag]
return information on the most recent condition on the stack. ctl_flag can be -NAME, -CONTinue_SWitch, and -RETurn_PerMIt.
[DATE [ctl]]
return date/time according to ctl. ctl can be -FULL, -USA, -UFULL, -DAY, -MONTH, -YEAR, -TIME, -AMPM, -DOW, -CAL, -TAG, -FTAG, -VFULL, or -VIS.

[GET_VAR expr]
return the value of the variable named expr if it has been defined; otherwise, $UNDEFINED$.

[QUERY text [default] [-TTY]]
on the terminal, return text in quotes and followed with a question mark. -TTY forces input from the terminal.

[RESCAN str]
return the result of stripping one level of quotes from str and evaluating any function calls or variable references no longer appearing in quotes.

[RESPONSE text [default] [-TTY]]
on the terminal, print text in quotes and followed with a colon. -TTY forces input from the terminal.
2.7. Command Descriptions

The following notations may indicated for a command:

(CF) - Also a Command Function
(DSM) - DSM restricted function
(EX) - External Command
(EPF) - EPF Command
(IN) - Internal Command
(LO) - Command executable when logged out
(NR) - Not Released
(OBS) - Obsolete.
(OP) - Operator Command
(P2) - Can be used in PRIMOS-II
(QT) - Qualified Tool
(SA) - System Administrator only
(revno) - New; released at revno

$$\text{batch-command}$$

Flag a command to be passed on to the batch monitor. (EX)
Ref: PRIMOS Commands Reference Guide [49].

**ABbrev [pathname] [options]**

Invoke abbreviation preprocessor. (IN)
Options:

-Change name1 ...[nameN]
-Change_Argument name1 ...[nameN]
-Change_Command name1 ...[nameN]
-Change_Name oldname newname
-CReate
-DeLete name1 ...[nameN]
-HELP
-LIST [name1 ...[nameN]]
-OFF | -ON
-STatus
-No_Query
-No_Verify | -Verify
-WILD
-Add name rest-of-line
-Add_Argument name rest-of-line
-Add_Command name rest-of-line
-EXECute rest-of-line
-EXPand rest-of-line
-Expand_Execute rest-of-line

Ref: PRIMOS Commands Reference Guide [49].

**ADdisk [PROTECT] pdev1 [...]pdev9 | -RENAME packname**

ADdisk packname1 [...]packname9 | -ON node-name

Add disks to system. (IN, OP)
Ref: Operator's Guide to System Commands [35].

**Add_Remote_ID user-id [password] -ON nodename**

[-PROJect project-id] [-PROMPT]
Specify id for slaves on remote machines. (IN)
Ref: PRIMENET Guide [45].

ADMIN_LOG logname [log-type] subcommand

Create, list, modify, purge or delete DSM logfiles. (EPF, 21.0, DSM)

Log-type:
-Private_LOG
-System_LOG [node | nodegroup]

Subcommands:
-CREATE [attributes]
-MODIFY attributes
-PURGE [age | ALL]
-DELETE
-LIST
-Help
-USAGE

Attributes:
-CYCLic | LINear
-MAXimum_SIZE records
-MINimum_SIZE records
-WARNING_LEVEL percent
-RETain [days]
-PURGE_TIME hh:mm

AIDS

Invoke the PRIMEAIDS system. (EX, OBS)
AMLC [Ty |-TRan | TTYHs | TRANHS | TTYNop | TTYUPC | TTYHUP |
| TT8BIT | [ASD] line [config [lword]]
Set AMLC line characteristics. (IN,OP)
line, config, lword are octal. Command is obsolete. use SET_ASYNC (20.2).

<table>
<thead>
<tr>
<th>config</th>
<th>Line Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>2033</td>
<td>110 BAUD</td>
</tr>
<tr>
<td>2113</td>
<td>134.5 BAUD</td>
</tr>
<tr>
<td>2213</td>
<td>300 BAUD</td>
</tr>
<tr>
<td>2313</td>
<td>1200 BAUD (default)</td>
</tr>
<tr>
<td>2413</td>
<td>9600 BAUD (programmable clock)</td>
</tr>
<tr>
<td>2513</td>
<td>75 BAUD (or by jumper or ICS JUMPER directive)</td>
</tr>
<tr>
<td>2613</td>
<td>150 BAUD (or by jumper or ICS JUMPER directive)</td>
</tr>
<tr>
<td>2713</td>
<td>1800 BAUD (or by jumper or ICS JUMPER directive)</td>
</tr>
</tbody>
</table>

lword:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Meaning when on</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Half duplex</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>2</td>
<td>No LF after CR if half duplex</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>3</td>
<td>XOFF/XON recognition</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>XOFF received, output suspended</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>5</td>
<td>Buffered protocol, use bit 6 for sense</td>
<td>004000</td>
<td>0800</td>
</tr>
<tr>
<td>6</td>
<td>If set, send XOFF on ^DTR else send XOFF on DTR</td>
<td>002000</td>
<td>0400</td>
</tr>
<tr>
<td>7</td>
<td>Enable error detection, send NAK on parity or overflow</td>
<td>001000</td>
<td>0200</td>
</tr>
<tr>
<td>8</td>
<td>Reserved</td>
<td>000400</td>
<td>0100</td>
</tr>
<tr>
<td>9-16</td>
<td>User number ( 0 =&gt; assignable)</td>
<td>000377</td>
<td>01FF</td>
</tr>
</tbody>
</table>


ARCHIVE [-LIST] pathname -MT n -VOLID name [options ]
Archives disk files onto magnetic tape. (EPF)
-LIST indicates that pathname contains a list of objects. Options:
-INDEX [pathname ]
-IndexL_Levels [n] (1 <= n <= 99)
-LeVel s n (1 <= n <= 99)
-No_Query
-VerIfY
-Ty
-REMARK [character-string ]
-DeLeTe
-OWNer user_name
-OVerWrite
-CAtalog_PAthname pathname
-Cam_To_Dam
-CAtalog_PAthname pathname
-Compati_able_VersioN rev
-VA Lidate
-Help [USER | OPERATOR | option | EXAMPLE | ERROR [error#] | ERROR_LIST | WILDCARDS | HELP]

and the standard wildcard options. Ref: Data Backup and Recovery Guide [7].

ARCHIVE_RELEASE -VOLID name [options]

Release a tape generated with ARCHIVE for reuse. (EPF)
Options:
-MT n (0 <= n <= 7)
-REEL n (1 <= n <= 255)
-OWNer user_name
-CAtalog_PAthname pathname
-No Query
-Help [USER | OPERATOR | option | EXAMPLE | ERROR [error#] | ERROR_LIST | WILDCARDS | HELP]

and the standard wildcard options. Ref: Data Backup and Recovery Guide [7].

ARCHIVE_RESTORE object-pathname [target-pathname]
-MT n [options]

Restore files from an ARCHIVE tape to disk. (EPF)
Options:
-VOLID name [name]...
-INDEX [pathname]
-INDEX_Levels [n] (1 <= n <= 99)
-REEL n (1 <= n <= 255)
-Try
-Cam_RBFB
-Dam_RBFB
-From_Logical_Tape n
-From_Save_Number n
-To_Logical_Tape n
-To_Save_Number n
-MAGSAV
-WRitten_After [date]
-From_Save_Number n (1 <= n <= 255)
-WRitten_Before [date]
-To_Save_Number n (1 <= n <= 255)
-No Query
-VERIFY
-OWNer user_name
-CAtalog_PAthname pathname
-COMBine
-REPLACE
-Help [USER | OPERATOR | option | EXAMPLE | ERROR [error#] | ERROR_LIST | WILDCARDS | HELP]

and the standard wildcard options. Ref: Data Backup and Recovery Guide [7].

Prime Restricted 2-13
ASRCWD [number]

Set virtual ASR control word. (IN, OBS)
Ref: PRIMOS Commands Reference Guide [49].

ASSign device [-WAIT]
ASSign DISK pdev [-PRlorty_SELECT]
ASSign ASYNC -LINE n
ASSign AMLC protocol amlc-line config lword

Assign peripheral device. (IN)

device: CARdr
   Cenpr
   CE2pr
   CRn (n=0,1)
   Disk pdev
   GSrn (n = 0..3)
   MGrn (n = 0..3)
   PBhist
   PLot
   PRn (n=0..3)
   PTr
   PUinch
   SMLCnn (nn=00..07)
   MTX-ALias MTIden (ldev=0..7)
   MTpdev [-ALias MTlde] [mt-options]
   (pdev, ldev=0..7)

mt-options can be:

   -TPID id
   -MOUNT
   [-RINGON | -RINGOFF]
   [-7TRK | -9TRK]
   -RETENSION
   -DENSITY bpi (bpi=800, 1600, 3200, 6250)
   -SPEED spd (spd=25, 100)

protocol, config, and lword are described under the AMLC command. Ref: PRIMOS

ATM

Enter Advanced Text Management Option Selection Menu of OAS. (EX)

ATM_ADMIN

Maintain OAS document database. (EX, OBS)
Replaced by OA_ADMIN. Ref: OAS System Administrator's Guide [30].

Attach [pathname ] [passwd ] [ldev ] [key ]

Attach to UFD. (IN)

ldev:

100000 - search MFDs of all started devices (default).
177777 - search MFD of current device.
n=0..77 - search MFD on logical device n.

key:
0 - attach to UFD and set home (default).
1 - don't set home UFD after attach to subUFD.
2 - set home UFD after attach to subUFD.
177777 - attach to UFD and don't set home.

Ref: PRIMOS Commands Reference Guide [49].

AUTOPSY [filename]

Dump analyzer. (EX/EPF, QT)
Internal commands:

Clrmap
   Clears out old maps so new ones may be read in.
CHkprt
   Prints a description of the last check handled.
COMsearch address
   Searches symbol table for the common block at address.
Dump start_address end_word [user]
   Dumps specified region of memory in octal.
DAte Displays date header for current dump.
DDqb [start [end]] [-FREE] [-USED] [-MeTerRs]
   Dumps Disk Queue Blocks.
DLcb [start [end]] [-LRU_list] [-Hash_Table]
   Dumps Locate Control Blocks.
DSemaphore address [user]
   Dumps semaphore at address.
Ecbsearch address
   Searches symbol table for the procedure with the given ECB.
From treename
   Reads a crash dump from treename
FSchk
   Checks file system tables for consistency.
Help [command | topic | NEW]
   Displays helpful information for the selected topic. NEW displays information on the latest AUTOPSY updates.
IPCDump
   Dump the inter-process communications area.
Keyprt keys modalss
   Decodes keys and modals. See 3.11.
Lprnt
   Displays status of all N1LOCKS. See 4.4.
LBsearch address
   Searches symbol table for procedure with given LB.
LBNames address
   Lists all procedures with specified LB.
LOCsearch address
   Searches for the symbol nearest to address.
Map [maptree1 [maptree2]]
   Reads in maps (by default, from MAPS UFD).
Othsearch address
   Searches for other symbol (not procedure or common) at address.
Pdump user
   Displays PCB (and concealed stack) for user. See 3.15.
PAgchk
   Checks memory maps for consistency. See 3.13.1.
PAUse  
Exits AUTOPSY but leaves everything in place so you can restart.
PBsearch address  
Searches for procedure closest to specified PB.
PMap segno user  
Prints HMAP and LMAP for specified segment. See 3.13.1.
Quit  
Exits AUTOPSY.
RDump [SLAVE | AP]  
Dumps absolute register set, either for master (default), slave, or AP board.
Read treename mtunit  
Reads dump into treename from tape unit mtunit.
REAL [ON | OFF]  
Use real memory as opposed to a read-in dump.
RESET segno user  
Restores given user's segment into seg 4001 and invokes VPSD.
RPmt [L:Live | AL:Last] [SL:Slave] [R: -REGno]  
Displays registers for live or last process (live is default). If SLAVE is specified, displays registers for live or last on slave ISU.
Status [user | ALL | US]  
Displays status for specified user, all processes, or only user processes.
SYMBOL symbol [symbol...]  
Returns information about given symbols.
Trace user [address]  
Traces stack for specified user, from current SB or address. Trace commands are:

   Father
      Move to the father of this frame.
   Son  Move to the son of this frame.
Displays specified terminal buffer in format indicated (default is user's input and output buffers in unformatted ASCII).
Unit [address | offset] [-UNFormatted]  
Displays unit table entry at address (or offset from UTCOM$).
UOwned [address | offset] [-DISP]  
Displays owner (user and unit number) of specified unit, with optional display of unit table entry.
UTime  
Gives time system was running in seconds.
UTbl user [-UNFormatted]  
Displays unit table for specified user.
UTEntry user [unit | -CURrent | -HOME | -INITial] [-UNFormated]  
Displays specified unit table entry.
Vpsd  
Enter VPSD.
I Primos_command_line  
Executes argument as a PRIMOS command. (Must be internal or an EPF.)

AVAIL [ partition | -LDEV n | * ] [-NORM]  
Show disk usage statistics. (EX)
Ref: PRIMOS Commands Reference Guide [49].

BACKUP [-LIST] pathname -MT n -VOLID name [options]  
Backup files from disk to magnetic tape. (EPF)
-LIST indicate that pathname contains a list of objects. Options:
-INDEX [pathname]  
-Index_Levels [n] (1 <= n <= 99)  
-Levels n (1 <= n <= 99)  
-No_Query  
-Cam_To_Dam  
-Compatible_Version rev  
-EXpiry_Date date  
-NO_CATALOG  
-No_Spawn_Disk_Reader  
-Spawn_Disk_Reader  
-OVERWrite  
-INcremental  
-VALIDate  
-VERIFY  
-Tty  
-REMARK [character-string]  
-HELP [USER | OPERATOR | option | EXAMPLE | ERROR [error#] | ERROR_LIST | WILDCARDS | HELP]  

and the standard wildcard options. Ref: Data Backup and Recovery Guide [7].

BACKUP_RELEASE -VALID name [options]

Release a BACKUP tape for reuse. (EPF) Options:

-MT n (0 <= n <= 7)  
-REEL n (1 <= n <= 255)  
-No_Query  
-HELP [USER | OPERATOR | option | EXAMPLE | ERROR [error#] | ERROR_LIST | WILDCARDS | HELP]  


BACKUP_RESTORE object-pathname [target-pathname]

-VALID name [options]

Restore a file from a BACKUP tape to disk. (EPF) Options:

-RECOVER  
-INDEX [pathname]  
-Index_Levels [n] (1 <= n <= 99)  
-REEL n (1 <= n <= 255)  
-Tty  
-Cam_RB (F)  
-Dam_RB (F)  
-From_Logical_Tape n  
-To_Logical_Tape n  
-MAGSAV  
-WRitten_After [date]  
-WRitten_Before [date]  
-From_Save_Number n (1 <= n <= 255)  
-To_Save_Number n (1 <= n <= 255)  
-No_Query  
-VERIFY  
-COMBine  
-REPLACE
-HELP [USER | OPERATOR | option | EXAMPLE | 
ERROR | ERROR_LIST | WILDCARDS | HELP]

and the standard wildcard options. Ref: Data Backup and Recovery Guide [7] and 
Operator’s Guide to System Backups [34].

BASIC [pathname]

BASIC language interpreter. (EX)
Ref: Interpretive BASIC User’s Guide [20].

BASICV [pathname] [-MIN]

Virtual memory BASIC. (EX)

BASINP pathname

Read BASIC program from paper tape. (EX)
Ref: Interpretive BASIC User’s Guide [20].

BATCH {-DisPlay | -STatus | 
SYSTEM {-START | -STOP | -PAUSE | -CONTINUE}}

Invoke BATCH monitor. (EX)
Ref: Operator’s Guide to the Batch Subsystem [31].

BATGEN {-STATUS | -DISPLAY [queue]}

Query BATCH queues. (EX)
Subcommands:
  BLock {queue | ALL}
  UNBLock {queue | ALL}
  CAP {queue | ALL}
  UNCAP {queue | ALL}
  DisPlay [{queue | ALL}]
  STatus
  File [pathname]
  Quit

Ref: Operator’s Guide to the Batch Subsystem [31].

Binary pathname

Open file unit 3 for binary output. (IN) Ref: PRIMOS Commands Reference Guide [49].

BIND [epf-name] [commands]

EPF linker. (EPF, 19.4)
Reference: Programmer’s Guide to BIND and EPFs. BIND subcommands are:
  LOad list-of-options-and-pathnames
    Loads a binary or runfile into the EPF currently being built. Options are: -PAGE,
    -FORCE, or -FORCE PAGE.
  Library list-of-options-and-pathnames
    Loads a binary file from LIB. Same options as LOad.
  ReLoad list-of-options-and-pathnames
    Reloads a binary into an existing EPF, replacing an old entry of the same name.
    Same options as LOad.
DYNT list-of-names
   Creates a dynamic entry for the list-of-names.
SYSymbol name definition [size]
   Creates a symbol, name, at the location specified by definition. (Default size is 0.)
ALLOCate name size
   Allocates size halfwords of storage for name.
MAP [map-dest] [map-option]
   Creates a load map. (Default is a full map without flags on the terminal.)
MAIN eob-name
   Changes the main entrypoint to eob-name. (Default is the first entry loaded.)
HELP [command | -LIST]
   Gives help on a command or a list of commands.
Quit
   Quits BIND without creating or modifying an existing run file.
FILE [epfname]
   File the EPF as epfname or as the current runtime name (either the same as the first binary file loaded or that given on the command line).
Common Warning
   Turns on common size mismatch checking. (Default)
No Common Warning
   Turns off common size mismatch warnings. Will still give error for illegal redefinition.
Resolve Deferred Common
   Allocates space for all deferred common blocks.
COMMENT comment
   Inserts a comment into the EPF comment field. Takes the remainder of the line. Cannot be entered on the command line.
VERSION string
   Sets the version stamp for this EPF to string.
ENTRYname list-of-names | -ALL | -NONE
   Add list-of-names as entrypoints to the current EPF library being built. -ALL implies that all successively loaded modules will have all entrypoints added; -NONE excludes subsequent entries from being added to the entrypoint list. (Default is EN -NONE)
LibMode library-class [-REGister]
   Generate a library EPF of the given library-class.
ProgMode [-NORMAL | -REGister]
   Generate a program EPF (default).
Initialize DATA [-OCTal] value
   Initialize all uninitialized static areas with value. Slows down program startup.
COMpress
   Removes data unnecessary to program execution; saves file space.
WildCard [file-type-options] [verify-option]
   Allow command line processing of wildcarded pathnames using @ and +. (Default is on with all file types but not RBF.)
No WildCard
   Disallow command line processing of wild cards.
ITeRation
   Allow command line iteration using parentheses. (Default)
No ITeRation
   Disallow command line iteration.
TreeWalk
   Allow tree-walking. (Default)
No TreeWalk
   Disallow tree-walking.
NameGenPos position
   Perform equal name generation from the positionth argument. (Default = 1)
No Generation
   Do not allow name generation.
Search Rule_Verify
   Causes BIND to print out the full path of each file it loads.
AKLMB
Allocates KLM block for serialization.

BOOT_ATTACH
Used by BOOT_SAVE/BOOT_RESTORE. (EX, P2)

BOOT_CREATE [pathname] [-Help] [-MT[n]] [-No_Query]
Makes a boot tape. (EX)
Ref: Operator's Guide to System Commands [35].

BOOT_IMPCODE
Used by BOOT_SAVE/BOOT_RESTORE. (EX, P2)

BOOT_RESTORE
Restore files from BRMS/BACKUP tape under Primos 2. (EX, P2)

BOOT_SAVE
Save files to BRMS tape under Primos 2. (EX, P2)

BOOT_TREE
Used by BOOT_SAVE/BOOT_RESTORE. (EX, P2)

BUILD [component] [-No_commands] [-DeBuG] [-From pathname]
[var1=value1 ...] [-Help]
BUILD reads a description file and brings a program (programs) up to date. (EPF, QT)
Ref: BUILD... [40].

CARDSPool
Submit a job from the card reader to an RJE site. (EX)

CBL filename [CE-options]
Low intermediate ANSI-74 COBOL compiler. (EX)
See compiler options, section 2.7.1.

CBLDML [input-pathname [output-pathname] [error-pathname] [options]]
COBOL Data Manipulation Language. (EX)
Options:
-Input pathname
-OUTPut pathname
-ERROR pathname
-DYNamic
-No_Line_Number
Ref: DBMS Data Manipulation Language Reference Guide.
CBLSUBS source [-Output pathname] [-List pathname]

CBL DBMS subschema processor. (EX)
Ref: DBMS Data Manipulation Language Reference Guide.

CC [pathname | -SOURCE pathname | -INPUT pathname] [options]

C compiler. (EX)
- BINARY [pathname | YES | no]
- NOBIG | -big
- BIT8 | -nabit8
- NOCOMPATIBILITY | -compatibility
- COPY | -nocopy
- CONVERT | -noconvert
- ERRTTY | -noerrtty
- NOFRM | -frn
- LISTING [<pathname> | YES | no | tty]
- NOEXPLIST | -explist
- NOSILENT | -silent
- statistics
- debug
- 64v
- noonunit
- NOANSI | -ansi
- xref
- xrefs
- CDBG | -spildbg
- production
- NOPOP | -pop
- INTL | -ints
- psi1
- psi2
- psi3
- NOVERBOSE | -verbose
- NOCHECKOUT | -checkout
- include pathname
- define name [1 | value]
- NO_STORE_OWNER_FIELD | -store_owner_field
- NOUNIX | -unix
- LBECB | -pbecb
- 32IX

Ref: C User’s Guide [5].

CDML

COBOL Data Manipulation Language. (EX)
Ref: DBMS COBOL Subschema Guide.

Change_PassWord [old-password]

Change login password. (IN)
A new password is then asked for twice with echo turned off. Ref: PRIMOS Commands Reference Guide [49].

CHap [-userno | ALL] [priority [timeslice]]
Change user priority. (IN, OP)

\[ \text{priority} = 0.3; \text{UP, DOWN, LOWER, -IDLE, -SUSPEND, DEFAULT (default=1).} \]

\[ \text{timeslice is in tenths-of-a-second (default=3).} \]


Close \{pathname | [-]ALL | -UNIT unit \( \_1 \) \( \ldots \) unit \( \_n \) | \( \ldots \) unit \( \_n \)\}

Close file unit(s). (IN)

[-]ALL closes all file units above unit 1; does not close the COMO unit. Use of pathname from the console will close the file for all users. Ref: PRIMOS Commands Reference Guide [49].

CLUP [-Userno user-number] [-FORCE]

Cleanup processor for ROAM, PRISAM and DBMS. (EX)
Ref: ROAM Administrator's Guide [53].

CMPF path \( \_1 \) path \( \_2 \) \( \ldots \) path \( \_5 \) [option...]

Compare ASCII files. (EX)
Options:

- MINL \( \{ n \} \) (default = 3)
- BRIEF
- REPORT report-pathname

Ref: PRIMOS Commands Reference Guide [49].

CN_RBF old-pathname new-filename [-ALL]

Change the name of an active or inactive ROAM file. (EX)
Ref: ROAM Administrator's Guide [53].

CName oldpathname newfilename

Change name of file. (IN)
Ref: PRIMOS Commands Reference Guide [49].

CNVTMA infilename outfilename

Convert load map for PMA. (EX, OBS)
Converts load map into format usable by PSD 'LS' command.

COBOL pathname [option...]

or

COBOL [option...] -I pathname [...option]

Invoke COBOL compiler. (EX, OBS)
Use CBL. Options can be:

- Binary [pathname | NO | YES]
  Define binary file generation. Default: YES.
  - EXPList
  Generate an expanded listing file.
- Input pathname
  pathname is source program.
-Listing [pathname | NO | YES | TTY | SPOOL]
  Define listing generation. Default: YES.

-NOEXPLIST
  Do not generate expanded listing file.

-64RGenerate relative-addressed code.
-64VGenerate segmented-addressed code.

COMinput pathname [funit]
  [-PAUSE | -CONTINUE [funit] | -TTY | -Start | -End]

  Change command input stream. (IN)
  CO -S = s; CO -CONTIN
  CO -E = CO -TTY

  Ref: PRIMOS Commands Reference Guide [49].

COMM_CONTROLLER [-Help | -INIT | -LOAD | -SHutdown | -UpLine_Dump]
  [options]

  Control a communications controller. (21.0, EPF)
  Options:
  -ALL
  -Dest_Node_Address (hh-hh-hh-hh-hh | hh-hh) (hex)
  -Dest_Node_Name node-name
  -DEVice (ICS1 | ICS2 | ICS3 | LHC | LTS)
  -Device_Address device-number (octal)
  -No_Query
  -PathName pathname
  -PRotocol protocol

  Ref: Operator's Guide to System Commands [35].

COMOutput [pathname] [-Continue | -Pause | -End | -Ntty | -Tty]

  Control routing of terminal output. (IN)
  Ref: PRIMOS Commands Reference Guide [49].

CONCAT [outputname] [option...]

  Concatenate files. (EX)
  Options:
  -APPend
  -BANner [line]
  -CLOse
  -COMmand (cmd mode)
  -DELETE
  -EJECT
  -HEader
  -INSert (insert mode)
  -JUNit [n] (dfilt=1)

  Insert Mode:
  Prompt char: 
  Enter 1 filename or pathname per line.
  Exit to command mode with a blank or null line.

  Command Mode:
Prompt char: >
* ignores rest of line
exit CONCAT with Quit command
enter 1 command per line:
BANner [line ]  NHEader
DELETE  NREsetp
EJECT  QUIT
HEAder  RESetp
INSert [pathname ]  TITle [title ]
NDElete

Ref: PRIMOS Commands Reference Guide [49].

CONFIG {DATA config-filename }
nusr pagdev comdev [maxpag [altdev
[namlc [nphan [nrusr [smlic ]]]]]]

Configure system. (IN, OP)
The numeric is obsolete as of rev 20.0. Config file directives:

ABBREV YES | NO
    Enables abbreviation expansion.
ALTDEV pdev [records ]
    Specifies the alternate paging device. Obsolete at Rev 21; use PAGING.
AMLBUF line [obufsz [obufsz [dmqsz ]]]
    Sets AMLC buffer sizes.
AMLCLK baudrate
    Sets the baudrate for the programmable AMLC line (4).
AMLIBL [buffer-size ]
    Sets the size of the AMLC input tumble tables.
AMLTIM [ticks [discitime [gracetime ]]]
    Sets time intervals for event timers.
ASRATE ctrl
    Sets the console baud rate.
ASRBUF line [obufsz [obufsz ]]
    Sets the sizes of the console terminal buffers.
ASYNC JUMPER speed5 speed6 speed7
    Set the speeds for the last three available baud rates.
COMDEV pdev
    Indicates the physical disk device to find CMDNC0 on.
COMDVM pdev
    Specifies the disk to use as the mirror for comdev (21.0).
CONFIG nusr pagdev comdev [maxpag [altdev ] [namlc ] [nphan ] [nrusr ] [smlic ] ]
    One line simple configuration.
DISLOG YES | NO | line-num
    Log out users if DTR drops. line-num new at 21.0.
DTRDRP
    Drop DTR on logout.
ERASE [char ] [octal-val]
    Set the system-wide erase character (default is ").
FILTER
    Turn on the network PDN filter (allows connections only from known nodes).
FILUNT rsvunt maxunt [lount ]
    Specifies number of file units. Outdated; do not use.
GO
    End of configuration file.
ICS CARDS device-addr config
    Check async LAC cards in ICS2 or 3.
ICS INPQSZ queue-size
   Set size of ICS input queues.
ICS INTRPT [interrupt-rate]
   Set async interrupt rate for ICS controllers.
ICS JUMPER speed5 speed6 speed7
   Set the speeds for the last three available baud rates. (Obsolete at 21.0; use ASYNC JUMPER.)
KILL [char | octal-val]
   Set the system-wide kill character (default is ‘?‘)
LHC number address
   Assign logical LHC number with physical board address. (21.0)
LOGBAD YES | NO
   Log failed login attempts on the console.
LOGLOG YES | NO
   Allow login-over-login.
LOGMSG YES | NO
   Show all logins on console.
LOGREC sys-logging-value
   Enable system logging. (Obsoleted at 21.0 by DSM)
LOTLIM minutes-to-login
   Set login time limit.
LOUTQM minutes-idle-til-logout
   Set inactivity time limit for automatic logout.
MAXPAG num-pages
   Maximum number of memory pages. (Outdated; do not use.)
MEMHLT [YES | NO]
   Halt on memory ECCU.
MIRROR
   Enables disk mirroring. (21.0)
NAMLCL num-assign-line-buffers
   Allocate assignable amic line buffers.
NET ON
   Start up the network (obsolete as of 19.3).
NETREC net-logging-value
   Number of records to use for net logging. (Obsolete at 21.0)
NLBUF num-locate-buffers
   Configure number of locate buffers.
NPUSR num-phantom-users
   Configure the number of phantom users.
NRUSR number-remote-users
   Configure the number of remote (network) users.
NSEG number-total-virtual-segs
   Set maximum number of virtual segments.
NSLUSR number-slave-users
   Configure the number of NPX slaves.
NTSABF line in Buff-size out Buff-size xoff-lag xon-lag
   Sets buffer sizes and xon/xoff thresholds for NTS assignable lines. (21.0)
NTSACL num-assign-lines
   Reserves buffers for assignable NTS lines.
NTSBUF line in Buff-size out Buff-size xoff-lag xon-lag
   Sets buffer sizes and xon/xoff thresholds for NTS lines. (21.0)
NTSUSR num-users
   Set number of NTS terminal users. (21.0)
NTUSR number-terminal-users
   Configure the number of terminal users.
NUSEG number-user-segs
   Set the number of segments per user (obsolete as of 19.4; use EDIT_PROFILE)
NVMFS number-vmfa-segs
Allocate VMFA segments.

PAGDEVD pdev [records ]
Indicates the disk partition for paging. Obsolete at Rev 21; use PAGING.
PAGING pdevi [...pdevg ]
Specify paging devices. (21.0)
PAGINM pdevi [...pdevg ]
Specify paging device mirrors. (21.0)

PRATIO alt-dev-ratio
Sets the ratio of how often to page to the alternate paging device. Obsolete at Rev 21; use PRATIO command.

PREPAG number-prepage-pages
Specify number of pages to pre-page. (Outdated; do not use.)

REMBUF in-buf-size out-buf-size
Sets the size of buffers to allocate for remote users.

RWLOCK rwlock-value
Sets the system default file read-write lock. (Outdated; do not use.)

SMLC [ON | DSC line strap proc recv | CNTRLR ctrl-num dev-adr | SMLCnn ctrl-num [line-num]]
Turns on the smlc driver. (Obsolescent as of rev 20.0, use SYNC directives.)

SYNC CNTRLR ctrl-num [dev-adr] [protocol]
Enables a sync line with a specified protocol.

SYNC DSC line strap proc recv
Specify data set control.

SYNC ON
Turn on sync line drivers.

SYNC SYMCnn [ctrl-num [line-num]]
Map logical line number to a physical line on a given controller.

SYSNAM system-name
Set the system name.

TPDUMP [YES | NO]
Allow tape dump before abnormal shutdown.

TYPOUT [YES | NO]
Indicates whether to echo config directives on the terminal.

UPS ups-number
Indicates whether an uninterruptable power supply is in use.

VPSD
Write VPSD into memory for debugging. (Obsoleted by Ring 0 debugger.)

WIRMEM
Print out the amount of wired memory.


CONFIG_DSM [options]

Builds and edits the DSM configuration file. (EPF, 21.0, DSM)

Options:
-TTP [TTY | PT45 | PST100 | PT200]
-No_Wait
-Helper [No_Wait]
-USA GE


CONFIG_NET [pathname ] [-Help] [-TTP terminal-type]

Network configurator. (EX)
See the Network Planning and Administration Guide, [27].
CONFIG NTS [config-pathname ] [options ]

Configure Network Terminal Support (NTS). (EPF)
Options:
- CReate
- DiSPlay
- No_Wait
- Terminal_TyPe {PT45 | PT200 | PST100 | TTY}
- EDit
- Listing [pathname ]
- SPOOL [spool-options ]
- LANGUAGE language

CONFIG UM [selection-name ] subcommand

Configures DSM unsolicited message handling on a system. (EPF, 21.0, DSM)
Subcommands:
- SELect [-ON node ]
- MODIFY [-ON node ]
- CANcel [-ON node ]
- LIST [-ON node ] [-No_Wait]
- Help [-No_Wait]
- USAGE [-No_Wait]

TOOLS> CONVERT_AMLC_COMMANDS [input_file output_file ]
- HELP | -INTERACTIVE

Convert AMLC commands to SET_ASYNC commands. (CPL)

COPY pathname [new-pathname ]
[-Copy_All | -DTM | -PROTect | -QUOTA | -RWLock | -Save_UFD]
[-DAM | -SAM | -CAM | -DeLete | -INCremental | -REPLACE]
[-FORCE] [-MERGE] [-ADD] [-MXL] [-NO_CMLV] [-NO_CHECk]
[-LeVeLS [n]] [-No_Query | -Query] [-RePorT] [-DEBUG]

Disk to disk copy utility. (EPF)
Ref: PRIMOS Commands Reference Guide [49].

COPY_DISK [-DO_VERIFY ] [-NO_BADS ] [-TTY ]
[-NO_RAT] [-NO_CHECkSUM]

Copy disk. (EX, OP)
Ref: Operator's Guide to System Backups [34].

COPY_RBF source-pathname dest-pathname [-DeLete] [-PROtect] [-DAM]
[-CAM] [-Min_eXt_Len ] [-RePorT]

Copy an RBF file. (EX)
Ref: ROAM Administrator's Guide [53].

CPL filename
Execute a CPL file. (IN)
See section 2.5. Ref: CPL User's Guide [6].

CPMPC pathname [-PRINT] [-CRn] \(n=0,1\)

Punch file on card punch. (EX)
Ref: PRIMOS Commands Reference Guide [49].

CRASH_AUDIT -MT n -DUMPFILE pathname -OUTFILE pathname
-MAP pathname

Completes a partially written security audit after a system halt. (EX).

CREate uidname [-PassWord] [-CAT acat ] [-MAX n ]

Create subUFD in current UFD. (IN)
Ref: PRIMOS Commands Reference Guide [49].

CREATK

Build multikeyed index files. (EX)

CRMPC pathname [-PRINT] [-CRn] \(n=0,1\)

Read cards. (EX)
Ref: PRIMOS Commands Reference Guide [49].

CRSER pathname

Read from serial card reader. (EX)
Ref: PRIMOS Commands Reference Guide [49].

CSUBS subschema-source [-Output pathname] [-List Pathname]

Invoke COBOL DBMS subschema. (EX, OBS)


Print date and time. (IN, LO)
Ref: PRIMOS Commands Reference Guide [49].

DBACP

Data Base Administrator Command Processor. (EX)
Subcommands:

ALLOCate {FILES | KEYS} [OF] [SCHEMA] schema

ALLOCate KEY [OF] [LOCK] lock
[OF SUBSCHEMA schema]

ALLOW [AI-RECoVery | BI-RECoVery | TRANS-ROLLback | MULTIUSERS]
[OF] [SCHEMA] schema

CHANGE KEY [OF] [LOCK] lock
[OF] [SCHEMA] schema

2-28 Prime Restricted
CHANGE KEYS [OF] [SCHEMA] schema
CLEAR FILES [OF] [SCHEMA] schema
CLEAR LISTing [filename]
DELETE {FILES | KEYS} [OF] [SCHEMA] schema
DELETE KEY [OF] [LOCK] lock [OF] [SCHEMA] schema
DELETE [SCHEMA] schema
DELETE SUBschema {ss-name | ss-num} [OF] [SCHEMA] schema
DELETE SUBschemaS [OF] [SCHEMA] schema
DiSallow [AI-RECoVery | BI-RECoVery | TRANS-ROLLback | MULTIUSERS] [OF] [SCHEMA] schema
EXPAND {AREA | CALC [OF] [RECORD] | SET} object-name [OF] [SCHEMA] schema
EXPAND FILES [OF] [SCHEMA] schema
LOCK [SCHEMA] schema
MOVE {AREA | CALC [OF] [RECORD] | SET} object-name [OF] [SCHEMA] schema
PACK {AREA | CALC [OF] [RECORD] | object-name} [OF] [SCHEMA] schema
RENAME [SCHEMA] schema
SAVE LISTing [filename]
UNLOCK [SCHEMA] schema
VERIFY {AREA | CALC [OF] [RECORD] | KEY [OF] [LOCK] | SET} object-name [OF] [SCHEMA] schema
VERIFY {AREAS | CALCS | FILES | KEYS | SETS | FILES | SUBSchemas}
VERIFY SCHEMAS [SCHEMA] schema
VERIFY SUBschema {ss-name | ss-num} [OF] [SCHEMA] schema
VERIFY SUBschemaS [OF] [SCHEMA] schema
Ref: DBMS Administrator's Guide [8].

DBASIC [pathname]

Double Precision Arithmetic BASIC. (EX)
Ref: Interpretive BASIC User's Guide [20].

DBG filename [-C0minput [-No_C0minput]
[-VeriFY_Proc | -No_VeriFY_Proc]
[-No_VeriFY_Symbols | -VeriFY_Symbols]
[-Load_ State pathname] [-FCN]
[-Full_Initialize | -Quick_Initialize]

Source level debugger. (EX)
Subcommands:

primos-command-line
     pass primos-command-line to the PRIMOS command processor.
* [value]
     execute command line value times or until an error occurs.
:([language-name [print-mode ]] expression | print-mode expression)
evaluate expression. language-name either PL1, PASCAL, CoBoL, vRPG, Cc, MODula-2(MOD), or FORTRAN, print-mode either Ascii, Bit, Decimal, Float, Hex, or Octal.
ActionList Suppress | Print
Control printing of action lists.
Again
Repeat last command.
ARGumentS [program-bik-name [act-num] | alt-entry-id]
display value of all arguments to specified program block.
[-IGNore | -NiGNore] [-EDIT] set and modify breakpoints.
CALL variable [ [arg-list]]
call a subroutine or function from the debugger command level.
CLEaR [brkpt-id ]
clear a breakpoint or tracepoint.
CLEaRALL [prog-bik-name [-DeSCend]] [-BRK | -TRA]
clear all breakpoints or tracepoints in the debugging environment or all breakpoints in a specified program block.
CmdLine
enter program command line arguments.
Continue
continue program execution following a breakpoint, condition signal, or single step operation.
ENVironment [prog-bik-name [act-num]] | -POP]
define the evaluation environment.
EnvList
print current evaluation environment and contents of the evaluation stack.
ETrace [ON | ARGS | OFF]
enable and disable entry and exit tracing.
GOTO [prog-bik-name [act-num]] [statement-id]
modify value of execution pointer and transfer control to a specific statement when execution resumes.
HELP [-LIST | -SYM_LIST | command_name | syntax_symbol ]
print debugger syntax.
IF expression act-list [ELSE act-list]
conditionally execute debugger commands.
IN continue program execution until next procedure is called.
IniCmdLine
enter initialization routine command line arguments (registered EPFs only).
INFO prog-bik-name | alt-entry-id | statement-id
print information about program block, alternate entry to a program block, or statement.
Ini_LIbrary
- initializes an EPF library.
Ini_LINKage
- initializes the EPF's linkage.
IPSD
enter IPSD.
LANGuage [PLI | ForTraN | PLP | PAScAL | F77 | CoBoL | vRPG | Cc | MODula-2]
specify language for expression evaluation.
LET variable = expression
assign new variable to a variable defined by the program.
LIST [brkpt-id ]
print attributes of one breakpoint or tracepoint.
LISTALL [prog-bik-name [-DSC]] [-BRK | -TRA]
print list of breakpoints and tracepoints.
LoadState filename
  restore DBG state contained in filename.
MACro [macro-name [command-list | -DeLete | -EDit | -Change_Name old-macro-name
  new-macro-name] | -ON] | -OFF
  control macro definition and execution.
MacroList [macro-name]
  list macro-name and associated command list or list all macros.
MAIN [prog-blk-name]
  define procedure called by REStART or print name of main program.
OUT continue execution until program block specified by execution environment pointer
  returns.
PAuse
  temporarily suspend debugging session and return to PRIMOS command level.
PMode print-mode var_1 [var_2...]
  set print mode of a variable.
PSD
  enter IPSD (rev 22) or VPSD (< rev 22).
PSYmbol
  print table containing names of special symbols and current character values.
Quit exit to PRIMOS command level and terminate debugging session.
REStArt [step-command]
  start or restart execution of program. step-command is either STEP, STEPin, or IN
  command line.
RESubmit
  edit and resubmit last command line entered.
SaveState filename [-MACros] [-BReakpoints] [-TRAcepoints]
  save state of DBG session to filename.
SEGments
  print list of segments in use.
Source source-command [arg]
  examine debug source files. edit-command can be Top, Bottom, BRIef, Verify, Print,
  PPrint, Where, POint, Next, MODE, Locate, Find, Symbol, PSymbol, *, EX, and NAMe.
STATUS
  print status information.
Step [value]
  resume program execution for value number of statements. Do not include statements
  within called procedures.
StepIn [value]
  resume program execution for value number of statements. Include statements within
  called procedures.
STrace [Full | Quiet | OFF]
  enable or disable statement tracing.
SYMbol symbol-name char-val
  set value of DBG character symbol.
TraceBack [-FROMR value [-LR]] [-F value] [-TO value] [-REV] [-DBG] [-OLN] [-ADR]
  print call/return and ownership information contained in stack frames.
  [-COunt value]
  set tracepoint.
TYPE expression
  evaluate expression and print attributes of result.
UnWatch [var_1 [var_2...]] [-ALL]
  remove variable(s) from watch list.
UNWIND
  release user program and debugger from procedure call; unwind stack and undefined
  execution pointer.

Prime Restricted
vPSD
   enter VPSD (< rev 22) or IPSD (rev 22).
VTrace [Full | Entry_Exit | OFF]
   enable or disable value tracing.
WAthch var, [var2,...]
   add variable(s) to watch list and enable value tracing.
WatchList
   print names of variables currently on watch list.
WHere [segno/wordno]
   print program location or value of execution pointer.
DBG edit subcommands allowed with RESUBMIT, BREAKPOINT, and MACRO:
   D delete character.
   F specify first character.
   L specify last character.
   A text append text to end of line.
   I text insert text following character under
      which "I" is positioned.
   O text overlay text beginning with character
      under which "O" is positioned.
   Q return to DBG command level.
DBG internal variables:
   $COUNT
   $COUNTERS
   $MR
   $RTN_FUNCTION_PTR
   $RTN_FUNCTION_STRUCTURE

Ref: Source Level Debugger User's Guide [57].
DBUTL
Data Base dump UTility. (EX)
Subcommands:
ADir [area-name | area-num]
Area [area-name | area-num] [-History]
BRief
Bucket start [end] [Octal | Asci] [Continue]
DBK [area-num rec-num occ bucket | int1 int2 int3]
Dump CALc [rec-name | rec-num] [-History]
Dump SHared
EDit [A | R | S | SC] [ent-name | ent-num]
subcommands:
DEN entry-num
FILE
Next [-] num-lines
POint line-num
Print num-lines [A | D | O]
Quit
Replace data
FIX [A | S] [ent-name | ent-num]
Help [command]
ID [A | D | I | L | R | S] [ent-name | ent-num]
LIST [rec-name | rec-num] [item-name | item-num]
List list-num
MON
NBrief
NEnt entry-point
Node [node-num | [L | P | R | S | = | >]
num ["][] [A | O | D]
ODir occ
Output [filename] [TTY]
Quit
RAM schema-name
RDDir [rec-name | rec-num]
Record rec-num occ bucket [Octal | Asci]
[Continue]
REWInd [A | R | S | SC] [ent-name | ent-num]
ROAM schema-name
Schema schema-name [-History]
SDir [set-name | set-num]
SET [set-name | set-num] [-History]
SST [rec-name | rec-num] [set-name | set-num]
VERify interval
WHere
Ref: DBMS Administrator's Guide [8].

DEFine_GVar [pathname [-CREATE] | -OFF]
Define global variable file. (IN)
Ref: PRIMOS Commands Reference Guide [49].

DELAY [min [max [width]]]
Set terminal delay characteristics. (IN)
Can issue prior to login. Defaults are: 6, 12, 72. Ref: PRIMOS Commands Reference Guide [49].
DELETE pathname [-No_Query | -Query] [-FORCE] [-RePorT] [-DEBUG]

Delete files or directories. (EPF)
DELETE supports the wildcard convention. Ref: PRIMOS Commands Reference Guide [49].

DELETE_RBF pathname [-No_Query] [-RePorT]

Delete an RBF file. (EX) Ref: ROAM Administrator's Guide [53].

DELETE_VAR id_1 [...id_n]

Delete global variables. (IN)
Ref: PRIMOS Commands Reference Guide [49].

DELSeg {segno [-TO segno ] | ALL}

Delete segment(s). (IN)
segno > 4000 Ref: PRIMOS Commands Reference Guide [49].

DENOTE

DEsign NOTEbook generator (SPS) (EX, QT).

DEREMER pathname [-GRaMmar] [-FSA] [-DEBUG] [-EXTernals]
[-EXTERNALS] [-CC | -PL1 | -PL1G | -SPL | -PLP]
[-No_PARser] [-No_ACTIONS] [-No_SR_Conflicts]
[-NOERRTTY]

Parser generator. (EX, QT)
Ref: PE-T 535 [38].

DEVice_ACLs { -ON | -OFF}

Enables/disables device ACLs in DEVICE*. (IN)

DIAG

Diagnostic utility for PRISAM. (EX)
Ref: PRISAM User’s Guide [50].

DISCOVER [options]

DBMS and PRISAM query/update/report generation tool. (EPF)
Options:
-TEST
  Don’t abort coml file on error (NR)
-TIME
  Produce timing messages after each command (NR)

-INITialize tree
  Initialize shared data structures from data in tree (default
  SYSTEM>DISCOVER.CONFIG) (20.2, 21.0) (ND)
-CLEanUP
   Clean up after abort (usually not needed)

-Edit_Cmd_Line
   Allow ECL-style command line editing (22.0)


DISCOVER_TCB

Generate Terminal Control Blocks for DISCOVER screen interface. (EX)

Disks [NOT] pdev_0 [...pdev_7]

Specify assignable disks. (IN, OP)
Ref: Operator's Guide to System Commands [35].

DISPLAY_LOG logname [options]

Displays messages from a DSM log. (EPF, 21.0)

Options:
-Private_LOG
-System_LOG [node-ID]
-out-file [-No_Query]
-Format [BRIEF | FULL | format-name]
-NOHeader
-CENSUS
-No_Wait
-PRODUCT products
-MeSSage_ID message-types
-NODE nodenames
-USER usernames
-SEVerity severities
-Logged_AFter [date/time]
-Logged_BeFore [date/time]
-REMARK text
-Help [-No_Wait]
-USAGE


DISTRIBUTE_DSM [options]

Distributes DSM configurations. (EPF, 21.0)

Options:
-TPP [TTY | PT45 | PST100 | PT200]
-No_Wait
-HELP [-No_Wait]
-USAGE


DLGEN

Generate a downline load file (.DL) from .DDL files. (EX, QT)
DMSTK

See DUMPSTACK.

DPTCFG config-pathname [-O outpathname]
Configure file for DPTX. (EX, OP)
Config file commands can be:
DEFINE GROUP n options (n=1,32)
  Options:
  -PROTOCOL SP3270
     EM3270
  -LINE n (where n=0,1)
  -ADDRESS nn (where nn=2-digit hex)
  -DEVICE n1 n2
     (n1 < n2)
DEFINE DEVICE n options (n=1,32)
  Options:
  -NAME 32-char-name
  -ADDRESS nn (nn=2-digit hex)
  -ENABLE [COMMAND],[BLOCK],[WRITE],[READ]
  -USER n
  -PRINTER [VFC] [PLATEN nn]

Ref: Distributed Processing Terminal Executive Guide [12].

DPTX {-ON | -DATA pathname | -OFF}
Enable DPTX terminals. (EX)
Ref: Distributed Processing Terminal Executive Guide [12].

DPTXMTR [-TOTS] [-FREQuency min]
DPTXMTR -QUEUE [-FREQuency sec]

DPTX communications line monitor. (IN)
Ref: Distributed Processing Terminal Executive Guide [12].

DROPDTR

Force dropping of Data Terminal Ready. (IN, LO only)

DuMP_Segement[ [segment1...segment10] [-Range start-segment end-segment] | -HELP]

Specify user segments to be dumped for partial tape dump. (IN, OP)
Ref: Operator's Guide to System Commands [35].

DuMP_STacK [-ALL | -BRIef | -FRamesn | -FROMn | -ON_Units]

Trace user command stack. (IN)
Ref: PRIMOS Commands Reference Guide [49].

DuMP_User {username1 [ ...username10]} | -HELP

Select which users will have their segments dumped to a partial tape dump. (IN, OP)
Ref: Operator's Guide to System Commands [35].

ED [filename]
Editor. (EX) No filename => new file. (str - text string) (/ = unique delimiter not in string)
Subcommands:

.CR. = INPUT TTY
Append str
Append to current line.
Bottom
Go to bottom of file.
BRief
Don’t display changes.
Change/str1/str2[/][n][G]
Change str1 to str2 for first occurrence on line, for all occurrences if G present, for n
lines if n present.
Delete [n]
Delete n (1) lines.
Delete TO str
Delete to line containing str.
DUnload fname [n]
Unload/delete n (1) lines.
DUnload fname TO str
Unload/delete up to (not incl) line containing str to fname.
Erase char
Make char the erase character (').
FILE [fname]
Write updated file to fname.
Find([column]) str
Find line with str starting in column.
Gmodify subcommands
Modify line with subcommands:

A/str - Append
Bn - Back n chars
Cc - Copy up to (not inc) char c
Dc - Delete up to char c
En - Delete next n chars
F - Copy to end of line
I/str - Insert str at curr pos.
Mn - Copy n chars
Nxx - Negate criteria of cmnd xx
O/str - Overlay at current position
R/str - Retype at current position
S - Reset to start of line

INPUT] (ASR) | (PRT) | (TTY)
Input text from specified device.
Insert str
Insert line (.NULL. => input mode).
Kill char
Make char new kill character.
Linesz n
Set max line size to n chars.
LOAD fname
Insert contents of fname.
Locate str
Locate line containing str.
MODE arg
Set editor mode. arg can be:
PRUPPER, PRALL, PRLOWER,
PROMPT, NPROMPT,
COUNT, NCOUNT,
NUMBER, NNUMBER,
COLUMN, NCOLUMN.

Modify /str;/str;[G][n]
Copy str; on top of str, starting with first char.
MOVE buf1, buf2
Move contents of buf2 to buf1.
MOVE buf1 str
Move contents of str to buf1. Buffers are EDLIN (command line), INLIN (current line to be edited), STR.1, ..., STR.10.

Next [n]
Advance n (1) lines.
NFind str
Find line not starting with str.
OUTPUT [DISPLAY |TTY]
Send verification output to specified device.
Overlay str
Overlay line with str. Blank leaves current char, WILD becomes blank.
PAuse
Back to PRIMOS, restart with 'S'.
POint n
Go to line n.
Print [n]
Print n (1) lines
PSymbol
Print symbols.
PTabset
Use physical tabs of terminal when printing.
Punch [n] [ASR | PTP]
Punch n lines on indicated device.
Quit Exit without filing.
Retype str
Replace line with str.
SAVE
???
Symbol name char
Define name symbol. name: BLANK (#), CPROMPT ($), COUNTER ( ), DPROMPT ($), ERASE (;), ESCAPE ( ), KILL ( ?), SEMICO ( ;), TAB ( ), WILD ( ).
TAbset tab1...
Set tab positions.
Top Go to top of file.
Unload from [n]
Unload n line into from.
Unload from TO str
Unload lines up to (but not incl) str to from.
Verify
Display all changed lines.
Where
Print current line number.
Xeq buff
Execute contents of buffer.
* [n] Repeat n (until bottom or forever) times.
Ref: New User's Guide to EDITOR and RUNOFF [28]
EDB \{(inpathname \| -Ptr \| -ASR) [outpathname \| -Ptr \| -ASR]\}

Binary editor. (EX)

**BRIEF**
No names printed.

**Copy \{(name \| ALL \| <SFL> \| <RFL>\)**
Copies up to (but not incl) specified point.

**ET**
Copies an EOT (end-of-tape) mark to the output file. OBSOLETE.

**Find \{(name \| ALL \| <SFL> \| <RFL>\)**
Position to object.

**GENET [G]**
Copy current routine and then write out an EOT. (G) indicates copying all files, each with an EOT appended. OBSOLETE.

**Insert pathname**
Insert pathname into the output file.

**Newftp pathname**
Open new input file after closing old one.

**Omit \{G\**
Copy current routine to output file, omitting any EOT. (G) causes this to occur for all routines. OBSOLETE.

**OPEN pathname**
Open new output file after closing old one.

**Quit**
Close all files and exit to PRIMOS.

**Replac \{fname\} pathname**
Replace fname with pathname.

**RFL**
Insert Reset Force Load flag.

**SFL**
Insert Set Force Load flag.

**TERSE**
Print 1st name in blocks.

**Top**
Top of input file.

**VERIFY**
Print all names.


**EDit_ACcess target acl [-No_Query]**

Modify existing access control list. (IN)
Ref: PRIMOS Commands Reference Guide [49].

**Edit_Command_line options**

EMACS like command line editor. (21.0)
Options:

-ON
-OFF
-\{-CASE\}_search
-\{-Clean\}_COMO
-\{-COL\}_major
-\{-COMP\}_onent
-\{-Edit\}_COMI
-ENTRY
-\{-Error_Brief \{'text\}'
-Help
-\{-INIT\}ialize
EDIT_EFU [pathname] [-DispLay] [-HELP [internal-cmd]]

Edit an SNA printer form. (EX)
Ref: PRIME/SNA Operator's Guide [56].

EDIT_PROFILE [pathname] [-PROJect project-id]

Users' system profile editor. (EX)
Subcommands:

Add_Project [project_id [-PA pa_name] [-CREate_pa] [-LIKE like_reference] [-PROFile]
[-SIZE entry_count] [-NO_Query]]
Add_User [user_id [-LIKE like_reference] [-NO_Query] [-PROJect | -DefaulT project_id]]
[-PROFile] [-PassWord [password]] [-SYStem] [-Verify_NS]
ATTach_Project [project_id]
Change_Project [project_id [-PROFie] [-LIST] [-SIZE entry_count]] [-PA [pa_name]] [-LIMits]
Change_System_Administrator [sa_name] [-ALL]
Change_System_Defaults [-Dynamic_Segments number] [-Static_Segments number] [-LEVelS number] [-PROGrams number]
Change_User [user_id [-PROJect project_id]] [-LIST] [-PassWord [password]] [-SYStem]
Delete_Project [project_id]
Delete_User [user_id [-PROJect project_id]]
DeTach_Project [project_id]
Force_Password [-ON | -OFF]
HELP [command_name]
[-ALL]]
[-DET ail]
List_User [user_id [-PROJect project_id] [-ALL]]
Minimum_PassWord_Len_gth length
No_Null_Password [-ON | -OFF]
Quit
REbuild [-PROJect project_id] [-SIZE entry_count]
System Defaults [-ON | -OFF]
Verify_User [user_id] | -ALL


Eligts tenths

Change user eligibility timeslice. (IN)
Ref: Operator's Guide to System Commands [35].

EMACS [filename] [options]

EMACS screen editor. (EX) Options can be:
- -Terminal_Type terminal_type
- -SPEED, -BPS bps
- -Height num_lines
- -Width num_columns
- -MOXOFF | -MOFF
- -DEBUG
- -LIBRARY
- -INITIALIZE_MACH, -IX {BUILDING | treename}
- -ECHO_CPL
- -HELP
- -SUI | -SUX
- -User_LIBRARY pathname
- -No_User_LIBRARY

Ref: EMACS Reference Guide [14]

EVENT_LOG [-NET] [-ON | -OFF]

Enable/disable event logging for system or network. (EPF, OP, OBS)

Expand_Search_Rules [object] [options]

Returns the full pathname of an object found by a search rule. (IN)
Options:
- -Access_CATegory
- -DIRectory
- -FILE
- -REFERencing_DIR pathname
- -SuFicX suffix
- -DIRectory
- -List_NAME listname
- -SEGment_DIRectory
- -Help


F77 pathname [CE-option...]

Invoke FORTRAN 77 compiler. (EX)
See Compiler options, 2.7.1. Ref: FORTRAN 77 Reference Guide [15].

F77DML [input-pathname [output-pathname] [error-pathname] | options]
F77 data manipulation language processor. (EX)
Options:
-Input pathname
-OUTPUT pathname
-ERROR pathname
-Dynamic
-No_Line_Number
Ref: DBMS Data Manipulation Language Reference Guide.

F77SUBS source [-Output pathname] [-List pathname]
F77 DBMS subschema processor. (EX)
Ref: DBMS Data Manipulation Language Reference Guide.

FAP
FORMS administrative processor. (EX)
Ref: FORMS Programmer's Guide [17].

FAU
File access utility for PRISAM. (EX) Ref: PRISAM User's Guide [50].

FDL pathname [options]
FORMS definitive language. (EX)
Options:
-INPUT {pathname | TTY}
-BINARY {pathname | YES | NO}
-LISTING {pathname | YES | NO | TTY | SPOOL}
-ERRLIST
-ERRTERM
-EXPLIST
-IOFLIST
-MACLIST
-OBJLIST
-REPLIST
Ref: FORMS Programmer's Guide [17].

FDML [input-pathname [output-name] [error-pathname] | options]
FORTRAN DML preprocessor. (EX)
Options:
-Input pathname
-OUTPUT pathname
-ERROR pathname
-Dynamic
-No_Line_Number
Ref: DBMS Data Manipulation Reference Reference Guide.

FED [-PROFILE pathname]
Forms EDitor. (EX)
Ref: FED User's Guide [16].
FILMEM [ALL]

Zero memory. (EX)
Note: 'ALL' => '100 - 77777 excluding PRIMOS II zeroed. Ref: PRIMOS Commands Reference Guide [49].

FILVER [pathname1 [pathname2]]

Compare binary files. (EX)
Prompts if no names entered. Ref: PRIMOS Commands Reference Guide [49].

FIND_RING_BREAK [-Help] [-Input filename]

Find ring break in local network. (EX)
Ref: PRIMENET Guide [45].

BATCHQ>FIXBAT [-DAYS n] [-QUIET] [-STARTUP {SAVE | SPOOL | DELETE | NOLOG}]

Check and fix the batch queue database integrity. (OP)
Ref Operator's Guide to the Batch Subsystem [31].

FIXRAT [options]

Fix record availability table for rev 18 disks. (EX, OBS)
Prompts:

FIX DISK? (Enter Yes or No.)
UFD COMPRESSION. (Enter Yes or No.)
PHYSICAL DISK= (Enter pdev—see Disk Addresses.)

If 'OPTIONS' specified:

TYPE DIRECTORIES TO LEVEL (Enter level.)
AUTO TRUNCATE DIRECTORIES NESTED TOO DEEPLY
(Enter Yes or No.)
TYPE FILENAMES (Enter Yes or No.)
TYPE FILE CHAINS (Enter Yes for disk addr.)

WARNING: Do not use on Rev. 19.0 or later disks! Use FIX_DISK.

FIX_DISK [-DISK | -PDEV] pdev [options]

Disk maintenance utility. (EX)

-FIX Fix the disk
-ufd_ComPRession Compress the disk
-LEVEL dec Lowest level in which UFD names are printed
-MAX_nested_level dec The max depth that UFDs are allowed to be nested
-Auto_Truncation Automatically truncates UFDs nested too deeply
-List_File The file names are printed
-No_Qquota  The partition is not a quota partition

-COMMAND_DEVIce  The disk being fixed is the command disk

-CONVERT_19  Convert the disk to a rev 19 style disk

-CONVERT_20  Convert the disk to a rev 20 style disk

-CONVERT_21  Convert the disk to a rev 21 style disk

-DUFE  Delete all unknown file entries (default)

-SUFE  Save all unknown file entries

-INTERactive  Interactively fix the DSKRAT

-List_BadSpots  List badspots and remapping records

-TRUncate  Truncate the file on error

-ADD_BADSpot oct[, oct,]
  Add badspot(s) to disk

-number_of_retries dec
  Modify the number of retries

-numrty dec  Modify the number of retries

-FAST  Perform fix checking last two data records

-CHECK  Determine if partition has been shutdown properly

-All_Controller  Convert a rev 21 disk to be used on all controllers

-Intelligent_Controller  Convert a rev 21 disk to be used only on intelligent disk controllers

-Dump_DBS  Dump the DBS file of a rev 21 disk

-Disk_Type  Specify the disk model type for a -CONVERT_21 option

-Override_Default_Interleave
  Override default allocation interleave

-Restore_Default_Interleave
  Restore default allocation interleave

-MINimum_extent_SIZE
  New value for the minimum extent size on partition

-MAXimum_extent_SIZE
  New value for the maximum extent size on partition

Ref: Operator's Guide to File System Maintenance [32].

FSUBS source-filename [-Output pathname ] [-List pathname ]
Invoke FORTRAN DBMS subschema. (EX)
Ref: DBMS Data Description Language Reference Guide [9].

FTGEN

FTS system administrator utility. (EX)
Ref: Network Planning and Administration Guide [27].

FTN [-Input] pathname [options ]

FORTRAN-66 compiler. (EX)
Input/output options can be:

-Binary [pathname | YES | NO] Specify binary file creation.
-ERRlist Generate errors-only listing.
-ERRty Print errors on terminal.
-Explist Generate expanded listing.
-Listing [pathname | NO | SPOOL | TTY | YES] Specify listing file creation.
-NORerrty Don't print errors on terminal.
-NORxref Do not generate a cross reference listing.
-XREFL Produce a cross reference, implies -L YES.
-XrefA Produce an abbreviated cross reference listing.

Memory usage options can be:

-BIG Handle arrays larger than one segment.
-Debase Conserve loader base areas.
-Dynm Dynamically allocate storage for local variables.
-Fp Generate floating point skip instruction.
-FRN Floating round before store.
-INTS Assume integers are INTEGER*2.
-Inll Assume integers are INTEGER*4.
-NOBIG Don't allow arrays to span segment boundaries.
-NOFP Don't generate floating point skip instruction.
-NOFRN Don't perform floating round before store.
-Pbexb Allocate entry control block in procedure frame.
-Save Statically allocate storage for local variables.
-64v Generate 64V-mode code.
-64R Generate 64R-mode object code.
-32r Generate 32R-mode code.

Compiler operation options can be:

-Opt Perform conservative do-loop optimization.
-STopt Perform standard optimizations.
-UncOpt Perform unconditional do-loop optimization.

Debugging options can be:

-DCLVAR Flag all undeclared variables.
-DEBUG Allow full use of DBG.
-NODclvar Don't flag undeclared variables.
-NODEBUG Allow no use of DBG.
-Notrace Don't generate code for trace output.
-PRODUCTION Allow production mode debugging.
-Spo System programmer option.
-Trace Generate code for trace output.

Device codes for Input, Listing, Binary:
FTOP [-HELP [subject] | server-option | manager-option]

File transfer service operator command. (EX, OP)

Server-option:
-Abnd_SrVr server-name
-Abort_SrVr Link server-name link-number
-List_SrVr_Sts [server-name]
-Start_SrVr server-name
-StoP_SrVr server-name

Manager-option:
-Start_MnGr [manager-name]
-StoP_MnGr

Ref: PRIMENET Guide [45].

FTR [source-file [dest-file] [-Dstn_Site sitename]
[-Dstn_User username] [-DEVice dev-name] [-HOLD]
[-LOG pathname] [-NAme ext-name]
[-Src_Site sitename] [-Src_User username]
[-ABORT | -CANCEL | -DISPLAY | -HOLD | -RELEASE]
requestname
-STATUS [requestname]
-MODIFY requestname [FTR-options]

File Transfer Request. (EX)
Ref: PRIMENET Guide [45].

FUTIL [-NORM]

File system utility. (EX)
Subcommands:
-Attach pathname (** => home ufd)
CLEAN prefix [level]
Copy file [newname] [file [newname]]...
COPYDum file [newname] [file [newname]]...
COPYSyM file [newname] [file [newname]]...
CREATE ufdname [owner [nonowner]]
DELETE file [file]...
FORCE ON or OFF
List [level] [First] [LISTFIL] [PROtect] [Size]
[Type] [Date] [RWlock] [PAsswd]
LISTSave filename [options as for Listf]
Protect file [owner [nonowner]]
Quit
Scan file [options as for Listf]
SRWloc file lockno
To pathname
TRECpy ufd [newname] [ufdname [newname]]...
TREAD ufdname [ufdname]...
TREP pro uidname [owner [nonowner]]
TRESrw uidname lockno
UFDCpy
UFDEDEL
UFDP pro [owner [non-owner]]
UFDSrw lockno level
  lockno:
  0 use system read/write lock (SYS)
  1 n readers or 1 writer (W/NR)
  2 n readers and 1 writer (1WNR)
  3 n readers and n writer (NNWR)

Ref: PRIMOS Commands Reference Guide [49].

GENERATE_CATALOG -MT n [options ]

Generate/validate a BRMS tape catalog. (EX)
Options:
  -CATalog_PAtchname pathname
  -No_Query
  -OWNer user-id
  -REEL n
  -VA Lidate
  -VOLID volume-id
  -HELP [subject]

Ref: Operators Guide to System Backups [34].

HDXSTAT

Display status of half duplex network. (EX)
Ref: PRIMENET Guide [45].

HELP [command-name | topic-name ]


HISTORY

Generate program history (SPS). (EX, QT)

HP SD

High PSD. (EX)
SA, EA = 147760, 156552. Start of initial P counter = 150000. For internal commands, see PSD.

IDBMS [-CONFIG]

Initialize DBMS. (EX, OP).
Ref: DBMS Administrator's Guide [8].

INFO

Enter INFORMATION. (EX)

INFORM
INstruction FORMatter for PLP programs (SPS). (EX, QT)

BATCHQ>INIT [-ReSeT_Queue] [-ADMINistrator user]

Initialize the BATCH data base. (EX, OP)
The -ADMIN option may be used several times on the command line. Ref: Operator's Guide to the Batch Subsystem [31].

Initialize_Command_Environment

Reinitializes user command environment. (IN, 19.4)
Ref: Programmer's Guide to BIND and EPFs [51].

Input pathname

Open file unit 1 for input. (IN)
Ref: PRIMOS Commands Reference Guide [49].

IPSD, IPSD0, IPSD16

Enter UNIX mode symbolic debugger. See PSD for commands.

IROAM [-COLDSTART]

Unwind incomplete transactions and initialize ROAM shared memory. (EX)
User 1 or .ROAM_ADMIN group only. Ref: ROAM Administrator's Guide [53].

JOB [pathname ] | [job-id ] [option ]

Submit batch job. (EX)
Monitor job options can be:
- STATUS
- DISPLAY

Job control options can be:
- ABORT
- CANCEL
- CHANGE

Operator options can be:
- ABORT
- CANCEL
- HOLD
- RELEASE
- RESTART

Submit job options can be:
- ACCT information
- ARGS cplargs
- CPL
- CPTIME seconds
  NONE
- ETIME minutes
  NONE
- FUNIT number
- HOME pathname
- PRIORITY value
-QUEUE queuename
-RESTART YES
 NO


KBUILD

Build keyed-index file. (EX)

KDEL

Delete records in keyed-index file. (EX)

LABEL MTn [options]

Create magnetic tape label. (EX)
Options:
- TYPE A | B | E
  -VOLUME | -VOLId | -VOLser vol-id (1-6 chars)
  -OWNER owner (1-14 chars)
  -ACCESS access (1 char)
  -HELP
  -INIT
  -OVERWRITE

Types are A (ASCII - ANSI), B (BCD - IBM) or E (EBCDIC - IBM). Ref: Magnetic Tape User's Guide [24].

LATE

Defer command execution. (EX) Prompts for time in the form HHMM. Ref: PRIMOS Commands Reference Guide [49].

LD [pathname] [options]

List file characteristics. (EX)
Options:
- BRIef
- CATegory Protected [acat-name]
- DeFaulT Protected
- DETail
- DTA
- DTB
- DTC
- DTM
- HELP
- No_Column_Headers
- No_HEader
- No_SORT
- No_Wait
- PROtect
- ReVerse
- SinGLE_COlumn
-SIZE (uses 1K record size)
-SORT_dIA
-SORT_dIB
-SORT_dIC
-Sort_Dtm, -SORTM
-SORT_Name, -SRTNM
-SPECific_Protected
-WIDE

and wildcard options. Ref: PRIMOS Commands Reference Guide [49].

LEM \{rbuf-filename>subfile-number | BFILE\}

List extent map of a CAM file. (EX)
Ref: ROAM Administrator's Guide [53].

LISP \[-INPUT_FILE source pathname \]
\[-OUTPUT_FILE output pathname \]
\[-ERROR OUT error pathname \]
\[-DYNAMIC number_of_segments -RESERVED number_of_segments \]

Invokes the Prime Common LISP Interpreter/Compiler (EPF).

Listf

List files in current UFD. (IN, 19.4-CPL/EPF)
Ref: PRIMOS Commands Reference Guide [49].

Listing pathname

Open file unit 2 for listing output. (IN, 19.4-CPL)
Ref: PRIMOS Commands Reference Guide [49].

List_ACcess \{object \}

List access rights. (IN)

LIST_ASSIGNED_DEVICES \{device-names | -USER \{user-names | user-numbers \} \}
\{general SIM options \}

Lists assigned devices on the system. (EPF, 21.0)

General SIM options:
-Help [-No_Wait]
-USAGE
-ON node | nodegroup
-Private_LOG pathname [-Ntty]
-System_LOG pathname [-Ntty]
-No_Wait
-FREQ integer
-TIMES integer
-START date/time
-STOP date/time

LIST_ASYNC [line-numbers
   -USER [user-names | user-numbers ]
   [general SIM options ]

Displays the status of any or all asynchronous lines. (EPF, 21.0)

General SIM options:
   -Help [-No_Wait]
   -USAGE
   -ON node | nodegroup
   -Private_LOG pathname [-Ntty]
   -System_LOG pathname [-Ntty]
   -No_Wait
   -FREQ integer
   -TIMES integer
   -START date/time
   -STOP date/time


LIST_CATALOG

List contents of archive or backup tape (BRMS). (EX)
Ref: Operator's Guide to System Backups [34].

LIST_COMM_CONTROLLERS [general SIM options ]

Displays information on comms controllers on the network. (EPF, 21.0)

General SIM options:
   -Help [-No_Wait]
   -USAGE
   -ON node | nodegroup
   -Private_LOG pathname [-Ntty]
   -System_LOG pathname [-Ntty]
   -No_Wait
   -FREQ integer
   -TIMES integer
   -START date/time
   -STOP date/time


LIST_CONFIG [directive-names ] [general SIM options ]

Displays the various values of system variables. (EPF, 21.0)

General SIM options:
   -Help [-No_Wait]
   -USAGE
   -ON node | nodegroup
   -Private_LOG pathname [-Ntty]
   -System_LOG pathname [-Ntty]
   -No_Wait
   -FREQ integer
   -TIMES integer
   -START date/time
   -STOP date/time

LIST_DISKS [disk-names] [-USERs] [-LOCAL] [-REMOTE]
  [general SIM options]

Displays information for local and remote disks. (EPF, 21.0)

General SIM options:
- Help [-NoWait]
- USAGE
- ON node | nodegroup
- Private_LOG pathname [-Ntty]
- System_LOG pathname [-Ntty]
- No_Wait
- FREQ integer
- TIMES integer
- START date/time
- STOP date/time


X.LIST_DISKS [disk-name] [-ON system] [-SIZE]
  [-SYstem system] [-LOCAL] [-DETail]

List disk status. (EPF, NR)

List_DuMP [-HELP]

List the current values for a partial tape dump. (IN, OP)
Ref: Operator's Guide to System Commands [35].

List_Epf [pathname_1,...pathname_N] [-ACTIVE | -NOT_ACTIVE]
  [-NOT_MAPPED] [-PROGRAM] [-LIBRARY] [-SEGMENTS]
  [-COMMAND_PROCESSING] [-EPF_DATA] [-DETAIL]
  [-NO_WAIT] [-HELP]

Display information about EPF mapped in. (IN, 19.4)
Ref: Programmer's Guide to BIND and EPFs [51].

List_Group

List ACL groups. (IN)
Ref: PRIMOS Commands Reference Guide [49].

LIST_LAN_NODES [lan-names] [-HOST] [-LTS]
  [general SIM options]

Displays all nodes on LAN300 networks. (EPF, 21.0)

General SIM options:
- Help [-No_Wait]
- USAGE
- ON node | nodegroup
- Private_LOG pathname [-Ntty]
- System_LOG pathname [-Ntty]
- No_Wait
- FREQ integer
- TIMES integer
- START date/time
LIST_LHC_STATUS [options]

Show status of LHC300 controllers. (EPF)

Options:
- Dest_Node_Name node-name
- Dest_Node_Address node-address (hex pairs)
- Dest_LHC_number lhc-number
- Lan_Name lan-name
- Help
- PERFORMANCE
- CONNECTION connection-type
- ManageMenT
- ALL
- No_Wait

Ref: PRIMENET Planning and Configuration Guide[46] and NTS Planning and Configuration Guide[29].

List_Library_ENTRIES [pathname1,...,pathname8]

[-ACTIVE | -NOT_ACTIVE] [-NO_Wait] [-Help]
[-ENTRYname entry1,...,entry8] [-NOT_MAPPED]

List entrypoints in EPF libraries. (IN, 19.4)
Ref: Programmer's Guide to BIND and EPFs [51].

List_Limits

List the limit of segments and program invocations/levels authorized. (IN, 19.4)
Ref: Programmer's Guide to BIND and EPFs [51].

LIST_LTS_STATUS [ options ]

Show status of LAN Terminal Servers. (EPF)

Options:
- Dest_Node_Name node-name
- Dest_Node_Address node-address (hex pairs)
- Help
- PERFORMANCE
- CONNECTION connection-type
- ManageMenT
- ALL
- No_Wait

Ref: NTS Planning and Configuration Guide[29].

LIST_MEMORY [user-names | user-numbers]

[-TYPE user-types]
[general SIM options]

Displays memory usage per user process. (EPF, 21.0)

General SIM options:
- Help [-NO_Wait]
-USAGE
-ON node | nodegroup
-Private_LOG pathname [-Ntty]
-System_LOG pathname [-Ntty]
-No_Wait
-FREQ integer
-TIMES integer
-START date/time
-STOP date/time


List_Mini_Commands [command_match]

Display the available commands at mini-command level. (IN, 19.4)
command_match is a command name that may contain wildcards. Ref: Programmer's Guide to BIND and EPFs [51].

LIST_PRIMENET_LINKS [node-names | PDN names]
    [-LINK link-devices]
    [general SIM options]

Displays the status of PRIMENET links. (EPF, 21.0)

General SIM options:
-Help [-No_Wait]
-USAGE
-ON node | nodegroup
-Private_LOG pathname [-Ntty]
-System_LOG pathname [-Ntty]
-No_Wait
-FREQ integer
-TIMES integer
-START date/time
-STOP date/time


LIST_PRIMENET_NODES [node-names]
    [-LINK link-devices]
    [general SIM options]

Displays all PRIMENET configured remote nodes. (EPF, 21.0)

General SIM options:
-Help [-No_Wait]
-USAGE
-ON node | nodegroup
-Private_LOG pathname [-Ntty]
-System_LOG pathname [-Ntty]
-No_Wait
-FREQ integer
-TIMES integer
-START date/time
-STOP date/time


LIST_PRIMENET_PORTS [port-numbers]
[USER user-names | user-numbers]
[general SIM options]

Displays a system's port assignments. (EPF, 21.0)

General SIM options:
- Help [-No_Wait]
- USAGE
- ON node | nodegroup
- Private_LOG pathname [-Ntty]
- System_LOG pathname [-Ntty]
- No_Wait
- FREQ integer
- TIMES integer
- START date/time
- STOP date/time


List_Priority_ACcess [disk-name]

Show any priority acs on a disk. (IN)

LIST_PROCESS [user-names | user-numbers]
- PROJECT project-groups
- TYPE user-types
- DETail
[general SIM options]

Displays the environment of a specified user. (EPF, 21.0)

General SIM options:
- Help [-No_Wait]
- USAGE
- ON node | nodegroup
- Private_LOG pathname [-Ntty]
- System_LOG pathname [-Ntty]
- No_Wait
- FREQ integer
- TIMES integer
- START date/time
- STOP date/time


List_Quota [pathname] [-BRief]

Show quota and current usage on a directory. (IN)
Ref: PRIMOS Commands Reference Guide [49].

LIST_RBF treename [-DETAIL] [-SIZE]

List attributes of an RBF file. (EX)

List_Remote_ID [-ON nodename]
List all of the remote IDs for this user. (IN)
Ref: PRIMOS Commands Reference Guide [49].

List_Search_Rules

List all of the search rules in effect for user. (IN, 19.4)
Ref: Programmer’s Guide to BIND and EPFs [51].

List_Segment [segno1...segno8] [-STatic] [-DYnamic]
[-BRief] [-No_Wait] [-Help] [-NAME]

Show segments in use for user. (IN, 19.4)
Ref: Programmer’s Guide to BIND and EPFs [51].

LIST_SEMAPHORES [semaphore-numbers]
[-USER user-numbers | user-names]
[-TYPE {NAMED | NUMBERED}]
[general SIM options]

Displays the value of all in-use semaphores. (EPF, 21.0)

General SIM options:
-Help [-No_Wait]
-USAGE
-ON node | nodegroup
-Private_LOG pathname [-Ntty]
-System_LOG pathname [-Ntty]
-No_Wait
-FREQ integer
-TIMES integer
-START date/time
-STOP date/time


LIST_SYNC [line-numbers]
[general SIM options]

Displays the configuration of all enabled synchronous lines. (EPF, 21.0)

General SIM options:
-Help [-No_Wait]
-USAGE
-ON node | nodegroup
-Private_LOG pathname [-Ntty]
-System_LOG pathname [-Ntty]
-No_Wait
-FREQ integer
-TIMES integer
-START date/time
-STOP date/time


LIST_TAPE

List the contents of an archive/backup tape(BRMS). (EX)
Ref: Operator’s Guide to System Backups [34].
LIST_UNITS [user-names | user-numbers]  
[-PATHNAME pathname-prefix]  
[general SIM options]

Displays information relating to files, units and attach points. (EPF, 21.0)

General SIM options:
-Help [-No_Wait]
-USAGE
-ON node | nodegroup
-Private_LOG pathname [-Ntty]
-System_LOG pathname [-Ntty]
-No_Wait
-FREQ integer
-TIMES integer
-START date/time
-STOP date/time


[-BATCH] [-SLaves] [-ALL_Disks]  
[-PROjects] [-Disks disk-name]

Display the current users on a system. (EPF, NR)

LIST_VAR [wild-card-name...]

List CPL global variables. (IN)

Ref: PRIMOS Commands Reference Guide [49].

LIST_VCS [VC-ID-numbers]  
[-USER user-names | user-numbers]  
[-NODE node-names]  
[-LINK link-devices]  
[-PORT port-numbers]  
[general SIM options]

Displays the state of virtual circuits. (EPF, 21.0)

General SIM options:
-Help [-No_Wait]
-USAGE
-ON node | nodegroup
-Private_LOG pathname [-Ntty]
-System_LOG pathname [-Ntty]
-No_Wait
-FREQ integer
-TIMES integer
-START date/time
-STOP date/time


LOAD

R- and S-mode linker. (EX)

Subcommands:
ATTach [ufd] [password] [disk] [key]
  key=0=>don't set home, 1=>set home.
AUTomate [n]
  Linkareas of length n around module. n = 0 turns feature off.
CHeck [symbol] [par1...par9]
COMMON address Set COMMON TOP - 1
DC [END]
ENTire pathname
ERROR [num]; num = 0, 1, or 2.
EXecute [a] [b] [x] Uses START entry
FORce load pathname [addr] [linkstart] [linkrange]
FI Force prefix for FO, LO, LI commands.
HARDware definition
  177700 Must be zero
  000040 1=>Prime 400 instruction set
  000020 Unused
  000010 1=>Double prec. fl. pt.
  000004 1=>Single prec. fl. pt.
  000002 1=>Prime 300 instruction set
  000001 1=>High speed arithmetic
INItialize pathname [addr] [linkstart] [linkrange]
  Resets everything and loads pathname.
Library [pathname] [addr]
  Loads binary from LIB; default is LIB>FTNLIB.
LOAD pathname [barea1]...[barea9] pathname [barea1]...[barea9] pathname symbol
  [barea1]...[barea9]
MAP [pathname] [option]
  Create a load map. Default pathname is $F. option = 0=>full map, 1=>load state,
  2=>load state and link info, 3=>unresolved references, 4=>same as 0, 5=>system
  programmer map, 6=>sorted unresolved references, 7=>sorted full map, 10=>symbol
  map for PSD.
MODE [D32R | D64R | D16S | D32S | D64V | D32I]
P/ Page boundary prefix for FO, LO, LI commands.
PAuse
PBrk [symbol] [par1...par9] * part [par2...par9]
QUIT Back to PRIMOS
RR
SAVE pathname [a [b [x [keys]]]]
SEtbase [linkstart] [linklen] * [end of sector] (*=>current sector)
SS symbol
SYMBOL symbol oldsymb [par1...par9] symbol addr [par2...par6] symbol * [par1...par3]
  parameters can contain + and - signs
SZ [NO | YES]
Virtualbase linkstart tosector
XPunge [y] [z] y: 0=>all but undefined symbols, 1=>all but undefined and COMMON. z:
  0=>all defined base areas, 1=>all but sector 0, 2=>return all.
Ref: SEG and LOAD Reference Guide [23].
LOGIN [username [-ON nodename]] [-PROJect project]
Login to system. (LO)
Ref: PRIMOS Commands Reference Guide [49].
LOGout [-USMO | ALL]
Logout user. (IN, OP)
usmo must have same login name as user unless issued by System user. Ref: PRIMOS
Commands Reference Guide [49].

LOGPRT [outfile]
[ LOGLST Tty ] [ -Help ]
[ -From [mmddyy hhmm] ]
[ -Input pathname ]
[ -Type [Cold | Warm | Timdat | CChecks | Disk | DSKnam
Overfl | Shutdn | CHK300 | Par300 | Mod300
TYPE10-TYPE15 | REMARK | POWERF ]
[ -Spool ]
[ -CONTin ]
[ -DBug ]
[ -Census ]
[ -Remark ]
[ -Dump ]
[ -Delete ]
[ -PURGE ]

Print LOGREC. (EX, OP, OBS)
Prompts for input pathname, default (just .CR.) is CMDNC0>LOGREC. Replaced by
PRINT_SYSLOG and PRINT_NETLOG at rev 20.0. Ref: Operator’s Guide to System
Monitoring [36].

LON [-ON | -OFF]
Control logout notification. (IN)
Ref: PRIMOS Commands Reference Guide [49].

LOOK [-usmo] [segno [access [mapseg]]]

Map segment to user 1. (IN, OP)
Defaults are 1 6000 200 4001. Ref: Operator’s Guide to System Commands [35].

LOOPBACK [source-options] dest-options
[ -LAN_NAME lan-name ] [-HELP]

Check network integrity. (EX)
Source-options:
-Src_Node_Name node-name
-Src_LHC_number lhc-number
-Src_Node_Address node-address
-Src_Lbk_Layer (NMSR | NME)

Dest-options:
-Dest_Node_Name node-name
-Dest_LHC_number lhc-number
-Dest_Node_Address node-address
-Dest_Lbk_Layer (NMSR | NME)

MAGNET [-SILENT] [-USER | -OPeRator] [-OVERWRITE]

Transfer data to and from tape. (EX)
Ref: Magnetic Tape User’s Guide [24].

Prime Restricted 2-59
MAGRST [-7TRK] [-TTY] [-QUERY] [-Cam_RBF] [-Dam_RBF]

Magtape restore. (EX)

Example:
  TAPE UNIT: 0 - 7
  ENTER LOGICAL TAPE NUMBER:
    logical tape number or 0 if positioned
  READY TO RESTORE:
    Yes
    No
    Partial
    $1 [filename ] [level ] (turn on indexing)
    $A ufd [password ] [kdev ] [key ] (attach to ufd)
    NW [filename ] [level ] (index only)
  TREE NAME:
    pathname per line. End with null line.


MAGSAV [-7TRK] [-INC] [-UPDT] [-VAR] [-P300] [-Cam_To_Dam]
 [-Save_UFD] [-TTY] [-No_Acl] [-NO_RBF] [-REV19]

Magtape save. (EX)
  TAPE UNIT: n (where 0 ≤ n ≤ 7)
  ENTER LOGICAL TAPE NUMBER: 0 -- tape already positioned
    or- n -- nth logical tape
  TAPE NAME: 6-character name
  DATE: mm dd yy or
    CR for today's date
  REV NO: an arbitrary integer
  NAME OR COMMAND:
    pathname to save an object (file or dir)
    $A ufd [password ] [kdev ] [key ]
      attach to ufd
    $Q terminate tape and return to PRIMOS
    $R terminate tape, rewind, and return to PRIMOS
    $I [filename ] [level ] print index to
      indicated level
    $UPDT [ON | OFF] set dumped switch
      (OFF default)
    $INC [ON | OFF] include only items with a
      set dumped switch
    OLD [ON | OFF] create old partition
      format
    $VALID [ON | OFF] check for conformance
      to new file name rules
    MFD save entire disk (must be attached to
      MFD)
    * save current directory


MAIL [filename ] [username [username... ] | !file ]
[send-options ] [options ]
Send and receive PDNMail (EPF, NR/Custom Systems)

Options:

- List List headers.
- -Help
  Display usage info.
- -Into Force interactive mode.
- -Delete_Mailbox
  Deletes the user's mailbox.
- -Set_Forward_to user-address
  Forward mail to another user/node.
- -Cancel_Forwarding
  Cancel mail forwarding.
- -Xmail
  Do not collect X.MAIL.
- -Alias filename
  Use filename for user aliases.
- -To user-addresses
  Users to send mail.
- -File filename
  File to send as mail.
- -Subject subject
  Set subject field.
- -CErtify
  Certify receipt of mail.
- -CC user-addresses
  Send 'carbon copies' to user-addresses.

User-address is of the form:

Local user (same machine): username
User at other company machine: username@machine-name
Through a relay machine: @relay,[@relay,...]:user@site
User at x.mail site: username@xmail-site.XMAIL

Ref: PDN Mailer User's Guide [41]

X.MAIL [user [pathname ] [-ON nodename ]] [options ]

Send and receive mail. (EX, NR)
To send mail, user must be specified. Terminate mail with a $ or a ctrl-C. To print mail, use no options. Options:

- -Check
  Reports mail availability
- -List List headers only
- -LFirst
  List headers and first lines
- -Append pathname
  Append current mail to a file pathname
- -Spool formtype
  Spool mail to printer using formtype
- -Held
  List all held mail by username
- -SRTM
  Sort by amount held.
- -NSRT
  Sort by name.
-ON node
  Perform action on node.
-PORT n
  Use x.25 port n.
-Users
  Provide list of users.
-NW
  Don’t paginate.
-NQ Don’t query about mail being read.

MAKE -DiSK   pdev -PARTition name -Disk_Type disk_type [options ]

Format disk. (EX)

options:

-Disk_Type disk_type
  Specifies what kind of disk. Valid types are:
  SMD      80MB or 300MB removable
  CMD      cartridge module device
  68MB     68 megabyte fixed media
  158MB    158 megabyte fixed media
  160MB    160 megabyte fixed media
  600MB    600 megabyte fixed media
  MODEL_4475315 megabyte fixed media
  MODEL_471160 megabyte fixed media (rev 21)
  MODEL_471589 megabyte fixed media (rev 21)
  MODEL_4714120 megabyte fixed media (rev 21)
  MODEL_4719258 megabyte fixed media (rev 21)
  MODEL_4735496 megabyte fixed media (rev 21, pickeral)
  MODEL_4845770 megabyte fixed media (rev 21, beluga)

FLOPPY floppy disk (diskette, OBSOLETE as of 21)

-SPLIT [#-of-paging-records ]
  Make part of the partition for paging. If number of paging records is not given, MAKE will print the total number available and ask for number of paging records.

-PRE_rev19 Create a pre-rev 19 partition.*

-BADspot_LEVEL bad-spot-checking-level
  Checking level can be from 0 to 4 inclusive. If level 0 is specified, no checking is done. Level 4 gives the best checking. The default is 1 for SMD or CMD, 4 for fixed media disks.

-BAUD_rate valid-baud-rate
  Set initial baud rate of system console. Valid baud rates are: 110, 300, 1200, or 9600. The default is 300.

-NO_INIT
  Do not initialize the file system part of the disk. Unless this is specified, the records are initialized.

-ForMaT
  Write hardware formats on the disk. Use this only if the disk has never been used on a Prime system.

-map_UNCORR
  Map out only records with uncorrectable errors. Default is map out all records with any error-uncorrectable or correctable. Use of this option is not recommended.

-Query_BADSpots
  Query user for known bad spots on disk.

-NEW_DISK Suppress the attempt to read the old badspot file.

-CoPY_badspots_by_NAME partition
  Copy the badspots from the disk specified by the name partition.
-CoPY_badspots_by_DEVice copy-pdev
  Copy the badspots from the disk specified by the device copy-pdev.
-DISK_REVision {18 | 19 | 20 | 21}
  Specify which revision of disk to make. Most recent rev is assumed. (20.0)
-Override_Default_Interleave
  Override default interleave. (21.0)
-NO_FLaw_MaP
  Disable the usage of the flaw map. (21.0)
-All_Controller
  Create a compatible disk, for all controllers (21.0)
-Intelligent_Controller
  Create a mirrorable, dynamic badspotting disk (21.0)
-ROBust
  Create a robust partition. (NR)
-MINExtent_SIZE
  Specify minimum extent size for a CAM file; default is 64 for robust partition,
  16 for normal. (NR)
-MAXExtent_SIZE
  Specify maximum extent size for a CAM file; default is 256 for robust partition,
  32 for normal. (NR)

Ref: Operator's Guide to File System Maintenance [32].

MAXSch n

Set scheduling constant. (IN, OP)
Default value is 3. Ref: Operator's Guide to System Commands [35].

MAXusr [number | ALL]

Limit number of logged-in users. (IN, OP)
Ref: Operator's Guide to System Commands [35].

MCLUP

Midas cleanup utility. (EX, OP, OBS)
Ref: MIDAS Reference Guide.

MDUMP

Utility for recovering MIDAS files. (EX)

MED_SPOOL

Spool a MEDUSA plot file. (EX)

MEDCONFIG [project-name ]

Medusa system configurator. (CPL, EX)

MEDUSA [workstation-directory ]

Medusa graphics design program. (CPL, EX)

Message [-usmo | username | ALL] [-NOW] [-FORCE]
  [-ON nodename]
  [-DEFER | -REJECT | -ACCEPT ]
-STATUS [username | user-id | ME]]

Send message to user(s) or system. (IN)

Mirror_OFF pdev1 pdev2 [options ]

Shuts off disk mirroring. (IN, OP)
Options:
- SHUT_BOTH
- SHUTPRIMARY
- SHUT_SECONDARY
- FORCE

Ref: Operator's Guide to File System Maintenance [32].

Mirror_ON pdev1 pdev2 [options ]

Turns on disk mirroring. (IN, OP)
Options:
- NO_QUERY
- PRIority_SELECT
- HELP

Ref: Operator's Guide to File System Maintenance [32].

MODULA pathname [CE-options ]

Modula-2 compiler. (NR)
See the compiler options, 2.7.1. Ref: Modula-2 Reference Guide [26].

MONITOR_NET [options ]

Monitor Primenet. (EX)
Options:
- Ring [D]
- Sync [line-number]
- Virtual
- PERIOD seconds
- LANGUAGE language
- TIMES repeat-count
- Reset_Day
- Reset.HOUR
- Input filename
- OUtput filename
- TRace
- Terminal_Type terminal
- Help
- RePorT filename

Ref: PRIMENET Guide [45].

MONITOR_RING

Monitor ring network. (EX, OBS)
Obsolete as of 19.4. Use MONITOR_NET. Ref: PRIMENET Guide [45].
MPACK

MIDASPLUS file packing utility. (EX)
Ref: MIDASPLUS Reference Guide [25].

MPLUSCLUP [-USER user-number | -ALL]

MIDASPLUS cleanup utility. (EX)
Ref: MIDASPLUS Reference Guide [25].

MRGF p_1 p_2 [..p_5] -OUTF p [-MINL [n]]
[-BRief] [-FORCE] [-REPORT pathname]

Merge ASCII files. (EX) MRGF edit commands:

A  insert all differing lines in p_1
B  insert all differing lines in p_2
C  insert all differing lines in p_3
D  insert all differing lines in p_4
E  insert all differing lines in p_5
An insert line n of p_1
En insert line n of p_5
PA print all differing lines in p_1
PE print all differing lines in p_5
PAm,n print lines m thru n of p_1
PEm,n print lines m thru n of p_5
OOPS undo previous editing for this discrepancy
GO terminate editing and continue MRGF
Quit terminate editing, close all files, and exit from MRGF

Ref: PRIMOS Commands Reference Guide [49].

MTDENs MTn [-6250BPI | -1600BPI]

Set magnetic tape density. (EX, P2)
PRIMOS II only.

MTRESUME MTn [-Logical_Tape ltn ]
{pathname [-CoMmand_line_OPTSions options]}
[-INDEX n [-Page_Length lines] [-No_Wait]
[-Help]}

Execute (resume) a command from magtape. (IN)
Ref: Operator's Guide to System Commands [35].

NCOBOL [same options as COBOL]

Nonshared old COBOL compiler. (EX, P2, OBS)

Net [-ASSIGN line | -START line [-SITE nodename ] |
-STOP {line | nodename} | -UNASSIGN line]

Control half-duplex network. (IN, OP) Ref: Network Planning and Administration Guide [27].
NETCFG [-NOCHECK] [-DSC]

Configure PRIMENET. (EX, OBS)
Prompts for the following about the RING, IPC, SMLC and PDN network types: name, PDN address, ID, slave #, line #, enable FAM, permit remote FAM to start disks, enable remote login. Use CONFIG_NET after 19.2. Ref: PRIMENET Guide [45].

NETLINK [options]

- Network linker. (EX)
- Prompts with an @. NETLINK subcommands are:
  - C address: Connect to an address
  - COninue: Continue a currently active circuit.
  - D: Disconnect the currently active circuit.
  - HELP: Invoke the help function.
  - NC address: Connect without reverse charging.
  - PAuse: Exit to PRIMOS but allow returns.
  - Quit: Exit to PRIMOS command level.

Ref: PRIMENET Guide [45].

NETLOG

Convert NETREC file to an ASCII file. (EX, OBS)

NETLVL

Change severity level of network errors. (IN, OP, OBS)

NSED [filename]

Non-shared EDitor. See ED for commands. (EX, P2)

NTS_ASSOCIATE [-LINE primos_line_number] [-Lts_NAME Its_name [-Lts_LINE Its_line_number]] [-PERManent]

Associates an LTS line number with a PRIMOS line number for assignment. (EX, 21.0)

NTS_LINE -CoMmanD

Sets the NTS line the user is logged on to into LTS command mode. (EX, 21.0)

NTS_LIST_ASSOCIATE [-LINE primos_line_number [-Lts_NAME Its_name [-Lts_LINE Its_line_number]]]

Lists NTS assigned line associations. (EX, 21.0)

NTS_UNASSOCIATE [-LINE primos_line_number [-Lts_NAME Its_name -Lts_LINE Its_line_number]]

Removes an association between an NTS line and an assignable PRIMOS line. (EX, 21.0)

NUMBER

Number or renumber a BASIC file. (EX)
Prompts for pathnames and starting and increment numbers. Ref: PRIMOS Commands Reference Guide [49].

Prime Restricted
OAS
Enter into Master Function Selection of OAS. (EX)

OA_ADMIN
Enter into System Administrator Function Selection of OAS. (EX)
Ref: OAS Administrator's Guide [30].

OA_TERM
Downline load PT65 (Ontel) for OAS. (EX)

Open [pathname] unit key
Open file on specified unit. (IN)

<table>
<thead>
<tr>
<th>key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Read</td>
</tr>
<tr>
<td>2</td>
<td>Write</td>
</tr>
<tr>
<td>3</td>
<td>R/W</td>
</tr>
<tr>
<td>4</td>
<td>Close</td>
</tr>
<tr>
<td>5</td>
<td>Delete</td>
</tr>
<tr>
<td>6</td>
<td>Exist</td>
</tr>
<tr>
<td>7</td>
<td>Rewind</td>
</tr>
<tr>
<td>10</td>
<td>Truncate</td>
</tr>
<tr>
<td>+0</td>
<td>File is in current directory</td>
</tr>
<tr>
<td>+100</td>
<td>File is entry in segdir open on funit</td>
</tr>
<tr>
<td>+1000</td>
<td>Change open mode of funit</td>
</tr>
<tr>
<td>+0000</td>
<td>New SAM</td>
</tr>
<tr>
<td>+2000</td>
<td>New DAM</td>
</tr>
<tr>
<td>+4000</td>
<td>New SAM segment</td>
</tr>
<tr>
<td>+6000</td>
<td>New DAM segment</td>
</tr>
<tr>
<td>+10000</td>
<td>New UFD</td>
</tr>
</tbody>
</table>

pathname optional only for Rewind and Truncate. Ref: PRIMOS Commands Reference Guide [49].

OPRpri [ 1 | 2 ]
Set operator privilege. (IN, OP, OBS)

ORigin
Attach user to origin (login) directory. (IN)
Ref: PRIMOS Commands Reference Guide [49].
OSLOG

Control OS logging facility.

OWLDSC [-FAST] [-NOLOCK] [-REPORT]

Owl interface program for DPTX. (EX)
Ref: Distributed Processing Terminal Executive Guide [12].

PASCAL filename [CE-options]

Invoke Pascal compiler. (EX)
Ref: Pascal Reference Guide [37]. See Compiler options, 2.7.1 for options.

PASSWD [owner-password [non-owner-password]]

Set passwords on current UFD. (IN)
If not given, passwords are blanks (no password). Ref: PRIMOS Commands Reference Guide [49].

PassWord_DIRs [-ON | -OFF]

Sets the ability to create password directories. (OP, IN)

PHantom pathname [funit | CPL-arguments]

Start phantom user. (IN)
Funit only valid for cominput files. If using a cominput file, file should end with 'Logout' command. Ref: PRIMOS Commands Reference Guide [49].

PHYRST [-UNMOD] [-TTY] [-NO_BADS] [-SPEED {25 | 100}]

Physical disk restore. (IN, OP)

UNIT NO: n | Quit (for tape drive n)
LOGICAL TAPE: n (for nth logical tape)
CORRECT TAPE? YES | NO
RESTORE ALL PARTITIONS TO ORIGINAL POS? YES | NO
RESTORE PARTITION **********? YES | NO
AS PARTITION: CR | pdevno
PARAMETERS OK? YES | NO

Ref: Operator's Guide to System Backups [34].

PHYSAV [-UNMOD] [-TTY] [-COMDEV] [-SPEED {25 | 100}]

Physical disk save. (IN, OP)

UNIT NO: n (for tape drive n)
LOGICAL TAPE: n (for nth logical tape)
**COMMENT** up to 80 char comment
PHYS. DEV. NO: physical-device-number-to-be-saved
USE THE RAT? YES | NO
40MB DISK? YES | NO
PARAMETERS OK? YES | NO

Ref: Operator's Guide to System Backups [34].
PL1 `pathname [CE-options ]`

Full PL/I compiler. (EPF)
Uses same options as PL1G, see 2.7.1. Ref: PL1 Reference Guide [42].

PL1G `pathname [CE-options ]`

PL/I subset G. (EPF)

PLIB

Random EDMS utility. (EX)

PLOT

PRIMEAIDS plot utility. (EX)

PLP `filename [options ]`

PLP compiler: (EX, QT)

Pm

Print user register vector. (IN)
Displays starting address(SA), ending address(EA), program counter(P), register contents for: A, B, X, keys(K), procedure base(PB), stack base(SB), linkage base(LB), and the temporary base(XB). Ref: PRIMOS Commands Reference Guide [49].

PMA [-Input] `ipath [-B btree ] [-L ltree ]`

[1/a-reg ] [2/b-reg ] [3/x-reg ]

Prime Macro Assembler. (EX)
Options:
Errlist Errors-only listing
EXplist Expanded listing

<table>
<thead>
<tr>
<th>A-REG</th>
<th>ON-OPTION</th>
<th>OFF-OPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100000</td>
<td>Unused</td>
</tr>
<tr>
<td>2</td>
<td>040000</td>
<td>Errlist</td>
</tr>
<tr>
<td>3</td>
<td>020000</td>
<td>EXplist</td>
</tr>
<tr>
<td>4-7</td>
<td>017000</td>
<td>Unused</td>
</tr>
<tr>
<td>8-10</td>
<td>000700</td>
<td>Input device (default = 7)</td>
</tr>
<tr>
<td>11-13</td>
<td>000070</td>
<td>Listing device (default = 7)</td>
</tr>
<tr>
<td>14-17</td>
<td>000007</td>
<td>Binary device (default = 7)</td>
</tr>
</tbody>
</table>

Device codes (Input, Listing, Binary):

0 - None 4 - Line Printer
1 - ASR 5 - Magtape
2 - PTR/PTP 6 - Cassette
3 - Card Reader 7 - Disk

B-REG (PRIMOS IV BUILD):

11-13 000020 64-user version
      000000 16-user version
      000001 Large 16-user version

Ref: Assembly Language Programmer's Guide [44].

POWER

Invokes the POWERPLUS data management facility. (EX)

PRATIO {value-pagdev0 [...value-pagdev7]} | -DISPLAY}

PRATIO
Sets or displays the ratio for the paging devices. (IN, OP)
Ref: Operator's Guide to File System Maintenance [32].

PRerr

Print ERRVEC and last error message. (IN)
Ref: PRIMOS Commands Reference Guide [49].

PRIMIX [UNIX-command]

Enter UNIX subsystem. (EPF)

PRIMOS [primos-directory]

Start Primos from Primos II. (OP, P2 only, OBS)
If primos-directory is given, that will replace the default for all subsequent executions.

PRINT_KSR

Print a file from SPOOLQ to keyboard printer (OAS). (EX, OBS)

PRINT_NETLOG [output-file | TTY] [options]

2-70 Prime Restricted
Convert a NETREC file to an ASCII file. (EX, OBS)
Replaced with DISPLAY_LOG at 21.0. Options:

-Census
   Reports totals of entries.
-COnIntinue
   Continue processing after a bad entry.
-DEBug
   Read entries from terminal for testing.
-Delete
   Delete output file when done.
-DUmp
   Display entries in octal.
-From [mmdyy [hhmm] | TODAY]
   Earliest entry to be processed.
-Help
   Print option usage.
-Input Pathname
   Input file to be used.
-PURGE
   Empty contents of event log file.
-Remark text
   Enter text into input file.
-Spool
   Spools the output file.
-Type type\_1 ... type\_n
   Process only these types.

Ref: PRIMENET Guide [45].

PRINT_SCS [pathname] [options]

Print a file containing SCS data streams (SNA). (EX)
Options:
-AS alias
-AT <destination>
-COPYies n
-DeFeR hh:mm
-DeLete
-DIsk [disk-name | ldev-number]
-ForMaT [NONE | PAGE]
-FORMat type
-NOHead
-No_Page_HeaDeR

Ref: Remote Job Entry Phase II User's Guide [52].

Print_Security_LOG -LOGFILE pathname [options]

Displays a report from a security audit file. (OP, 21.0)
Options:
-USERS userid-list
-NUMBER_OBJECT num-obj-list
-TEXT_OBJECT text-obj-list
-EVENTS [FILE SYSTEM] [SYSTEM] [PRIV_OPS] [ATTACHES]
-EVENT_Types [SUCCESS] [NO_ACCESS] [FAILURE]
-NO_WAIT

Prime Restricted 2-71
PRINT_SYSLOG

Converts LOGREC files to ASCII. (EX, OBS)

PRMPC pathname

Print file on line printer (PR0). (EX)
Ref: PRIMOS Commands Reference Guide [49].

PROP phantom-name option...

-STATUS

Control spooler phantoms. (EX, OP)
Environment options can be:

-CREATE -MODIFY
-DELETE -STATUS
-DISPLAY -COMPRESS

Phantom options can be:

-ABORT -BACK n
-CONTINUE -DROP
-HANG [NOW | FINISH | IDLE]
-RESTART -START
-STOP [NOW | FINISH | IDLE]

-CREATE and -MODIFY subcommands (obsolete at 21.0):

COMOUT [ON | OFF ]
DEST synonym
DEVICE [PR0 | PR1 | PR2 | PR3 | CENPR | CE2PR | PLOT | AMLC n ]
FILE
FORM synonym
HEADER [0 | 1 | 2]
LARGE [n ] (default: 30)
LENGTH [n ] (default: 38)
LIMIT [n ] (default: 3000)
LINES [n ] (default: LENGTH+13)
LOWER [n ] (default: 0)
MESSAGE text
PAPER [name ] (default: ‘ ’)
PLOT [ON | OFF]
PRINT [ON | OFF]
QUIT
UNDEST synonym
UNFORM synonym
UPCASE [ON | OFF]
UPPER [n ] (default: 63)
WIDTH [n ] (default: 180)

Ref: Operator's Guide to the Spooler Subsystem [33].
PROtec path[owner [nonowner]]

Set protection on file. (CPL, OLD)
0-No access (default), 1-Read, 2-Write, 3-R/W, 4-Delete/Truncate, 5-D/T/R, 6-D/T/W, 7-All.
Default on file creation equals 70. Ref: PRIMOS Commands Reference Guide [49].

PROTECT pathname [owner-code [non-owner-code]] [-RePorT]

Sets protection rights for password protected objects. (EPF)
Codes are:
NIL - no access (default)
R - read
W - write
D - delete
RW - read and write
RD - read and delete
WD - write and delete
RWD - all
Ref: PRIMOS Commands Reference Guide [49].

PRSER pathname

Print file on serial line printer. (EX)
Ref: PRIMOS Commands Reference Guide [49].

PRTDSC station1 [...stationn]

Printer emulation program. (EX, OP)
Ref: Distributed Processing Terminal Executive Guide [12].

PRVER pathname

Print file on Versatec. (EX)
Ref: PRIMOS Commands Reference Guide [49].

PSD [token...

Prime Symbolic Debugger. (EX)
(NOTE: VPSD has: segment, base register operations, does not have: symbols, trace.)

TERMINATORS for 'A'

.CR. "+1
*+1
^ "-1 (uparrow)
.n "+n
.-n "-n
@ Effective address
\ Back to last
( To contents of *
) Back to last defined (
= EA + contents, no update of *
/ Return, do not close *
? return, do not close *
I Return, close *

MODES
Expressions: Locations can be expressions including:

*(current location)*

`[+]number-in-current-mode`

`>number-relative-to-relocation-constant`

Subcommands:

**Access loc**
- Access location.

**Breakpoint loc**
- Set breakpoint (up to 10).

**BR**
- Print base registers.

**Copy from to new-addr**
- Copy block of memory to new location.

**Define sym val**
- Define symbol.

**Dump from to [ncoil] [mode]**
- Dump contents of memory.

**Effective from to match [mask]**
- Search for effective address.

**EXecute**
- Execute segmented program.

**FAddress fld-addr-reg-no**
- Access field address register.

**FLength fld-len-reg-no**
- Access field address register.

**Fill from to val**
- Fill memory block with val.

**GO [count] [a b [x k]]**
- Continue at breakpoint.

**Jumptrace [start] [a b]**
- Execute obj prog and produce diagnostic listing.

**Keys value**
- Set keys to value.

**LB sn wn**
- Set link base.

**List loc**
- list location.

**LS**
- Load symbols (unit 1).

**MAP**
- Print load map symbols.

**MO [D16S | D32R | D64R | D64V | D32S | D32I]**
- Set address mode.

**Monitor [start] [a b] addr**
- Trace obj prog for mem ref instr.

**Not-equal from to nmatch [mask]**
- Negative search.

**Open frame unit key**
- Open unit.

**PATCH loc1 loc2**
- Patch instr in loc2 into loc1.
Print
Print brkpt. contents, a, b, x, keys, relocation.
PRceed [newbrk [a [b [x [k]]]]]
Set new brkpt and resume execution.
Quit
Quit.
RELocate reloc-val
Set relocation constant.
Run [loc [a [b [x [keys]]]]]
Run program.
SB sn wn
Set stack base.
Search from to match [mask]
Search memory block.
SN sn
Set segment number.
SY 0 Symbol mode off.
SY 1 Symbol mode on.
Trace [addr [a [b [val [-1 interval]]]]]
Trace program.
Update loc val
Update location.
Verify from to copy-addr
Verify block of memory.
VERSION
Print version, restart address.
Where
Display brkpts and proceed counts.
X reloc-val
Set relocation constant.
XB sn wn
Set X base.
XR val
Set X register.
YR val
Set Y register.
Zero [brk-loc]
Remove brkpt (current)

Ref: Assembly Language Programmer's Guide [44].

PSD20

PSD for 16K PRIMOS II. (EX, P2)
See PSD.

PST100DSC

IBM 3277 emulation program for PST100 (DPTX). (EX, OBS)
Replaced by PTDSC.

PTDSC

IBM 3277 emulator for PST100 or PT200 (PDTX). (EX)
Ref: Distributed Processing Terminal Executive Guide [12].

PTELE

Access OAS telephone inquiry function. (EX)
PT45DSC

IBM 3277 emulator for PT45. (EX)
Ref: Distributed Processing Terminal Executive Guide [12].

PT46DSC

IBM 3277 emulation for PT46(DPTX). (EX)
Ref: Distributed Processing Terminal Executive Guide [12].

RDY [-LONG | -BRief] [-ON | -OFF]
[-Ready_Long text] [-Ready_Brief text]
[-Error_Long text] [-Error_Brief text]
[-Warning_Long text] [-Warning_Brief text]

Choose prompt messages. (IN)
Ref: PRIMOS Commands Reference Guide [49].

REeNter

Re-enter a subsystem after quitting. (IN)
Ref: PRIMOS Commands Reference Guide [49].

REFORM

Representation formatter for files with STROMA constructs. (EX, QT)

ReLeaSe_level [-ALL | -TO n | -LeVels n]

Release one or more stack levels. (IN)
Ref: PRIMOS Commands Reference Guide [49].

REMote PERMIT [option]

Set remote access to local files. (IN, OBS)
Obsolete as of 19.3. Option can be:

- node pdev₁ [ . . . pdevₙ ]
- node -ALL
- NET pdev₁ [ . . . pdevₙ ]
- NET -ALL

REMove_EPF [pathname] [-ACtive | -Not_Active] [-Help]
[-VeriFY | -No_VeriFY] [-QUERY | -No_Query]
[-PRoGram | -Library] [-Force]

Unmap EPF from user workspace. (IN, 19.4)
Ref: Programmer's Guide to BIND and EPFs [51].

Remove_Priority_ACcess disk-name

Removes priority ACLs from disk-name. (IN, OP, SA)
Ref: Operator's Guide to System Commands [35].

Remove_Remote_ID -on nodename
Removes a remote ID established by ARID. (IN)

RENT

Re-enter subsystem after quit. (IN)  
See also: REENTER.

REPLY { -username -TAPE {GO | ABORT | pdn | RESEND} |  
-TAPE RESEND | -ALL -RESEND | -username -RESEND |  
-REPEAT seconds}  
Reply to a tape drive request. (IN, OP)  
Ref: Operator's Guide to System Commands [35].

Reset_DuMP [-HELP]

Resets partial tape dump parameters to their default values. (OP, IN)  
Defaults are:  
0 to 1777 (ring 0 PRIMOS)  
6000 to 6003 for all logged-in users  
4000 to 7777 for the process that was using the CPU at the time  
Ref: Operator's Guide to System Commands [35].

RESTATE

Representation converter(SPS). (EX, QT)

RESTor pathname

Restore external program. (IN)  
Ref: PRIMOS Commands Reference Guide [49].

RESTORE_RBF src-pathname [dest-pathname] [-No_Query] [-PROtect]  
[-DAM] [-CAM] [-Min_eXt_Len] [-RePort]

Activate an inactive RBF file. (EX)

Resume pathname [arguments...] [p [a  
[b [x [k ]]]]]

Run an EPF, CPL or static mode program. (IN)  
P, a, b, x and k are only valid for static mode programs. Ref: PRIMOS Commands  
Reference Guide [49].

REsus subcommand

Remote Systems User facility. (EPF, 21.0)

Subcommands:
-ENABLE  
-DISABLE [-FORCE]  
-START [-ON node-name]  
-STOP  
-STATUS [-ON node-group]  
-Help [-No_Wait]  
-USAGE  
REVERT_PASSWORD

Change current directory from ACL to password. (EX)
Ref: PRIMOS Commands Reference Guide [49].

RJ1004, RJ200UT, RJ7020, RJX80, RJGRTS, RJJHASP

Submit job to remote site (EX, OLD)
Replaced by RJOP. Ref: Remote Job Entry Guide.

RJOP

RJE operator command processor. (EX)
Ref: Remote Job Entry Phase II User's Guide [52].

RJQ path name [-TO] {queue name | sitename} [options]
RJQ [-LIST | -CANCEL | -RESET] [OWN | ALL | RJxxx] [options]

Send a job to a remote mainframe via RJE system. (EX)
Options for submitting:

-WITH protocol
-Defer time
-No_Copy
-DeLete
-No_Translate
-AS internal-name
-DEVice {CR[n] | LP[n] | CP[n]}
-VFC {NONE | FTN}
-Keep_Request
-LU lu_port_name
-MEDSUB medium/subaddress

Options for all others:

-TO queue name
-WITH protocol
-Defer time  (-LIST only)

Ref: Remote Job Entry Phase II User's Guide [52].

RLS

Release stack history. See RELEASE_LEVEL.(IN)
Ref: PRIMOS Commands Reference Guide [49].

RO_TRACE_EVENTS filename [-SYSTEM | -USER userno]
[-ON (option|option2 ... | -ALL)]
[-OFF [option1 option2 ... | -ALL]]
[-REMOTE_NODE node] [-DEBUG] [-No_Query]
[-DISPLAY] [-HELP]

Display ROAM actions taken by entire system or a user. (EX)

ROSAU

ROAM system administrator utility. (EX)
Ref: ROAM System Administrator's Guide [53].
ROUTL [-DumpFILE treename] [-No_INVoKe]

Interactive tool to examine ROAM shared memory. (EX)

RPG filename [-SEQCHK | -NOSEQCHK] [-BANNER | -NOBANNER]
 [-OBDATA | -NOOBDATA] [-STATUS | -NOSTATUS]
 [-ERRTTY | -NOERRTTY] [-LISTING | -BINARY]

Report Program Generator (RPG II). (EX)
Obsolete after rev 20.0. Use VRPG.

RSTERM [-READ] [-WRITE]

Empty terminal I/O buffers. (IN)
Ref: PRIMOS Commands Reference Guide [49].

RUNOFF [pathname]

Text formatter. (EX)
Notes:
• When embedded in text, all runoff command lines begin with a period; when
issued at command level, runoff commands do not begin with a period.
• In the table below, some runoff command actions are followed by brk, ejt,
and/or deft to indicate the command causes a break, ejects a page, and/or is
the default. Also, if the runoff command has a default value, that value is
specified.

(str = text string) Subcommands:

.*. NULL. Start processing (from command mode).
* str. Comment line.
+ str. Enter verbatim string.
// // // Left/Center/Right/ strings.
> str. Center string.
Adjust. Enter adjust/fill modes (brk, deft).
BLank char. Define blank substitute character (.NULL.).
BMargin n. Set bottom margin (brk, ejt, 5).
Break. Break (start new line).
CMargin n. Set column margin (brk, ejt, 5).
Column. Set number of columns (brk, ejt, 1).
DDown str. Down Decimal level.
DDSUp str. Down decimal level, no decimal number.
DEfine sym str. Define symbol value.
DIndent lev before after. Set decimal indents. 0 => all levels
DLevel n. Go to decimal level n (1).
DLimit n. Set highest decimal level to appear in Table of Contents (all).
DNext str. Next heading on current decimal level.
DNSUp str. Next heading on current decimal level, suppress number.
DReset n. Reset number on decimal level n.
DSkip lev before after. Set decimal heading skip values. 0 => all; -1 => eject before
DUp [n]. Go up n decimal levels (1).
EEven. Eject to next even numbered page.
EFooter // // // Define even-page footer.
EHeader // // // Define even-page header.
Eject Page eject (brk, ejt).
EOdd Eject to next odd numbered page.
ERase char Define cmd mode erase char.
ERRgo Continue on error.
FILE fn Specify output file.
Fill Enter fill mode.
FLOAT fn Floating insert of fn.
FFooter / / / / Define footer for all pages.
FROM n First page number to output.
HEADER / / / / Define header for all pages.
HYphen char Define phantom hyphen char (.RUBOUT.).
INdent n Indent left margin (5).
INDEX str Write str and page number to index.
INSert fn [[parms]] Insert fn.
INSert unit Insert from unit.
IXfile fn Define index file (16).
Kill char Define command line kill char (?).
Length Specify physical page length (brk, ejt, 66).
NAdjust Leave adjust mode (brk).
NEed n Eject if < n lines (1).
NERgo Stop on error encountered (defft).
NFILE No output to file.
NFill Leave fill and adjust modes (brk).
NIXfile Stop output to index file.
NParagraph No paragraph indentation (defft).
NPAUse No pause between pages (defft).
NPERforate No perforation marks (defft).
NTty No output to TTY (defft).
OFooter / / / / Define odd-page footer.
OHeader / / / / Define odd-page header.
PAGen n Set page number (1).
Paragraph [m] [n] Start paragraph, indentm, skip n.
PAUse Pause between output pages.
PERforate Print perforation marks.
Picture n Leave n lines together (1).
PUrge Force in outstanding floats.
Quit Exit RUNOFF (brk, ejt).
RBBar [ON] Start revision bars.
RBBar [OFF] Stop revision bars.
REturn n Return to prev input file (0).
RIndent n Indent right margin (5).
RUndent n Undent right margin (0).
Skip n Skip n lines (brk, 1).
SM n Specify side margins (brk, ejt, 7).
SO n Print nth source line # (1).
SPace n Specify single/double, etc. spacing (1).
STOP Conditional .QUIT/.RETURN.
SYchar char Define symbol delimiter (%).
Tab char n1 ... Set tab character and stops.
TMargin n Specify top margin (brk, ejt, 7).
TO n Specify last page to print (32767).
TOFc fn lim Specify table of contents file.
TOFc [opt ] Close, stop, start table of contents for opt=omitted, 0, 1.
TTOf c str  Enter string in table of contents.
TTy    Output to TTY.
UNDEFINE sym
        Undefine symbol.
Undent n Undent left margin.
WIDOW n Specify allowable widow size (0).
Width n Specify paper width (brk, ejt, 85).
{{str}}  Underline str.

Ref: New User’s Guide to EDITOR and RUNOFF [28].

RWLOCK pathname lock [-RePorT]

Set file read/write lock. (EX)
lock may be: SYS - system default; EXCL - N readers OR 1 Writer; UPDT - N readers AND
1 writer; NONE - N readers and N writers. Ref: PRIMOS Commands Reference Guide [49].

SAve pathname [sa [ea [pa [ab [x
            [keys ]]]]]]

Save memory image. (IN)
Do not use SAvE with 64V or 32I segmented files or EPFs. Ref: PRIMOS Commands
Reference Guide [49].

SAVE_RBF src-pathname dest-pathname [-PROtect] [-DAM] [-CAM]
            [-Min_eXt_Len] [-RePorT]

Make a backup copy of an RBF. (EX)
Ref: ROAM Administrator's Guide [53].

SCHDEC [[-SCHEMA] schema-name [-LISTING] out-file ]

Invoke DBMS schema decompiler. (EX)
Ref: DBMS Administrator's Guide [8].

SCHED [pathname ]

Alter definition of database. (EX)
Ref: DBMS Administrator’s Guide [8].

SCHEMA pathname [-OUTPUT pathname ] [-LIST pathname ] [-DAM]

Invoke DBMS DDL compiler. (EX)
Ref: DBMS Data Description Language Reference Guide [9].

SCRIBE pathname [options ]

SCRIBE document formatter. (EPF)
Options:

-Agile Generate output for an Agile printer.
-Diablo Generate output for a Diablo printer.
-DEVICE name Generate output for the device name.
-DOCument name Produce output in file name.

Prime Restricted
DOVER
Generate output for a Dover printer.

-DRAFT [value]
Set the variable draft to value (default for value is 1).

-File
Generate output for the device file.

-Gal
Generate output for a GSI photocomposer.

-GIGI
Generate output for a GIGI.

-HypVocab
Create a lexicon showing the hyphenation points of each word in the document.

-HYD
Create a lexicon showing each hyphenation decision.

-IMPrint, -IMPTINT10
Create output for an Imagen Imprint10 laser printer.

-KEEPfiles
Don't delete temporary files.

-Lpt
Create output for a Line PrinTer.

-LA36
Create output for a DEC LA36.

-LGP1
Create output for an LGP1.

-PAGEDFILE
Create output for a PagedFile.

-Quiet
Don't print error messages on the terminal.

-Voc, -VOCABulary
Generate sorted word listings in a .LEX file.

-Words, -WORDCOUNT
Count the number of words in the document.

-X, -X9700
Create output for a Xerox X9700.

SECurity_MONitor [options] [-HELP]
Enables/disable audit collection. (IN, OP)

SECurity_STatus [options] [-HELP]
Display status of system events being audited. (IN, OP)

SEG [pathname [1/1] | -LOAD]
Segmented loader. (EX)
Giving pathname executes that segdir. 1/1 causes the segdir to be loaded and execution is passed to VPSD (See PSD for commands). -LOAD causes SEG to go into the LOAD subprocessor. Subcommands:

DELETE [filename]
deletes runfile.
HEIpprint list of SEG commands.
LOad [pathname]
  define runfile and invoke loader for creation.
LOad * [pathname]
  define runfile and invoke loader for appending.
ATTach [UFName] [password] [ldisk] [key]
  attach to UFDb.
ASYmble sname [segtype] segno size
  define a symbol in memory and reserve space for it using absolute segment
  numbers.
Common [ABS] segno
  relocate COMMON using absolute segment numbers.
Common REL segno
  relocate COMMON using relative segment assignment.
D/IL, D/LOAD, D/Library, D/FORCLOAD, D/PL or D/RL
  load using previous parameters. D/ and F/ may be combined.
EXECute
  save load to disk and execute program.
Fhx [filename] [addr psegno lsegno]
  force load all routines in object file.
IL [addr psegno lsegno]
  load impure FORTRAN library.
INItialize [pathname]
  initialize and restart loader.
Library [pathname] [addr psegno lsegno]
  load library file.
LO [pathname] [addr psegno lsegno]
  load object file.
MAP [filename] option
  generate load map.
MIXup [ON | OFF]
  mixes procedure and static data.
MV moves portion of loaded file. Will prompt for info.
NSCW
  Do not generate warnings for smaller redefinition of common blocks.
Operator [0 | 1]
  relax/impose high-level restrictions
PL [addr psegno lsegno]
  load pure FORTRAN library.
P/hx [filename] option] [psegno lsegno]
  load on a page boundary.
QUIT return to PRIMOS command level.
RETURN
  return to SEG command level.
RL pathname [addr psegno lsegno]
  reload a routine.
RSymble sname [segtype] segno size
  define symbol in memory and reserve space for relative segment assignment.
SAVE [a] [b] [x]]]
  save load to disk.
SCW
  Emit warnings for smaller redefinition of common blocks (default).
SE segno len
  create base area for desectorization.
SPLIT [segno addr | addr | addr ssizeo saddr esegno]
  break data into data and procedure portions
SS sname
   save symbol.
STack size
   change stack size.
SYmbol [sname] segno addr
   define a symbol at specific location in memory.
Sxx [filename] addr psegno lsegno
   load a specific absolute segment.
XP dsymbol dbase
   expunge symbols from symbol table and delete base information.
MAP {filename1 [.*] [filename2] [option]}
   prints specified load map.  option: 0=full map (default), 1=extent map only, 2=extent
   map and base areas, 3=undefined symbols only, 4=full map, 5=system programmer's
   map, 6=undefined symbols sorted, 7=full map sorted, 10=symbols by ascending addr,
   11=symbols sorted.
MOdify [filename] or SA [filename]
   invoke modification subprocessor.
NEW filename
   write new copy of runfile to disk.
PAtch
   modify save range of existing segment.
REturn
   return to SEG command level.
SK {size | segno addr}
   alter stack size and/or location.
STart segno addr
   change program execution start address.
WRite
   write all segments to disk.
PArams [filename]
   display parameters of runfile.
PSd invoke VPSD debugging utility.
Quit return to PRIMOS command level.
RESTore [pathname]
   bring runfile into user memory.
RESuMe [pathname]
   restore runfile and begin execution.
SShare [pathname]
   create R mode runfiles for segments below '4001.
Single [pathname] segno
   create R mode file image of single segment.
Time [pathname]
   print time and date of last runfile modification.
VERSION
   display SEG version number.

Ref: SEG and LOAD Reference Guide [23].

SEtime -mmddyy -hhmm

Set date and time.  (IN, OP)
Must be issued before user logins possible (unless machine has newer CP (2000, 4000,
6000 and 9000) series).  Ref: PRIMOS Commands Reference Guide [49] and Operator's
Guide to System Commands [35].
SETMod {-User | -Operator | -Noassign}

Control tape drive assignment. (IN, OP)
Ref: Operator’s Guide to System Commands [35].

Set_Access target [acl [-No_Query] | -LIKE reference
| -CATegory acat-name ]

Set access rights to an object. (IN)
Ref: PRIMOS Commands Reference Guide [49].

SET_ASYNC {-LINE n [-TO m] [options] | -DisPlay | -Help}

Set async line configurations (EPF, 21.0)
Options:
-DEFault
-SYSTEM
-ASSIGNable {NO | yes}
-Char_Length {5 | 6 | 7 | 8}
-[NO]_Data_Set_Control
-[NO]_Data_Sense_Enable
-Data_Set_Sense {HIGH | low}
-[NO]_DISLOG
-[NO]_ECHO
-[NO]_ERROR_Detection
-[NO]_Line_Feed
-[NO]_LOOP_line
-PARity {NONE | odd | even}
-Protocol [name]
-[NO]_REVerse_XOFF
-SPEED [value]
-[NO]_Speed_Detect
-STOP_bits {1 | 2}
-USER_number x
-[NO]_XOFF


SET_DELETE pathname [-PROtect | -No_PROtect]

Set/reset protection from deletion. (EX)
Ref: PRIMOS Commands Reference Guide [49].

Set_PRIORITY_Access disk-name acl

Put a priority ACL on a disk. (IN, OP)
Ref: Operator’s Guide to System Commands [35].

Set_Quota pathname [-Max n]

Set maximum number of records allowed on a directory. (IN)
Ref: PRIMOS Commands Reference Guide [49].

SET_RBF pathname [-Alrcv] [-Btrcv] [-No_Alrcv] [-No_Btrcv]
[-TRANS_rollback] [-No_TRANS_rollback] [-AICLK] [-No_AICLK]
[-USAGE {TRANS | NON-TRANS}] [-LOCK] [-No_LOCK] [-Write_Access]
Set the attributes of a ROAM file. (EX)
Ref: ROAM Administrator's Guide [53].

Set_Search_Rules [pathname] [-DefaulLT] [-No_System]
[-List_NAME search-rule-name] [-Help]

Use set of rules to search for objects. (IN)
Ref: Programmer's Guide to BIND and EPFs [51].

SET_TIME [-ON list-of-remote-nodes]

Set system clock from remote machine. (EX, NR, OP)

Set_Time_Info [-TIMEZONE timezone-offset] [-HELP]
-DLST [NO | YES [start-date end-date dslt-offset]}

Set time zone information. (OP, IN)
Ref: Operator's Guide to System Commands [35].

SET_VAR gvar-name [=] value

Set the value of a global variable. (EX)
See section on global variables. Ref: PRIMOS Commands Reference Guide [49].

SHARE [pathname] segno [access]

Specify shared segment. (IN, OP)
Omitted pathname => change access only. access: 000-no access, 200-read access, 600-
read/execute access(default), 700-read, write, execute access. segno < '4000. Note:
'OPRpri 1' must be issued before this command can be used on revs prior to 21.0. Caution
should be used when sharing OS segments. Ref: Operator's Guide to System Commands
[35].

SHutdn [pdev1 ... pdevn | ALL] [-FORCE]
SHutdn pdev [-RENAME] packname
SHutdn packname1 [...packnameN] -ON nodename [-FORCE]

Shutdown disk(s) or system. (IN, OP)
-FORCE is used for malfunctioning disk drives. Ref: Operator's Guide to System
Commands [35].

SIZE [pathname] [-NORM] [-HELP]

Print size of file. (EX)
Uses 1024 word record size. -NORM option causes size to be given in 440 word records.
Size on UFDS, segdirs, and acats show number of entries. Ref: PRIMOS Commands
Reference Guide [49].

SLIST [pathname]

Print file to terminal. (EX)
Ref: PRIMOS Commands Reference Guide [49].

SNA_3270 [config-pathname] [-START [-ENTRY_ID entry-id] [options]
-STOP [stop-action] [options]
-STATUS [-NO_WAIT] [options]
-MESSAGE_LEVEL msg-level
-AUTO_STOP delay-time
-NO_AUTO_STOP

Administrator command to invoke and control PRIME/SNA interactive. (EPF, 19.4, OP)

[config-pathname ]
   Server configuration pathname.
-START
   Start the interactive server.
-Entry_ID entry-id
   Interactive configuration name.
-STOP [stop-action ]
   Stop the server. Quit, Idle, Finish, Now
-STATUS -No_Wait -MesSage_Level msg-level Brief, Medium, Detailed -Auto_STOP
delay-time
   Number of minutes to delay automatic logout of Interactive when no user sessions are
   active.
-No_Auto_STOP
-Remote_System rsname
   Remote system name in Server configuration file.
-Remote_System_group rsname
   Remote system group name in Server configuration file.

Ref: PRIME/SNA Operator's Guide [56]

SNA_3270_CONFIG [config-pathname ] [-ENTRY_ID entry-id ] [options ]

Create/edit SNA interactive config files. (EPF, 19.4)

Options:

{-CReate | -EDit | -DisPlay |
 -Listing [listing-pathname ] |
 -SPOOL [spool-options ] |
 -Entry_ID entry-id |
 -No_Wait |
 -Terminal_Type terminal-type }

Ref: PRIME/SNA Administrator's Guide [55]

SNA_PRINT [-NO_LOG]

Start and control SNA interactive printer emulation. (EPF, OP, 19.4)

Internal commands:

START printer-name [spool-options ]
STOP [printer-name ] [-NOW ]
STATUS [printer-name ]
DisPlay
Quit
CANCEL printer-name
{PA1 | PA2 } printer-name
Help

Ref: PRIME/SNA Operator's Guide [56]

SNA_SERVER [config-pathname ] -START [options ]
-STOP [stop-action ] [options ]
-STATUS [-NO_WAIT ] [options ]
-MESSAGE_LEVEL msg-level[options]
-STATISTICS [stats-file] | -NO_STATISTICS

Invoke and control the SNA server. (EPF, OP, 19.4)
options:
-LINE lname
-LINE_GROUP lgroup
-REMOTE_SYSTEM rname
-REMOTE_SYSTEM_GROUP rsname

Ref: PRIME/SNA Operator's Guide [56]

SNA_SERVER_CONFIG [config-pathname] [options]

Create/edit SNA server config file. (EPF, 19.4)

Ref: PRIME/SNA Administrator's Guide [55]

SORT [-BRief] [-SPace] [-MErge] [-TAG | -NONTAG]

Sort files. (EX)
Prompts for: input filename, output filename, number of pairs of starting and ending columns, input pairs of starting and ending columns, reverse order, and data type. If
-MERGE: number of files to be merged followed by input files, one per line.

Multiple input files, file types, and record length information may be specified by using the following keywords. These keywords may be used to bypass the standard dialog. Enter in any order on single line:
-INPUTFILE name
-OUTPUTFILE name
-KEYS n
-INTYPE COMPRESSED
    UNCOMPRESSED
    FIXED
    VARIABLE
-OUTTYPE type
-INLENGTH n
-OUTLENGTH n
-START n
-END n
-DESCENDING
-TYPE code

Ref: PRIMOS Commands Reference Guide [49].

SPAC

See Set_Priority_ACI.

SPL pathname [CE-options]

SPL compiler. (EX, QT)
SPOOL [pathname] [options]
SPOOL -MODify request-number options
SPOOL -CANCEL {PRTn | n | -ALL} [options]
SPOOL -LIST [options]

Print queue manager. (EX)
NB: Pre-rev 21 spoolers will not print files from rev 21 spool queues. File submittal options:

-ALias name Replace user name in header (21.0)
-AS name Replace file name in header
-AT name Same as -ATT
-ATTRIBUTE names Specify device attributes (21.0)
-COB Cobol format (21.0)
-COPIes n No of copies (1 - 99)
-DEfer hh:mm Defer printing to given time
-DISK name Place request in named pre-rev 21 queue
-FORM name Same as -ATT
-FTN Fortran format
-HeaDeR text Use text as page header (21.0)
-LNNumbers Print with line numbers
-NCopy Print from original file location (21.0)
-NEject Suppress form feed at end
-NOFormat Inhibit all format actions
-NOHeader Print without header page
-NO Suppress overprinting (was -CRLF) (21.0)
-Notify Notify user on completion
-NPH Suppress page header in paginate mode (21.0)
-ON node Place request in queue on named node (21.0)
-OPEN Open file in spool queue
-PLOT n Plot raster size (words/scan)
-PROC name PostScript procedure name (21.0)
-RUSH Priority listing (Administrator only)
-SFI Suppress file info in banner (21.0)
-TRUnicate Truncate long lines
-TO page_number Stop at page_number (21.0)
-TUNIT n File unit for -OPEN (default 2)

-MODIFY options:
any of the above excluding: -NOCOPY, -OPEN, -TUNIT

-NODEfer
Cancel deferral

-NORush
Cancel rush priority

-CANCEL options:

-DISK disk-name
Cancel request on pre-21 queue

-ON node
Cancel request in queue on named node

-LIST options:
request-number -USER name
  Only requests for named user

-ATTRIBUTE names
  Only requests with named attributes

-Brief
  Short form report

-DETAil
  More detailed report

-FULL
  Extended form report

-DISK name
  Report pre-rev 21 queue on named partition

-ON node
  Report queue on named node

-ALL Report on all known queues

-NoWait
  Suppress --More-- prompt

Ref: Operator's Guide to the Spooler Subsystem [33].

  [-PRINTBACK] [-MAXERROR]

Statistics Package for the Social Sciences. (EX, OBS)

SPSSX input-file

SPSS-X statistics package. (EX, JM)

SPY

MIDASPLUS information utility. (EX)

SQ

See Set_Quota.

Start [token... [p[a[x[k]][]]=]

Start execution. (IN)
Ref: PRIMOS Commands Reference Guide [49].

START_DSM [options ]

Starts DSM on the system. (EPF, 21.0)

Options:
  -Multi_Node
  -Help [-No_Wait]
START_LSR

Start the login server. (OP, EPF).
Ref: Operator's Guide to System Commands [35].

START_NET [config-filename] [-NODE node-name] [-NT] [-CACHE]

Start network. (EX, OP)
Node-name need not be given if SYSNAM config directive used (21.0). Ref: PRIMENET Guide [45].

START_NTS [config_pathname]

Start the network terminal service server. (OP, EPF)

STARTUp [PROTECT] [nodename] dvno ...

Startup disk(s). (IN, P2)

STATUs [AL] | COnm | DEvice | Disks | ME | NEwork |
PRoject | SEmaphores | SYstem | UNits | USers |

Print user or system status. (IN)
Ref: PRIMOS Commands Reference Guide [49].

STATUS_DSM [options]

Displays status of DSM configurations on nodes. (EPF, 21.0)

Options:
-TTP [TTY | PT45 | PST100 | PT200]
-No_Wait
-Help [-No_Wait]
-USAGE


STOP_DSM [-Help [-No_Wait] | -USAGE]

Shuts down DSM on the system by logging out the DSM processes. (EPF, 21.0)

STOP_LSR

Logs out the login server. (OP, EPF)
Ref: Operator's Guide to System Commands [35].

STOP_NET

Stop network processing. (EX, OP)
Ref: PRIMENET Guide [45].

STOP_NTS

Stops the NTS server. (OP, EPF)
SVcsw [1 | 0]

Set user SVC switch. (IN)
1 => bounce (except class 5), 0 => don't bounce. Ref: *PRIMOS Commands Reference Guide* [49].

SYSLOG

Convert LOGREC file to ASCII file. (EX, OBS)

TA_ADMIN

Start transport agent admin utility for OAS mail. (EX, OBS)

TAP

1-sector octal-mode debugger. (EX, OBS)

TCF [-Host hname | -Autoport n] -Terminal tname [option]

DPTX Transparent connect facility. (EX)
*Option can be:*
-Quit [q-string] (default: TCF$QUIT)
-PA n (n=1, 2, or 3)
-PF n (n=1..12)
-TR
Ref: *Distributed Processing Terminal Executive Guide* [12].

TDOS64

Run PRIMOS II emulator under PRIMOS. (EX, OBS)

TEMPLATE

File construction utility(SPS). (EX, QT)

TERM options

Set terminal characteristics. (EX)
*Options:*
-DISPLAY
-ERASE char
-KILL char
-BREAK (ON | OFF)
-HALF [LF | NOLF]
-FULL
-NOXOFF
-XOFF [-HALF]
Ref: *PRIMOS Commands Reference Guide* [49].

Time

Print time statistics. (IN)
Rev 18: HH'MM logged in, MM'SS CPU time, MM'SS I/O time. Ref: *PRIMOS Commands Reference Guide* [49].
TIMER

OAS timer facility. (EX)

TP [-BE [-ID workstation ] [[-HELP] [-Log_Input pathname ]
[-NO_OVERPRINT] [-NO_OVP] [-RESTORE_CONFIG] [-SAVE_CONFIG]
[-SCRIBE pathname ] [-TERM] [-TTP {PT200 | PST100 | PT45}]]

Start PRIMEWAY subsystem. (CPL)

TP_EXO

Invoke execute-only version of PRIMEWAY. (CPL)

TPLINK -ON {mc | :addr} -ID ws-id [options ]

Start PRIMEWAY network utility. (EX)
Options:
-PORT port
-HELP

TPBE

Transaction Processing Business Environment. (CPL)

TRACE_RO [SYSTEM | USER] [-USERNO user-no ] [-ON] [-OFF] [-STATUS]

Display ROAM actions taken by entire system or a user. (EX)

TRAMLC [TRANSMIT | RECEIVE] filename line [T]

Amic I/O. (EX)
Ref: PRIMOS Commands Reference Guide [49].

Transfer_LOG options

Backs up and restores audit logs. (IN, OP)

TRANSPORT [-LIST] pathname -MT n [options ]

Copy files from disk to tape for transport to another Prime(BRMS). (EX)
Options:
-Cam_To_Dam
-Compatible_Version rev
-HELP [argument ]
-INDEX [pathname ]
-IndeX_Levels [n ]
-LeVeLs [n ]
-No_Query
-REMARK [character-string ]
-SAVE_Protection
-Tty
-VALIDdate
-VeriFy
-VOLID volume-id

and wildcard date (after/before) options. Ref: Data Backup and Recovery Guide [7].
TRANSPORT_RELEASE -MT n [options]

Release a transport tape for reuse (BRMS). (EPF)
Options:
- VOLID name
- No_Query
- REEL n
- HELP [argument]

Ref: Data Backup and Recovery Guide [7].

TRANSPORT_RESTORE source-path [target-path] -MT n [options]

Copy file from tape to disk (BRMS). (EX)
Options:
- VOLID name [name...]
- RECOVER
- INDEX [pathname]
- IndexX_Levels [n] (1 <= n <= 99)
- REEL n (1 <= n <= 255)
- Tty
- Cam_RBF
- Dam_RBF
- From_Logical_Tape n
- To_Logical_Tape n
- MAGSAV
- Written_After [date]
- Written_Before [date]
- From_Save_Number n (1 <= n <= 255)
- To_Save_Number n (1 <= n <= 255)
- No_Query
- Verify
- COMBine
- REPLACE
- HELP

Ref: Data Backup and Recovery Guide [7].

TYPE text

Print text at terminal. (IN)
Ref: PRIMOS Commands Reference Guide [49].

ULOAD

Loader for code from Z8080MA, Z8KMA, and Z80MA. (EX, QT)

Unassign {device | [-ALIAS] MT n [-UNLOAD] | ASYNC -LINE n [-TO m]}

Release peripheral device. (IN)
See ASSIGN for device types. Can unassign device held by another user if issued from console. Ref: PRIMOS Commands Reference Guide [49].

UPCASE inpathname [outpathname]

Translate file to upper case. (EX)
If outputname is not given, output will go on file open on unit 2. Ref: PRIMOS Commands Reference Guide [49].

[-ON nodename ] [-TIMES n ] [-USER]**

Display system resource utilization. (EX)

**USERs**

Print current number of users. (IN)
Ref: PRIMOS Commands Reference Guide [49].

**USRasr usmo**

Connect system ASR to user. (IN, OP)
Must type USRASR in full if usmo is not logged in. USRASR 1 returns console to normal.
Ref: PRIMOS Commands Reference Guide [49].

**VISTA**

Invoke DBMS/QUERY (Obsolete, use DISCOVER). (EX, 19.4)

**VPSD[16]**

Virtual mode PSD. (EX)
Supports V-mode. For internal commands, see PSD.

**VRPG pathname [CE-options]**

RPG II V-mode compiler. (EX)
See Compiler options, 2.7.1, for options. Ref: RPG II V-Mode Compiler Reference Guide [54].

**Vrtssw [sense-switch-setting]**

Set virtual sense switches. (IN)
Ref: PRIMOS Commands Reference Guide [49].

**WORD [document-name] [options]**

Invoke PRIMEWORD word processor. (EX)
Options:

- **-CREATE [format_type]**
  Creates a document with the name optionally with PRIMEWORD format type specified.
- **-PRINT [n] [-DISK new-name | -MENU | -VIEW]**
  Prints the document named.
- **-SPELL [-MENU | -OUTPUT new-name]**
  Checks spelling of a document.
- **-GGLGlossary_name**
  Starts execution of a PRIMEWORD Global Glossary item.
- **-GLOSS glossary_name> [-STEP]**
  Starts execution of a PRIMEWORD Local Glossary item.
- **-NOEXIT**
  Proceeds to PRIMEWORD Main Menu after finishing the specified functions.
Ref: PRIMEWORD Administrator's Guide, [48].

WP_ADMIN

Word Processor administrator. (EX, OBS)
Replaced by OA_ADMIN.

WPS

Word processing system. (EX)

WS1004, WS200UT, WS7200, WSX80, WSGRTS, WSHASP

Control RJE workstations. (EX, OBS)
Replaced by RJOP.

Z80MA

Z80 cross assembler. (EX, QT)

Z8KMA

Z8000 cross assembler. (EX, QT)

2.7.1. Standard compiler options
These options apply to the common backend compilers.

Usage: compiler_name input-file [options ]

compiler_name may be one of: CBL, F77, MODULA, PASCAL, PL1, PL1G, SPL, VRPG.

input-file
Source program name. If "TTY", then take source from terminal.
-32I Generates 32I mode code.
-32IX Generates 32I mode with general register relative instructions.
-64V Generates 64V mode code.
-ALLerrors Override the limit of 100 fatal diagnostics. (CBL)
-Allow_PREconnection Allow use of pre-opened listing or binary output files.
-No_Allow_PREconnection Negation of -Allow_PREconnection.
-BANner Prints column index banner before each non-comment line. (VRPG)
-No_BANner Negation of -BANner.
-BIG Handle segment-spanning data properly when unclear from program itself.
-No_BIG Negation of -BIG.
-Binary tree Specifies binary object file.
-No_Binary Negation of -Binary tree.
-CALCindex
  Calculate index offsets when referenced instead of when SET. (CBL)
-No_CALCindex
  Negation of -CALCindex.
-CLuster
  Cluster routines for optimization. (SPL, PL1, PL1G, PASCAL, F77)
-COMP
  Use full hardware capacity (15 or 31 bits) of COMP fields. (CBL)
-No_COMP
  Negation of -COMP.
-Compiler_DATA [tree]
  Specifies path to non-standard compiler data.
-Conformant_Arrays
  Used for ISO conformance. (PASCAL)
-No_Conformant_Arrays
  Negation of -Conformant_Arrays (PASCAL)
-COPY
  Copies, not originals, of constants are passed by reference. (SPL, PL1, PL1G, PASCAL, MODULA, CBL)
-No_COPY
  Negation of -COPY.
-CORrMap
  Insert into listing a map of CORRESPONDING matches. (CBL)
-No_CORrMap
  Negation of -CORrMap.
-D_StateMenT
  Interpret statements with a "D" in column 1 as compilable source text. (F77)
-No_D_StateMenT
  Interpret statements with a "D" in column 1 as comments. (F77)
-DCivar
  Flags undeclared variables. (F77)
-No_DCivar
  Negation of -DCivar.
-DeBuG
  Generates code for full debugger (DBG) functionality.
-No_DeBuG
  Negation of -DeBuG.
-DO1, -DO
  Performs one-trip DO-loops according to FORTRAN IV standard. (F77)
-No_DO1, -ND0
  Negation of -DO1, -DO.
-DYnm
  Allocates local storage dynamically, opposite of -SAve. (F77)
-EntryTRaCe
  Generate runtime code to display PROGRAM-ID & DATE-COMPILED. (CBL)
-No_EntryTRaCe
  Negation of -EntryTRaCe.
-ERRList
  Produces an errors-only listing.
-No_ERRList
  Negation of -ERRList.
-ERRorFile
  Create a file of diagnostics called <source file name>.CBL.ERROR. (CBL)
-No_ERRorFile
  Negation of -ERRorFile.
-ERRTty
  Outputs error messages to user terminal.
-No_ERRTy
  Negation of -ERRTy.
-Escape34
  Convert from IBM to Prime RPG. (VRPG)
-No_Escape34
  Negation of -Escape34.
-EXPlist
  Expands program listing to include assembler-like output.
-No_EXPlist
  Negation of -EXPlist.
-Extended_Character_Set
  Prime-Extended-Character-Set. (SPL, F77)
-EXTERNAL
  Allows object file to be linked with other Pascal procedures and functions. (PASCAL)
-NO_EXTERNAL
  Negation of -EXTERNAL.
-FIPS dec
  The decimal number signals the FIPS syntax level to check. (CBL)
-FORCEbinary
  Forces binary even if fatal diagnostics were issued. (CBL)
-NO_FORCEbinary
  Negation of -FORCEbinary.
-FRN
  Better accuracy of single-precision floating-point calculations. (SPL, PL1, PL1G, PASCAL, F77, MODULA, CBL)
-NO_FRN
  Negation of -FRN.
-FTN_Entry
  All procedures passed as actual parameters are to be passed in the FTN way. (F77)
-NO_FTEN_Entry
  Negation of -FTN_Entry.
-Full_Help
  Most detailed online help from the compiler.
-Full_OPTimize
  Optimize as much as possible.
-Help
  Produces usage information and option list.
-HEXaddress
  Addresses in the listing file are printed in hexadecimal notation. (CBL)
-NO_HEXaddress
  Negation of -HEXaddress.
-Input tree
  Specifies source input file.
-INTL
  Makes INTEGER default to INTEGER*4. (F77)
-INTS
  Makes INTEGER default to INTEGER*2. (F77)
-LCase
  Distinguishes lower and uppercase characters in the source program.
-Listing [tree]
  Generate source listing. Write it to tree if specified.
-NO_Listing
  Negation of -Listing [tree].
-LOGL
  Makes LOGICAL default to LOGICAL*4. (F77)
-LOGS
  Makes LOGICAL default to LOGICAL*2. (F77)
-MAIN id
  Specifies the main entrypoint of the program, useful with -CLUster. (SPL, PL1, PL1G, PASCAL, F77)

-MAP
  Produce a listing with a map of data and procedure names.

-No_MAP
  Negation of -MAP.

-MAPSort
  Same as -MAP except names are sorted alphabetically. (CBL)

-MAPWide [dec]
  Same as -MAP except use dec character lines instead of 80 (108 assumed if dec not given).

-Max_Growth_percent dec
  Specify optimization space growth limits. (SPL, PL1, PL1G, PASCAL, F77, MODULA, CBL)

-Max_Inline_Statements dec
  Sets threshold procedure size for inline expansion. (SPL, PL1, PL1G, PASCAL, F77, MODULA, VRPG)

-MAXErrors dec
  Specifies the max number of errors to display before compilation abort.

-Nesting
  Adds nesting level numbers in program listing. (SPL, PL1, PL1G, PASCAL, MODULA, VRPG)

-No_Nesting
  Negation of -Nesting.

-OFFset
  Print locations of each executable statement in listing.

-No_OFFset
  Negation of -OFFset.

-Old_Semantic
  Allows non-standard semantics. (PASCAL)

-OLDio
  Allow only I/O constructs allowed by the previous COBOL compiler. (CBL)

-No_OLDio
  Negation of -OLDio.

-Optimize dec
  Specifies the level of optimization to perform.

-Optimization_Selection char
  Specify specific optimization to perform or not.

-OVerFlow
  Enables integer exception checking. (F77, MODULA, PASCAL, PL1, PL1G, SPL)

-No_OVerFlow
  Negation of -OVerflow.

-PBECB
  Load Entry Control Blocks (ECBs) into the procedure frame. (SPL, PL1G, F77, MODULA)

-No_PBECB
  Negation of -PBECB.

-PreFix char
  Prefix the argument to the source file. Used to define modes. (SPL)

-PRODuction
  Generates code for partial debugger functionality.

-No_PRODuction
  Negation of -PRODuction.

-PROFile
  Generates code that will produce execution profile information.

-No_PROFILE
  Negation of -PROFile.

-RANGE
  Generates runtime code that checks subscript ranges.
No_Range
  Negation of -RAnge.
-Range_NonFatal
  Generate non-fatal run-time code to check subscript ranges. (CBL)
-RMARGIn
  Extend Area B of each source line to column 160. (CBL)
-Save
  Allocates local storage statically, opposite of -DYnm. (F77)
-SEQchk
  Checks columns 3-5 for proper sequencing. (VRPG)
-No_SEQchk
  Negation of -SEQchk.
-Set_Defaults
  Set compiler defaults in a compiler data file.
-SIGnalerrors
  Abort at runtime if any overflow errors. (CBL)
-No_SIGnalerrors
  Negation of -SIGnalerrors.
-Slent severity-limit
  Suppress reporting of errors of specified or lower severities.
-SLACKbytes
  Issue diagnostic whenever compiler-generated FILLER is inserted. (CBL)
-No_SLACKbytes
  Negation of -SLACKbytes.
-Source tree
  Specifies source input file.
-SPACE
  Prefer space over time in optimization.
-STDAnnard
  Warning of variances from language standard. (PASCAL, F77, MODULA)
-No_STDAnnard
  Negation of -STDAnnard.
-STATistics
  Displays compilation statistics at terminal.
-No_STATistics
  Negation of -STATistics.
-STATUS
  Displays statement types on terminal as parsed. (VRPG)
-No_STATUS
  Negation of -STATUS.
-Store_Owner_Field
  Store identity of called routines in stack.
-No_Store_Owner_Field
  Negation of -Store_Owner_Field.
-SYNtaxmsg
  Print "syntax checking suspended/resumed" messages. (CBL)
-No_SYNtaxmsg
  Negation of -SYNtaxmsg.
-SYM tree
  Specify directory into which Modula-2 Definition Symbol files go. (MODULA)
-TIME
  Prefer time over space in optimization.
-TPs
  Generate errors for Transaction Processing System. (CBL)
-TRUNCdiags
  Issue diagnostics for truncated result. (CBL)
-No_TRUNCdiags
  Negation of -TRUNCdiags.
-TTrace
  Specifies that the Time Trace routines are to be called on entry to procedures and begin blocks.
-NTTrace
  Specifies that the Time Trace routines will not be called.
-UPcase
  Map source program to uppercase (except quoted literals).
-XRef
  Produce listing with cross reference of data/procedure names.
-No_XRef
  Negation of -XRef.
-XREFS
  Specifies that the xref map contain only the symbols that are actually used. (SPL, F77, MODULA)
-XRefSort
  Like -XREF except sorted alphabetically. (CBL)
3. ARCHITECTURE

3.1. Argument Pointer (AP)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit</td>
<td>Bit Number</td>
<td>170000</td>
<td>F000</td>
</tr>
<tr>
<td>I</td>
<td>Indirect</td>
<td>004000</td>
<td>0800</td>
</tr>
<tr>
<td>BR</td>
<td>Base Register: 00 - PB 01 - SB 10 - LB 11 - XB</td>
<td>001400</td>
<td>0300</td>
</tr>
<tr>
<td>L</td>
<td>Last AP in argument list</td>
<td>000200</td>
<td>0040</td>
</tr>
<tr>
<td>S</td>
<td>Store this argument</td>
<td>000100</td>
<td>0020</td>
</tr>
<tr>
<td>Word</td>
<td>Word displacement from Base Register</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.2. Cache entries

450, 250-II, 550-II, 2250:

<table>
<thead>
<tr>
<th>PPN</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>750, 850, 2350, 2450, 2550, 9650:</td>
<td></td>
</tr>
<tr>
<td>PPN</td>
<td>Contents</td>
</tr>
<tr>
<td>2755, 4150, 6350, 6550, 9750, 9950, 9955:</td>
<td></td>
</tr>
</tbody>
</table>

3.3. Checks

<table>
<thead>
<tr>
<th>Header</th>
<th>Handler</th>
<th>Type of check</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/200</td>
<td>4/204</td>
<td>Fail, Environment on 9650, 9750, 9950, 9955.</td>
</tr>
<tr>
<td>4/270</td>
<td>4/274</td>
<td>Memory parity(ECCU, ECCC)</td>
</tr>
<tr>
<td>4/300</td>
<td>4/304</td>
<td>Machine check(MCHK)</td>
</tr>
<tr>
<td>4/310</td>
<td>4/314</td>
<td>Missing memory module(MMOD)</td>
</tr>
<tr>
<td>4/320</td>
<td>4/324</td>
<td>Recoverable machine check (9955)</td>
</tr>
</tbody>
</table>

On entry to fault handler, mode=64V, MCM=0 for all but ECCC, for which MCM = MCM-at-check - 1, and recoverable machine check(MCM = 2).

MMOD interrupts any other check in progress.
MCHK and ECCU interrupt ECCU in progress if MCM = 2 (QUIET).

3.3.1. Check header

<table>
<thead>
<tr>
<th></th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Keys</td>
</tr>
<tr>
<td>4</td>
<td>Modals</td>
</tr>
</tbody>
</table>

3.4. Concealed Stack/Queue
Valid only between time of fault and subsequent CALF instruction.

| PCB+74 | FIRST |
| PCB+75 | NEXT |
| PCB+76 | LAST |

<table>
<thead>
<tr>
<th></th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>KEYS</td>
</tr>
<tr>
<td>3</td>
<td>FCODE</td>
</tr>
<tr>
<td>4</td>
<td>FADDR</td>
</tr>
</tbody>
</table>

PB, KEYS are those of procedure causing the fault.

3.5. Diagnostic Status Word (DSW)
Register mapping:

| DSWRMA | R34 |
| DSWSTAT | R35 |
| DSWPB | R36 |
| DSWPARITY | R27 |
3.5.1. DSWSTAT

3.5.1.1. 6350, 6550

DSWSTATH:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check immediate</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>2</td>
<td>Machine check</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>3</td>
<td>Memory parity (ECC)</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>Missing memory module</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>5</td>
<td>E (Execute) unit parity error</td>
<td>004000</td>
<td>0800</td>
</tr>
<tr>
<td>6</td>
<td>IS (Instruction/Storage) unit reported parity error</td>
<td>002000</td>
<td>0400</td>
</tr>
<tr>
<td>7</td>
<td>CS (Control Store) unit reported parity error</td>
<td>001000</td>
<td>0200</td>
</tr>
<tr>
<td>8</td>
<td>MC (Memory Controller) reported error</td>
<td>000400</td>
<td>0100</td>
</tr>
<tr>
<td>9</td>
<td>ECCU - ECC Uncorrectable (if bit 3 on)</td>
<td>000200</td>
<td>0080</td>
</tr>
<tr>
<td>10</td>
<td>ECCU - ECC Correctable (if bit 3 on)</td>
<td>000100</td>
<td>0040</td>
</tr>
<tr>
<td>11</td>
<td>Reserved</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>12</td>
<td>RCM parity error reported by CS board</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>13-14</td>
<td>RP backup count (subtract from DSWPB)</td>
<td>000014</td>
<td>000C</td>
</tr>
<tr>
<td>15</td>
<td>Check occurred during DMx operation</td>
<td>000002</td>
<td>0002</td>
</tr>
<tr>
<td>16</td>
<td>Check occurred during I/O</td>
<td>000001</td>
<td>0001</td>
</tr>
</tbody>
</table>

DSWSTATL:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-7</td>
<td>ECCC Syndrome(^1):</td>
<td>177000</td>
<td>FE00</td>
</tr>
<tr>
<td></td>
<td>000 No error</td>
<td></td>
<td></td>
</tr>
<tr>
<td>002</td>
<td>CB0 057 16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>004</td>
<td>CB1 061 18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>006</td>
<td>CB2 062 19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>007</td>
<td>1 064 21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>010</td>
<td>CB3 067 24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>020</td>
<td>CB4 070 25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>040</td>
<td>CB5 073 28</td>
<td></td>
<td></td>
</tr>
<tr>
<td>043</td>
<td>4 075 30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>045</td>
<td>6 076 31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>046</td>
<td>7 100 CB6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>051</td>
<td>10 141 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>052</td>
<td>11 142 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>054</td>
<td>13 144 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Low order address bit of module in error</td>
<td>000400</td>
<td>0100</td>
</tr>
<tr>
<td>9</td>
<td>DSWRMA is invalid</td>
<td>000200</td>
<td>0080</td>
</tr>
<tr>
<td>10</td>
<td>Recoverable machine check</td>
<td>000100</td>
<td>0040</td>
</tr>
<tr>
<td>11</td>
<td>Hard error (permanent error that should be fixed)</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>12</td>
<td>Unused</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>13</td>
<td>Internal microcode error. Algorithm code in DSWRMA</td>
<td>000010</td>
<td>0008</td>
</tr>
<tr>
<td>14</td>
<td>Processor in dual configuration</td>
<td>000004</td>
<td>0004</td>
</tr>
<tr>
<td>15</td>
<td>Slave processor reported error</td>
<td>000002</td>
<td>0002</td>
</tr>
<tr>
<td>16</td>
<td>Memory bus that had the error (&lt;&gt;bit 14 of address)</td>
<td>000001</td>
<td>0001</td>
</tr>
</tbody>
</table>

\(^1\)MB - Multibit; RP - Right Parity; CBn - Check Bit n
### 3.5.1.2. 9750, 9950, 9955

#### DSWSTATH:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check immediate</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>2</td>
<td>Machine check</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>3</td>
<td>Memory parity (ECC)</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>Missing memory module</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>5</td>
<td>E1 board parity error</td>
<td>004000</td>
<td>0800</td>
</tr>
<tr>
<td>6</td>
<td>F or S unit reported parity error</td>
<td>002000</td>
<td>0400</td>
</tr>
<tr>
<td>7</td>
<td>I unit reported parity error</td>
<td>001000</td>
<td>0200</td>
</tr>
<tr>
<td>8</td>
<td>Memory controller reported parity error</td>
<td>000400</td>
<td>0100</td>
</tr>
<tr>
<td>9</td>
<td>ECCU - ECC Uncorrectable (if bit 3 on)</td>
<td>000200</td>
<td>0080</td>
</tr>
<tr>
<td>10</td>
<td>ECCU - ECC Correctable (if bit 3 on)</td>
<td>000100</td>
<td>0040</td>
</tr>
<tr>
<td>11</td>
<td>Control store reported parity error</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>12</td>
<td>RCM parity error reported by CS board</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>13-14</td>
<td>RP backup count (subtract from DSWPB)</td>
<td>000014</td>
<td>000C</td>
</tr>
<tr>
<td>15</td>
<td>Check occurred during DMx operation</td>
<td>000002</td>
<td>0002</td>
</tr>
<tr>
<td>16</td>
<td>Check occurred during I/O</td>
<td>000001</td>
<td>0001</td>
</tr>
</tbody>
</table>

#### DSWSTATL:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-7</td>
<td>ECCC Syndrome$^2$</td>
<td>177000</td>
<td>FE00</td>
</tr>
<tr>
<td></td>
<td>000 No error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>001 CB0 057 16 147 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>002 CB1 061 18 150 9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>004 CB2 062 19 153 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>007 1 064 21 155 14</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>010 CB3 067 24 156 15</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>020 CB4 070 25 160 17</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>040 CB5 073 28 163 20</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>043 4 075 30 165 22</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>045 6 076 31 166 23</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>046 7 100 CB6 171 26</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>051 10 141 2 172 27</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>052 11 142 3 174 29</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>054 13 144 5 177 32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Low order address bit of module in error</td>
<td>000400</td>
<td>0100</td>
</tr>
<tr>
<td>9</td>
<td>DSWRMA is invalid</td>
<td>000200</td>
<td>0080</td>
</tr>
<tr>
<td>10</td>
<td>9955: recoverable mach check</td>
<td>000100</td>
<td>0040</td>
</tr>
<tr>
<td>11-16</td>
<td>Unused</td>
<td>000077</td>
<td>003F</td>
</tr>
</tbody>
</table>

---

$^2$MB - Multibit; RP - Right Parity; CBn - Check Bit n
### 3.5.1.3. 2250, 2550, 9650

#### DSWSTATH:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Check immediate</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>2</td>
<td>Machine check</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>3</td>
<td>Memory parity (ECC)</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>Missing memory</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>5-7</td>
<td>Machine Check Code (Valid if bit 8=1):</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 none</td>
<td>007000</td>
<td>0E00</td>
</tr>
<tr>
<td></td>
<td>1 Peripheral Data (BPD) parity</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 Memory Data (BMD) parity</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 Cache Data (RCD)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 Peripheral Addr (BPA) parity</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 STLB parity</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 Memory Address (BMA) parity</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 A-board parity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Not control unit (RCM) parity</td>
<td>000400</td>
<td>0100</td>
</tr>
<tr>
<td>9</td>
<td>ECCU - ECC Uncorrectable error</td>
<td>000200</td>
<td>0080</td>
</tr>
<tr>
<td>10</td>
<td>ECCC - ECC Correctable error</td>
<td>000100</td>
<td>0040</td>
</tr>
<tr>
<td>11</td>
<td>RP backup count (BUP) is invalid</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>12-14</td>
<td>RP backup count (subtract from DSWPB)</td>
<td>000034</td>
<td>001C</td>
</tr>
<tr>
<td>15</td>
<td>Check occurred during DMx</td>
<td>000002</td>
<td>0002</td>
</tr>
<tr>
<td>16</td>
<td>Check occurred during I/O operation</td>
<td>000001</td>
<td>0001</td>
</tr>
</tbody>
</table>

#### DSWSTATL:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>ECCC Syndrome(^3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>000xxx MB</td>
<td>100xxx MB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>004xxx MB</td>
<td>104xxx 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>010xxx MB</td>
<td>110xxx MB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>014xxx 15</td>
<td>114xxx 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>020xxx MB</td>
<td>120xxx MB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>024xxx 14</td>
<td>124xxx 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>030xxx 13</td>
<td>130xxx 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>034xxx 9</td>
<td>134xxx CB2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>040xxx MB</td>
<td>140xxx 8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>044xxx MB</td>
<td>144xxx 6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>050xxx MB</td>
<td>150xxx 5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>054xxx 12</td>
<td>154xxx CB5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>060xxx 16</td>
<td>160xxx 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>064xxx 11</td>
<td>164xxx CB4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>070xxx 10</td>
<td>170xxx CB3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>074xxx RP,CB1</td>
<td>174xxx No error</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>OP -- Overall Parity</td>
<td>002000</td>
<td>0400</td>
</tr>
<tr>
<td>7</td>
<td>Unused</td>
<td>001000</td>
<td>0200</td>
</tr>
<tr>
<td>8</td>
<td>Low order address bit of module in error</td>
<td>000400</td>
<td>0100</td>
</tr>
<tr>
<td>9</td>
<td>DSWRMA contents invalid</td>
<td>000200</td>
<td>0080</td>
</tr>
<tr>
<td>10-16</td>
<td>Unused</td>
<td>000177</td>
<td>007F</td>
</tr>
</tbody>
</table>

\(^3\)MB - Multibit; RP - Righ Parity; CB\(n\) - Check Bit \(n\)
### 3.5.1.4. All other 50 series

**DSWSTATH:**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CI - Check Immediate</td>
<td>100 000</td>
<td>8000</td>
</tr>
<tr>
<td>2</td>
<td>MC - Machine Check</td>
<td>040 000</td>
<td>4000</td>
</tr>
<tr>
<td>3</td>
<td>MP - Memory Parity (ECC)</td>
<td>020 000</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>MM - Missing Memory</td>
<td>010 000</td>
<td>1000</td>
</tr>
<tr>
<td>5-7</td>
<td>Machine Check Code (Valid if bit 8=1):</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 Peripheral Data (BPD) Output</td>
<td>007 000</td>
<td>0E00</td>
</tr>
<tr>
<td></td>
<td>1 Peripheral Addr (BPA) Input</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 Memory Data (BMD) Output</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 Cache Data (RCD)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 Peripheral Addr (BPA) Output</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 RDX-BPD Input</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 Memory Address (BMA)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 Register File (RF)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Not RCM Parity (P500, XCS)</td>
<td>000 400</td>
<td>0100</td>
</tr>
<tr>
<td>9</td>
<td>ECCU -- ECC Uncorrectable Error</td>
<td>000 200</td>
<td>0080</td>
</tr>
<tr>
<td>10</td>
<td>ECCC -- ECC Correctable Error</td>
<td>000 100</td>
<td>0040</td>
</tr>
<tr>
<td>11</td>
<td>BUP Invalid -- RP Backup Count Invalid</td>
<td>000 040</td>
<td>0020</td>
</tr>
<tr>
<td>12-14</td>
<td>RP Backup Count – Sub from DSWPB</td>
<td>000 034</td>
<td>001C</td>
</tr>
<tr>
<td>15</td>
<td>Check During DMX</td>
<td>000 002</td>
<td>0002</td>
</tr>
<tr>
<td>16</td>
<td>IO Bus -- DMX, PIO, μ-code check</td>
<td>000 001</td>
<td>0001</td>
</tr>
</tbody>
</table>

**DSWSTATL:**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>ECCC Syndrome&lt;sup&gt;4&lt;/sup&gt;</td>
<td>174 000</td>
<td>F800</td>
</tr>
<tr>
<td></td>
<td>000xxx MB</td>
<td>100xxx MB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>004xxx MB</td>
<td>104xxx 7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>010xxx MB</td>
<td>110xxx MB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>014xxx 15</td>
<td>114xxx 3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>020xxx MB</td>
<td>120xxx MB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>024xxx 14</td>
<td>124xxx 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>030xxx 13</td>
<td>130xxx 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>034xxx 9</td>
<td>134xxx CB2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>040xxx MB</td>
<td>140xxx 8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>044xxx MB</td>
<td>144xxx 6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>050xxx MB</td>
<td>150xxx 5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>054xxx 12</td>
<td>154xxx CB5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>060xxx 16</td>
<td>160xxx 4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>064xxx 11</td>
<td>164xxx CB4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>070xxx 10</td>
<td>170xxx CB3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>074xxx RP, CB1</td>
<td>174xxx No error</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>OP -- Overall Parity</td>
<td>002 000</td>
<td>0400</td>
</tr>
<tr>
<td>7</td>
<td>Unused</td>
<td>001 000</td>
<td>0200</td>
</tr>
<tr>
<td>8</td>
<td>Low order address bit of module in error</td>
<td>000 400</td>
<td>0100</td>
</tr>
<tr>
<td>9</td>
<td>RMA Invalid</td>
<td>000 200</td>
<td>0080</td>
</tr>
<tr>
<td>10</td>
<td>Unused</td>
<td>000 100</td>
<td>0040</td>
</tr>
<tr>
<td>11-16</td>
<td>U-Verify Test Number</td>
<td>000 077</td>
<td>003F</td>
</tr>
</tbody>
</table>

<sup>4</sup> MB - Multibit; RP - Right Parity; CBn - Check Bit n
### 3.5.2. DSWPARITY

#### 3.5.2.1. 6350, 6550

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CS: I/O parity error</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>2</td>
<td>CS: BPD high side, left parity error</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>3</td>
<td>CS: BPD high side, right parity error</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>CS: BPD low side, left parity error</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>5</td>
<td>CS: BPD low side, right parity error</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>6</td>
<td>PIOS: BPA parity error</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>7</td>
<td>PIOS: BPD parity error</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>8-10</td>
<td>CS: RCC parity error: FRCCPE(n+1)</td>
<td>034000</td>
<td>0700</td>
</tr>
<tr>
<td>11</td>
<td>Unused?</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>12</td>
<td>CS: Decode net high side parity error</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>13</td>
<td>CS: Decode net low side parity error</td>
<td>000010</td>
<td>0006</td>
</tr>
<tr>
<td>14-16</td>
<td>E: parity error</td>
<td>000007</td>
<td>0007</td>
</tr>
<tr>
<td>17</td>
<td>MC: lost error</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>18</td>
<td>Memory address shift control:</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td></td>
<td>0 8 mbyte slot decode</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 16 mbyte slot decode</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Memory array number 1 reported error</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>20</td>
<td>Memory array number 2 reported error</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>21</td>
<td>Memory array number 3 reported error</td>
<td>004000</td>
<td>0400</td>
</tr>
<tr>
<td>22</td>
<td>Memory array number 4 reported error</td>
<td>002000</td>
<td>0200</td>
</tr>
<tr>
<td>23</td>
<td>Memory array number 5 reported error</td>
<td>001000</td>
<td>0100</td>
</tr>
<tr>
<td>24</td>
<td>Memory array number 6 reported error</td>
<td>000400</td>
<td>0040</td>
</tr>
<tr>
<td>25</td>
<td>Memory array number 7 reported error</td>
<td>000200</td>
<td>0020</td>
</tr>
<tr>
<td>26</td>
<td>Memory array number 8 reported error</td>
<td>000100</td>
<td>0010</td>
</tr>
<tr>
<td>27</td>
<td>MC: BB parity error</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>28</td>
<td>MC: BD parity error</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>29</td>
<td>MC: BIP in parity error</td>
<td>000010</td>
<td>0008</td>
</tr>
<tr>
<td>30</td>
<td>MC: BIP out parity error</td>
<td>000004</td>
<td>0004</td>
</tr>
<tr>
<td>31</td>
<td>MC: memory time-out error</td>
<td>000002</td>
<td>0002</td>
</tr>
<tr>
<td>32</td>
<td>MC: CIT error</td>
<td>000001</td>
<td>0001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IS: BDH left parity error</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>2</td>
<td>IS: BDH right parity error</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>3</td>
<td>IS: BDL left parity error</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>IS: BDL right parity error</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>5-8</td>
<td>Unused</td>
<td>074000</td>
<td>0F00</td>
</tr>
<tr>
<td>9</td>
<td>Fatal cache parity error</td>
<td>002000</td>
<td>0200</td>
</tr>
<tr>
<td>10</td>
<td>IS: Branch Cache recoverable error</td>
<td>001000</td>
<td>0100</td>
</tr>
<tr>
<td>11</td>
<td>IS: cache data parity error: Element B, even data, low byte</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>12</td>
<td>IS: cache data parity error: Element B, odd data, low byte</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>13</td>
<td>IS: cache data parity error: Element A, even data, low byte</td>
<td>000010</td>
<td>0008</td>
</tr>
</tbody>
</table>
### Bit Description | Octal | Hex
---|---|---
14 | IS: cache data parity error: Element A, odd data, low byte | 000004 | 0004
15 | IS: cache index parity error: Element B, low byte | 000002 | 0002
16 | IS: cache index parity error: Element B, high byte | 000001 | 0001
17 | IS: cache data parity error: Element A, even data, high byte | 100000 | 8000
18 | IS: cache data parity error: Element A, odd data, high byte | 040000 | 4000
19 | IS: cache data parity error: Element B, even data, high byte | 020000 | 2000
20 | IS: cache data parity error: Element B, odd data, high byte | 010000 | 1000
21 | IS: cache index parity error: Element A, low byte | 004000 | 0800
22 | IS: cache index parity error: Element A, high byte | 002000 | 0400
23 | IS: STLB parity error: Element B, physical addr, low byte | 001000 | 0200
24 | IS: STLB parity error: Element A, physical addr, low byte | 000400 | 0100
25 | IS: STLB parity error: Element B, access bits | 000200 | 0080
26 | IS: STLB parity error: Element B, process ID | 000100 | 0040
27 | IS: STLB parity error: Element B, virtual address tag | 000040 | 0020
28 | IS: STLB parity error: Element A, physical addr, high byte | 000020 | 0010
29 | IS: STLB parity error: Element B, physical addr, high byte | 000010 | 0008
30 | IS: STLB parity error: Element A, access bits | 000004 | 0004
31 | IS: STLB parity error: Element A, process ID | 000002 | 0002
32 | IS: STLB parity error: Element A, virtual address tag | 000001 | 0001

### 3.5.2.2. 9750, 9950, 9955

| Bit | Description | Octal | Hex |
---|---|---|---|
1 | RCC parity error | 100000 | 8000 |
2 | I/O parity error | 040000 | 4000 |
3-8 | Parity error code: | 037400 | 3F00 |
6 | Encoding of error bits 1-8 | 002000 | Logical OR of bits 1-8 |
7 | Error bit 9 | 001000 |
8 | 000400 | 0 |
3 | Left byte of BPA or BPD | 020000 |
4 | Right byte of BPA or BPD | 010000 |
5 | CPU detected BPD parity error | 004000 |
6 | CPU detected BPA parity error | 002000 |
7 | Cntlr detected BPD par error | 001000 |
8 | Cntlr detected BPA par error | 000400 |
9 | Unused | 000200 | 0080 |
10 | E1 board detected BBH left byte error | 000100 | 0040 |
11 | E1 board detected BBH right byte error | 000040 | 0020 |
12 | E1 board detected BBL left byte error | 000020 | 0010 |
13 | E1 board detected BBL right byte error | 000010 | 0008 |
14 | E1 board detected BAH right byte error | 000004 | 0004 |
<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>E1 board detected BAL right byte error</td>
<td>000002</td>
<td>0002</td>
</tr>
<tr>
<td>16</td>
<td>E1 board detected BAE right byte error</td>
<td>000001</td>
<td>0001</td>
</tr>
<tr>
<td>17</td>
<td>BD parity error (memory control unit)</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>18</td>
<td>Latched memory data parity error (MCU)</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>20</td>
<td>010000 LMDH left byte</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>004000 LMMD left byte</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>002000 LMDL left byte</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>001000 LMDL right byte</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Latched memory addr parity error (MCU)</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>20</td>
<td>010000 MCADDR High byte</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>004000 MCADDR Low byte</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>002000 MCADDR Extended byte</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>001000 Unused</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>ECCU detected by memory control unit</td>
<td>000400</td>
<td>0100</td>
</tr>
<tr>
<td>25</td>
<td>i-unit error</td>
<td>000200</td>
<td>0080</td>
</tr>
<tr>
<td>26-28</td>
<td>i-unit error location:</td>
<td>000160</td>
<td>0070</td>
</tr>
<tr>
<td></td>
<td>xxx00x No error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xxx02x Unused</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xxx04x Unused</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xxx06x Decode net, right byte</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xxx10x Decode net, left byte</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xxx12x Base register file high</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xxx14x Base register file low</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xxx17x Index register file</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>For S unit error</td>
<td>000010</td>
<td>0008</td>
</tr>
<tr>
<td>30-32</td>
<td>For S unit error bits</td>
<td>000007</td>
<td>0007</td>
</tr>
</tbody>
</table>

For S unit error bits:

- **9955:**
  - 0 No error
  - 1 LPID out of STLB in error
  - 2 LBPA out of STLB in error
  - 3 LBVA out of STLB in error
  - 4 ARR out of STLB in error
  - 5 Cache index
  - 6 Cache data, high side
  - 7 Cache data, low side
- **9750, 9950:**
  - 0 PID or STLB control bits
  - 1 LBPA out of STLB in error
  - 2 Cache index, right 16 bits
  - 3 Cache index, left 16 bits
  - 4 Cache data, high side
  - 5 Cache data, low side
  - 6 LBVA out of STLB in error
  - 7 Branch cache parity error
### 3.5.2.3. 2550, 9650

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Last memory operation was a wide word</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>2</td>
<td>Last memory operation was interleaved</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>3</td>
<td>STL8 parity error</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>Cache index parity error</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>5</td>
<td>Cache data even parity error</td>
<td>004000</td>
<td>0400</td>
</tr>
<tr>
<td>6</td>
<td>Cache data odd parity error</td>
<td>002000</td>
<td>0200</td>
</tr>
<tr>
<td>7</td>
<td>BMD backplane parity error</td>
<td>001000</td>
<td>0100</td>
</tr>
<tr>
<td>8</td>
<td>BPD backplane parity error</td>
<td>000400</td>
<td>0040</td>
</tr>
<tr>
<td>9</td>
<td>RFH left byte parity error</td>
<td>000200</td>
<td>0020</td>
</tr>
<tr>
<td>10</td>
<td>RFH right byte parity error</td>
<td>000100</td>
<td>0010</td>
</tr>
<tr>
<td>11</td>
<td>RFL left byte parity error</td>
<td>000040</td>
<td>0004</td>
</tr>
<tr>
<td>12</td>
<td>RFL right byte parity error</td>
<td>000020</td>
<td>0002</td>
</tr>
<tr>
<td>13</td>
<td>RFH parity error during late cycle</td>
<td>000000</td>
<td>0000</td>
</tr>
<tr>
<td>14</td>
<td>RFL parity error during late cycle</td>
<td>000004</td>
<td>0004</td>
</tr>
<tr>
<td>15</td>
<td>BMD or BPD per err read into A-board</td>
<td>000002</td>
<td>0002</td>
</tr>
<tr>
<td>16</td>
<td>BMA or BPA per err read into A-board</td>
<td>000001</td>
<td>0001</td>
</tr>
<tr>
<td>17-32</td>
<td>Reserved.</td>
<td>177777</td>
<td>FFFF</td>
</tr>
</tbody>
</table>

### 3.5.2.4. 750, 850

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RPA parity error, type 1</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>2</td>
<td>RPA parity error, type 2</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>3</td>
<td>Burst-mode DMx parity error</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>DMx parity on output if 1, on input if 0.</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>5-7</td>
<td>J board parity error:</td>
<td>007000</td>
<td>0E00</td>
</tr>
<tr>
<td></td>
<td>0 - peripheral reports BPD error(output)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - base register file high</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - memory reports BMD error (write)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 - prefetch buffer address</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 - peripheral reports BPA error (output)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 - base register file low</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 - memory reports BMA error</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 - prefetch buffer instruction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>RCM parity error, if no board error</td>
<td>000400</td>
<td>0100</td>
</tr>
<tr>
<td>9</td>
<td>ECC8 error (uncorrectable)</td>
<td>000200</td>
<td>0080</td>
</tr>
<tr>
<td>10</td>
<td>Prefetch board error</td>
<td>000100</td>
<td>0040</td>
</tr>
<tr>
<td>11</td>
<td>BPA input parity error</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>12</td>
<td>RDX parity error</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>13</td>
<td>Register file parity error</td>
<td>000010</td>
<td>0008</td>
</tr>
<tr>
<td>14</td>
<td>REA parity error</td>
<td>000004</td>
<td>0004</td>
</tr>
<tr>
<td>15</td>
<td>DMx cycle parity error</td>
<td>000002</td>
<td>0002</td>
</tr>
<tr>
<td>16</td>
<td>AP board parity error</td>
<td>000001</td>
<td>0001</td>
</tr>
<tr>
<td>17</td>
<td>C board parity error</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>18</td>
<td>BMD input parity error, even word</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>19</td>
<td>BMD input parity error, odd word</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>20</td>
<td>Missing memory module</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>21</td>
<td>BMA parity error</td>
<td>004000</td>
<td>0800</td>
</tr>
<tr>
<td>22</td>
<td>RMA was incremented at time of error</td>
<td>002000</td>
<td>0400</td>
</tr>
<tr>
<td>23</td>
<td>BMA15 indicator</td>
<td>001000</td>
<td>0200</td>
</tr>
<tr>
<td>24</td>
<td>BMA16 indicator</td>
<td>000400</td>
<td>0100</td>
</tr>
<tr>
<td>25</td>
<td>ECCU error on cache miss</td>
<td>000200</td>
<td>0080</td>
</tr>
<tr>
<td>26</td>
<td>ECC error on cache miss</td>
<td>000100</td>
<td>0040</td>
</tr>
<tr>
<td>27</td>
<td>Cache index parity error</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>28</td>
<td>Cache data odd word parity error</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>29</td>
<td>Cache data even word parity error</td>
<td>000010</td>
<td>0008</td>
</tr>
<tr>
<td>Bit</td>
<td>Description</td>
<td>Octal</td>
<td>Hex</td>
</tr>
<tr>
<td>-----</td>
<td>---------------------------</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>30</td>
<td>Cache cycle purpose:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0  - prefetch</td>
<td>000004</td>
<td>0004</td>
</tr>
<tr>
<td></td>
<td>1  - execute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31-32</td>
<td>Unused</td>
<td>000003</td>
<td>0003</td>
</tr>
</tbody>
</table>

3.5.3. DSWRMA
Memory address register. Valid on: ECCU, ECCC, recoverable error (9955), or missing memory.

3.5.3.1. 6350, 6550
32 bit virtual address. On ECCC, ECCU or MISMOD: DSWRMAH = PPN of failing physical memory location; DSWRMAL = 0.

3.5.3.2. 9955
32 bit virtual address. Cleared on cache parity error.

3.5.3.3. 9750, 9950
Bits 1-13 of 23 bit physical address.

3.5.3.4. All other 50 series
32 bit virtual address.

3.5.4. DSWPB
Extended program counter. Always valid.

3.6. Descriptor Table Address Register (DTAR)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Size Number of SDWs in table</td>
<td>177700</td>
<td>FFC0</td>
</tr>
<tr>
<td></td>
<td>SDTU Bits 1-6 of Physical Addr of Table</td>
<td>000077</td>
<td>003F</td>
</tr>
<tr>
<td></td>
<td>B Same as bit 18</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td></td>
<td>SDTL Bits 7-21 of table physical address (Bit 22 taken as zero.)</td>
<td>077777</td>
<td>7FFF</td>
</tr>
</tbody>
</table>
3.7. Entry Control Block (ECB)

<table>
<thead>
<tr>
<th></th>
<th>Procedure</th>
<th></th>
<th>Base</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Stack frame size</td>
<td>(0 - use current)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Stack root segment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Disp of arglist in s.f.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Number of arguments</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Link</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Base</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Keys</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Locations '11 through '17 are set to 0.

3.8. Faults

Locations in current register set:
FCODEH: 26H
FADDR: 27

<table>
<thead>
<tr>
<th>Fault</th>
<th>#</th>
<th>Offset</th>
<th>Vect</th>
<th>FCODEH</th>
<th>FADDR</th>
<th>Ring</th>
<th>Saved</th>
<th>PB</th>
</tr>
</thead>
<tbody>
<tr>
<td>RXM</td>
<td>0</td>
<td>0</td>
<td>62</td>
<td>-</td>
<td>addr</td>
<td>current</td>
<td>0</td>
<td>backed</td>
</tr>
<tr>
<td>PROCESS</td>
<td>1</td>
<td>4</td>
<td>63</td>
<td>ABFLAGS</td>
<td>-</td>
<td>addr</td>
<td>0</td>
<td>current</td>
</tr>
<tr>
<td>PAGE</td>
<td>2</td>
<td>10</td>
<td>64</td>
<td>-</td>
<td>addr</td>
<td>0</td>
<td>current</td>
<td>backed</td>
</tr>
<tr>
<td>SVC</td>
<td>3</td>
<td>14</td>
<td>65</td>
<td>-</td>
<td>-</td>
<td>current</td>
<td>backed</td>
<td></td>
</tr>
<tr>
<td>ULL</td>
<td>4</td>
<td>20</td>
<td>66</td>
<td>cur PBL</td>
<td>addr</td>
<td>current</td>
<td>backed</td>
<td></td>
</tr>
<tr>
<td>SEMAPHORE</td>
<td>5</td>
<td>24</td>
<td>67</td>
<td>undfl 0</td>
<td>sem addr</td>
<td>0</td>
<td>backed</td>
<td></td>
</tr>
<tr>
<td>ILL</td>
<td>10</td>
<td>40</td>
<td>72</td>
<td>cur PBL</td>
<td>addr</td>
<td>current</td>
<td>backed</td>
<td></td>
</tr>
<tr>
<td>ACCESS</td>
<td>11</td>
<td>44</td>
<td>73</td>
<td>code</td>
<td>addr</td>
<td>0</td>
<td>current</td>
<td>backed</td>
</tr>
<tr>
<td>ARITH</td>
<td>12</td>
<td>54</td>
<td>74</td>
<td>code</td>
<td>addr</td>
<td>0</td>
<td>current</td>
<td>backed</td>
</tr>
<tr>
<td>STACK</td>
<td>13</td>
<td>54</td>
<td>75</td>
<td>code</td>
<td>addr</td>
<td>0</td>
<td>current</td>
<td>backed</td>
</tr>
<tr>
<td>SEGMENT</td>
<td>14</td>
<td>60</td>
<td>76</td>
<td>code</td>
<td>addr</td>
<td>0</td>
<td>current</td>
<td>backed</td>
</tr>
<tr>
<td>POINTER</td>
<td>15</td>
<td>64</td>
<td>77</td>
<td>code</td>
<td>ptr addr</td>
<td>0</td>
<td>current</td>
<td>backed</td>
</tr>
</tbody>
</table>

Code at offset is usually a HLT instruction or a CALF to a fault handling routine.

3.8.1. Fault table entry

<table>
<thead>
<tr>
<th></th>
<th>CALF instruction</th>
<th></th>
<th>pointer to ECB</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Unused</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 3.9. Floating Point formats

#### 3.9.1. Memory formats

**Single precision:**

<table>
<thead>
<tr>
<th>1</th>
<th>24 25 32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mantissa</td>
<td>Exp</td>
</tr>
</tbody>
</table>

**Double precision:**

<table>
<thead>
<tr>
<th>1</th>
<th>48 49 64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mantissa</td>
<td>Exponent</td>
</tr>
</tbody>
</table>

**Quad precision:**

<table>
<thead>
<tr>
<th>1</th>
<th>48 49 64 65</th>
<th>112 113 128</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mantissa</td>
<td>Exponent</td>
<td>Mantissa</td>
</tr>
</tbody>
</table>

#### 3.9.2. Register formats

**Single precision (2250, 550-II, 650, older machines):**

<table>
<thead>
<tr>
<th>1</th>
<th>32 33 48</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mantissa</td>
<td>Exponent</td>
</tr>
</tbody>
</table>

**Single precision (750, 850, 9950):**

<table>
<thead>
<tr>
<th>1</th>
<th>48 49 64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mantissa</td>
<td>Exponent</td>
</tr>
</tbody>
</table>

**Double precision:**

<table>
<thead>
<tr>
<th>1</th>
<th>48 49 64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mantissa</td>
<td>Exponent</td>
</tr>
</tbody>
</table>

**Quad precision:**

<table>
<thead>
<tr>
<th>1</th>
<th>48 49 64 65</th>
<th>112 113 128</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mantissa</td>
<td>Exponent</td>
<td>Mantissa</td>
</tr>
</tbody>
</table>
3.10. Indirect Pointers (IP)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Fault bit: 1 - Missing Pointer</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>RR</td>
<td>Ring Number (00-11)</td>
<td>060000</td>
<td>6000</td>
</tr>
<tr>
<td>E</td>
<td>Extension bit: 1 - Word 3 is present with bit</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td></td>
<td>offset</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Segment</td>
<td>Segment Number</td>
<td>007777</td>
<td>0FFF</td>
</tr>
<tr>
<td>Word</td>
<td>Halfword offset</td>
<td>177777</td>
<td>FFFF</td>
</tr>
<tr>
<td>Bit</td>
<td>Bit offset within half word; present only if E is</td>
<td>170000</td>
<td>F000</td>
</tr>
<tr>
<td></td>
<td>set.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As an effective address in a base reg, F and E are ignored.
3.11. KEYS, MODALS
(CRS 24 RFILe 124,164 CRASH 50,150)

KEYSH (Keys):
S, R modes:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C Bit(CBIT)</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>2</td>
<td>Double Precision(DBL)</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>3</td>
<td>Reserved</td>
<td>020000</td>
<td>2000</td>
</tr>
</tbody>
</table>
| 4-6 | Addressing Mode:
     | x00xxx 16S (0)                   |       |      |
     | x02xxx 32S (1)                   |       |      |
     | x04xxx 64R (2)                   |       |      |
     | x06xxx 32R (3)                   |       |      |
     | x10xxx 32I (4)                   |       |      |
     | x14xxx 64V (6)                   |       |      |
| 7   | Floating exception (FEX)
     | 0: set CBIT & fault              | 001000| 0200 |
     | 1: set CBIT                      |       |      |
| 8   | Integer exception(IEX)
     | 0: set CBIT                      | 000400| 0100 |
     | 1: set CBIT & fault              |       |      |
| 9-16| Visible shift count.             | 000377| 00FF |

V, I modes:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C bit(CBIT)</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>2</td>
<td>Reserved, must be zero.</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>3</td>
<td>Link bit (L, LINK)</td>
<td>020000</td>
<td>2000</td>
</tr>
</tbody>
</table>
| 4-6 | Addressing mode:
     | x00xxx 16S (0)                   |       |      |
     | x02xxx 32S (1)                   |       |      |
     | x04xxx 64R (2)                   |       |      |
     | x06xxx 32R (3)                   |       |      |
     | x10xxx 32I (4)                   |       |      |
     | x14xxx 64V (6)                   |       |      |
| 7   | Floating exception (FEX)
     | 0: set CBIT & fault              | 001000| 0200 |
     | 1: set CBIT                      |       |      |
| 8   | Integer exception(IEX)
<pre><code> | 0: set CBIT                      | 000400| 0100 |
 | 1: set CBIT &amp; fault              |       |      |
</code></pre>
<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>Less than condition code (LT, CCLT)</td>
<td>000200</td>
<td>0080</td>
</tr>
<tr>
<td>10</td>
<td>Equal condition code (EQ, CCEQ)</td>
<td>000100</td>
<td>0040</td>
</tr>
<tr>
<td>11</td>
<td>Decimal exception (DEX)</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>12</td>
<td>ASCII-8</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>13</td>
<td>Floating Round if set</td>
<td>000010</td>
<td>0008</td>
</tr>
<tr>
<td>14</td>
<td>P850</td>
<td>000004</td>
<td>0004</td>
</tr>
<tr>
<td>15</td>
<td>ID (In Dispatcher)</td>
<td>000002</td>
<td>0002</td>
</tr>
<tr>
<td>16</td>
<td>SD (Save Done)</td>
<td>000001</td>
<td>0001</td>
</tr>
</tbody>
</table>

**KEYSL (Modals):**

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ENB (1: intpts enabled; 0: intpts disabled)</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>2</td>
<td>VIM (1: vectored int. mode; 0: std int. mode)</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>3-8</td>
<td>Unused, must be zero.</td>
<td>037400</td>
<td>3F00</td>
</tr>
<tr>
<td>9-11</td>
<td>CRS:</td>
<td>000340</td>
<td>01E0</td>
</tr>
<tr>
<td></td>
<td>xxx00x  Reg File 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>xxx04x  Reg File 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>MIO (1: mapped I/O; 0: unmapped I/O)</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>13</td>
<td>PXM (1: process exch enabled)</td>
<td>000010</td>
<td>0008</td>
</tr>
<tr>
<td>14</td>
<td>SEG (1: segmentation enabled)</td>
<td>000004</td>
<td>0004</td>
</tr>
<tr>
<td>15-16</td>
<td>MCM (Machine Check Mode):</td>
<td>000003</td>
<td>0003</td>
</tr>
<tr>
<td></td>
<td>0 No reporting</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 Memory Parity, uncorrected (ECCU)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 Quiet; all unrecovered errors</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 Record; report all errors</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.12. Modals

See KEYS (KEYSL).
3.13. Page maps

3.13.1. HMAP, LMAP

Locations of pagemaps:

<table>
<thead>
<tr>
<th>Rev</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 19.2</td>
<td>4/4000</td>
</tr>
<tr>
<td>19.2</td>
<td>segments 401-420</td>
</tr>
<tr>
<td>&gt; 19.2</td>
<td>segments 601-620</td>
</tr>
</tbody>
</table>

HMAP, LMAP interleaved in 64-word chunks, thus 128 words/segment in system.

HMAP (Hardware Map):

9950, 9955:

```
1 2 3 4 5 16 17 19 20 32
R U M S  Software     000     PPN
```

Other:

```
1 2 3 4 5          16
R U M S          PPN
```

<table>
<thead>
<tr>
<th>Mnem</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>If set, page is resident</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>U</td>
<td>If set, page has been used</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>M</td>
<td>If reset, page has been modified</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>S</td>
<td>If set, page is shared (inhibits cache)</td>
<td>010000</td>
<td>1000</td>
</tr>
</tbody>
</table>

Software Reserved for software:

```
Reserved for software: 007777 0FFF
```

PPN Physical Page Number

```
9950: 017777 1FFF
```

High order 13/12 bits of physical page address:

```
Other: 007777 0FFF
```

If non-resident, bits 3,5 software defined:

3,5 024000 Page status:

- 000000 Not resident, copy on disk
- 020000 Not resident, no copy on disk
- 004000 In transition, coming in
- 024000 In transition, going out
LMAP (Software Map – HMAP+’100):

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2</td>
<td>Lock counter (0 - unlocked)</td>
<td>140000</td>
<td>C000</td>
</tr>
<tr>
<td>3</td>
<td>First Time (just paged in)</td>
<td>020000</td>
<td>0200</td>
</tr>
<tr>
<td>4</td>
<td>Use alternate paging device</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>5-16</td>
<td>Record index (1 val/8 pages)</td>
<td>007777</td>
<td>0FF</td>
</tr>
</tbody>
</table>

3.14. MMAP entry

<table>
<thead>
<tr>
<th>Page Status</th>
<th>Unused</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>17</td>
</tr>
</tbody>
</table>

PAGE STATUS < 0 page is not to be used.
PAGE STATUS = 0 page is available.
PAGE STATUS > 0 page is in use; PAGE STATUS points to an HMAP entry.

3.15. Process Control Block (PCB)

<table>
<thead>
<tr>
<th>Word</th>
<th>Description</th>
<th>Word</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Level (in ready list)</td>
<td>40</td>
<td>GR7</td>
</tr>
<tr>
<td>1</td>
<td>Link (next PCB in list)</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Wait list SN (0 = ready)</td>
<td>42</td>
<td>FP0</td>
</tr>
<tr>
<td>3</td>
<td>Wait list word number</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Abort flags</td>
<td>44</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Last Reg set used</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Reserved</td>
<td>46</td>
<td>FP1</td>
</tr>
<tr>
<td>7</td>
<td>Reserved</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Elapsed timer (Low)</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Elapsed timer (High)</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>DTAR2H</td>
<td>52</td>
<td>PBH</td>
</tr>
<tr>
<td>13</td>
<td>DTAR2L</td>
<td>53</td>
<td>PBL</td>
</tr>
<tr>
<td>14</td>
<td>DTAR3H</td>
<td>54</td>
<td>SBH</td>
</tr>
<tr>
<td>15</td>
<td>DTAR3L</td>
<td>55</td>
<td>SBL</td>
</tr>
<tr>
<td>16</td>
<td>Interval timer</td>
<td>56</td>
<td>LBH</td>
</tr>
<tr>
<td>17</td>
<td>Reserved</td>
<td>57</td>
<td>LBL</td>
</tr>
<tr>
<td>20</td>
<td>SAVE MASK</td>
<td>60</td>
<td>XBH</td>
</tr>
<tr>
<td>21</td>
<td>KEYS</td>
<td>61</td>
<td>XBL</td>
</tr>
<tr>
<td>22</td>
<td>GR0</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
<td>63</td>
<td>Fault vector, RING 0</td>
</tr>
<tr>
<td>24</td>
<td>GR1</td>
<td>64</td>
<td>Fault vector, RING 1</td>
</tr>
<tr>
<td>25</td>
<td></td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>GR2</td>
<td>66</td>
<td>Reserved</td>
</tr>
<tr>
<td>27</td>
<td></td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>GR3</td>
<td>70</td>
<td>Fault vector, RING 3</td>
</tr>
<tr>
<td>31</td>
<td></td>
<td>71</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>GR4</td>
<td>72</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td>73</td>
<td>Page fault vector</td>
</tr>
<tr>
<td>34</td>
<td>GR5</td>
<td>74</td>
<td>Concealed stack, First</td>
</tr>
<tr>
<td>35</td>
<td></td>
<td>75</td>
<td>Concealed stack, Next</td>
</tr>
<tr>
<td>36</td>
<td>GR6</td>
<td>76</td>
<td>Concealed stack, Last</td>
</tr>
<tr>
<td>37</td>
<td></td>
<td>77</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

Word 5 has the following format for 850s:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
</table>

3-18 Prime Restricted
### 3.16. Queue Control Block (QCB)

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>Restrict process from ISU:</td>
<td>170000</td>
<td>F000</td>
</tr>
<tr>
<td></td>
<td>0000 - no restrictions</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0100 - bar from this ISU</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1000 - bar from other ISU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Reserved</td>
<td>004000</td>
<td>0800</td>
</tr>
<tr>
<td>6-7</td>
<td>Last ISU</td>
<td>003000</td>
<td>0600</td>
</tr>
<tr>
<td></td>
<td>01 - this ISU</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10 - other ISU</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Register have been saved in PCB</td>
<td>000400</td>
<td>0100</td>
</tr>
<tr>
<td>9-11</td>
<td>Last register set used. (same as modal CRS)</td>
<td>000340</td>
<td>00E0</td>
</tr>
<tr>
<td>12</td>
<td>Reserved</td>
<td>000020</td>
<td>0100</td>
</tr>
<tr>
<td>13-16</td>
<td>Process is lock to ISU:</td>
<td>000017</td>
<td>00F</td>
</tr>
<tr>
<td></td>
<td>0000 - neither</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0100 - this ISU</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1000 - other ISU</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 3.17. READY LIST

- **PPA**: current level/current PCB
- **PPB**: next level/next PCB
- **LEVELn**: first PCB on LEVELn
- **LEVELn+1**: last PCB on LEVELn
- **PCB+0**: level this PCB is on
- **PCB+1**: next pcb, 0 if last

Ready list in Segment 4, starting at 4/600:

<table>
<thead>
<tr>
<th>Lev</th>
<th>PCB on level</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>CLOCK PROCESS</td>
</tr>
<tr>
<td>602</td>
<td>SMLC PROCESS</td>
</tr>
<tr>
<td>604</td>
<td>AMLC PROCESS</td>
</tr>
<tr>
<td>606</td>
<td>MPC, MP2 PROCESSES</td>
</tr>
<tr>
<td>610</td>
<td>VERSATEC PROCESS</td>
</tr>
<tr>
<td>612</td>
<td>IPC PROCESS</td>
</tr>
</tbody>
</table>
### 3.18. Registers

Register file allocations.

**2550, 9650:**

<table>
<thead>
<tr>
<th>Register File</th>
<th>Absolute Loc</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF0</td>
<td>'0'-37</td>
<td>Microcode (set 1 for 9650)</td>
</tr>
<tr>
<td>RF1</td>
<td>'40'-77</td>
<td>DMA channels (32)</td>
</tr>
<tr>
<td>RF2</td>
<td>'100'-137</td>
<td>User register set 2</td>
</tr>
<tr>
<td>RF3</td>
<td>'140'-177</td>
<td>User register set 3</td>
</tr>
<tr>
<td>RF4</td>
<td>'200'-237</td>
<td>User register set 4</td>
</tr>
<tr>
<td>RF5</td>
<td>'240'-277</td>
<td>User register set 5</td>
</tr>
<tr>
<td>RF6</td>
<td>'300'-337</td>
<td>User register set 6</td>
</tr>
<tr>
<td>RF7</td>
<td>'340'-377</td>
<td>User register set 7</td>
</tr>
<tr>
<td>RF8</td>
<td>'400'-437</td>
<td>User register set 8</td>
</tr>
<tr>
<td>RF9</td>
<td>'440'-477</td>
<td>User register set 9</td>
</tr>
<tr>
<td>RF10</td>
<td></td>
<td>Microcode set 2 for 9650</td>
</tr>
</tbody>
</table>

All others:

<table>
<thead>
<tr>
<th>Register File</th>
<th>Absolute Loc</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF0</td>
<td>'0'-37</td>
<td>Microcode (set 1 for 9750, 9950, 9955)</td>
</tr>
<tr>
<td>RF1</td>
<td>'40'-77</td>
<td>DMA channels (32)</td>
</tr>
<tr>
<td>RF2</td>
<td>'100'-137</td>
<td>User register set 2</td>
</tr>
<tr>
<td>RF3</td>
<td>'140'-177</td>
<td>User register set 3</td>
</tr>
<tr>
<td>RF4</td>
<td>'200'-237</td>
<td>User register set 4 (9750, 9950, 9955)</td>
</tr>
<tr>
<td>RF5</td>
<td>'240'-277</td>
<td>User register set 5 (9750, 9950, 9955)</td>
</tr>
<tr>
<td>RF6</td>
<td>'300'-337</td>
<td>User register set 6 (9750, 9950, 9955)</td>
</tr>
<tr>
<td>RF7</td>
<td>'340'-377</td>
<td>Microcode set 2 for 9750, 9950, 9955</td>
</tr>
</tbody>
</table>
## Microcode registers:

### 9750, 9950, 9955:

<table>
<thead>
<tr>
<th>Reg num</th>
<th>Contents</th>
<th>Crash addr</th>
<th>Reg num</th>
<th>Contents</th>
<th>Crash addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>TR0</td>
<td>300</td>
<td>300</td>
<td>DGR0 (STLBRF1)</td>
<td>200</td>
</tr>
<tr>
<td>1</td>
<td>TR1</td>
<td>302</td>
<td>301</td>
<td>DGR1 (STLBRF2)</td>
<td>202</td>
</tr>
<tr>
<td>2</td>
<td>TR2</td>
<td>304</td>
<td>302</td>
<td>DGR2 (RDMP)</td>
<td>204</td>
</tr>
<tr>
<td>3</td>
<td>TR3</td>
<td>306</td>
<td>303</td>
<td>DGR3</td>
<td>206</td>
</tr>
<tr>
<td>4</td>
<td>TR4</td>
<td>310</td>
<td>304</td>
<td>DGR4</td>
<td>210</td>
</tr>
<tr>
<td>5</td>
<td>TR5</td>
<td>312</td>
<td>305</td>
<td>DGR5</td>
<td>212</td>
</tr>
<tr>
<td>6</td>
<td>TR6</td>
<td>314</td>
<td>306</td>
<td>DGR6, RSSAV(9955)</td>
<td>214</td>
</tr>
<tr>
<td>7</td>
<td>TR7</td>
<td>316</td>
<td>307</td>
<td>DGR7</td>
<td>216</td>
</tr>
<tr>
<td>8</td>
<td>TR8, FR032</td>
<td>320</td>
<td>310</td>
<td>DGR10</td>
<td>220</td>
</tr>
<tr>
<td>9</td>
<td>TR9</td>
<td>322</td>
<td>311</td>
<td>DGR11</td>
<td>222</td>
</tr>
<tr>
<td>10</td>
<td>TR10, FR132</td>
<td>324</td>
<td>312</td>
<td>DGR12</td>
<td>224</td>
</tr>
<tr>
<td>11</td>
<td>TR11</td>
<td>326</td>
<td>313</td>
<td>DGR13, FF80(9955)</td>
<td>226</td>
</tr>
<tr>
<td>12</td>
<td>REOIV, UCSADDR</td>
<td>330</td>
<td>314</td>
<td>DGR14</td>
<td>230</td>
</tr>
<tr>
<td>13</td>
<td>RDSAVE</td>
<td>332</td>
<td>315</td>
<td>DGR15</td>
<td>232</td>
</tr>
<tr>
<td>14</td>
<td>CFF00, C00FF</td>
<td>334</td>
<td>316</td>
<td>DGR16</td>
<td>234</td>
</tr>
<tr>
<td>15</td>
<td>RATMP</td>
<td>336</td>
<td>317</td>
<td>DGR17</td>
<td>236</td>
</tr>
<tr>
<td>16</td>
<td>RMSAVE</td>
<td>340</td>
<td>320</td>
<td>MINUS1</td>
<td>240</td>
</tr>
<tr>
<td>17</td>
<td>342</td>
<td>321</td>
<td>ONE32</td>
<td>242</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>PARREG1</td>
<td>344</td>
<td>322</td>
<td>KMASK, IUART</td>
<td>244</td>
</tr>
<tr>
<td>19</td>
<td>PARREG2</td>
<td>346</td>
<td>323</td>
<td>C3F, C3F</td>
<td>246</td>
</tr>
<tr>
<td>20</td>
<td>PARREG3</td>
<td>350</td>
<td>324</td>
<td>C8000</td>
<td>250</td>
</tr>
<tr>
<td>21</td>
<td>PBSAVE</td>
<td>352</td>
<td>325</td>
<td>C0000, C0080</td>
<td>252</td>
</tr>
<tr>
<td>22</td>
<td>SYSREG1</td>
<td>354</td>
<td>326</td>
<td>C9000, C0080</td>
<td>254</td>
</tr>
<tr>
<td>23</td>
<td>DSWPARITY</td>
<td>356</td>
<td>327</td>
<td>CB100, CB200</td>
<td>256</td>
</tr>
<tr>
<td>24</td>
<td>PSWPB</td>
<td>360</td>
<td>330</td>
<td>CB6666</td>
<td>260</td>
</tr>
<tr>
<td>25</td>
<td>PWSKEYS</td>
<td>362</td>
<td>331</td>
<td>CB1K, ACK2</td>
<td>262</td>
</tr>
<tr>
<td>26</td>
<td>PLA, PPA</td>
<td>364</td>
<td>332</td>
<td>FERRET6</td>
<td>264</td>
</tr>
<tr>
<td>27</td>
<td>PLB, PPB</td>
<td>366</td>
<td>333</td>
<td>FERRET5</td>
<td>266</td>
</tr>
<tr>
<td>28</td>
<td>DSWMA</td>
<td>370</td>
<td>334</td>
<td>FERRET4</td>
<td>270</td>
</tr>
<tr>
<td>29</td>
<td>DSWSTAT</td>
<td>372</td>
<td>335</td>
<td>FERRET3</td>
<td>272</td>
</tr>
<tr>
<td>30</td>
<td>DSWPB</td>
<td>374</td>
<td>336</td>
<td>FERRET2</td>
<td>274</td>
</tr>
<tr>
<td>31</td>
<td>RSVPTR</td>
<td>376</td>
<td>337</td>
<td>FERRET1</td>
<td>276</td>
</tr>
<tr>
<td>Reg num</td>
<td>Contents</td>
<td>Crash addr</td>
<td>Reg num</td>
<td>Contents</td>
<td>Crash addr</td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td>------------</td>
<td>---------</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>0</td>
<td>TR0</td>
<td>1000</td>
<td>500</td>
<td>DEC0</td>
<td>1100</td>
</tr>
<tr>
<td>1</td>
<td>TR1</td>
<td>1002</td>
<td>501</td>
<td>DEC1</td>
<td>1102</td>
</tr>
<tr>
<td>2</td>
<td>TR2</td>
<td>1004</td>
<td>502</td>
<td>DEC2</td>
<td>1104</td>
</tr>
<tr>
<td>3</td>
<td>TR3</td>
<td>1006</td>
<td>503</td>
<td>DEC3</td>
<td>1106</td>
</tr>
<tr>
<td>4</td>
<td>TR4</td>
<td>1010</td>
<td>504</td>
<td>DEC4</td>
<td>1110</td>
</tr>
<tr>
<td>5</td>
<td>TR5</td>
<td>1012</td>
<td>505</td>
<td>DEC5</td>
<td>1112</td>
</tr>
<tr>
<td>6</td>
<td>TR6</td>
<td>1014</td>
<td>506</td>
<td>DEC6</td>
<td>1114</td>
</tr>
<tr>
<td>7</td>
<td>TR7</td>
<td>1016</td>
<td>507</td>
<td>DEC7</td>
<td>1116</td>
</tr>
<tr>
<td>10</td>
<td>RDMX1</td>
<td>1020</td>
<td>510</td>
<td>DEC10</td>
<td>1120</td>
</tr>
<tr>
<td>11</td>
<td>RDMX2</td>
<td>1022</td>
<td>511</td>
<td>RECC3</td>
<td>1122</td>
</tr>
<tr>
<td>12</td>
<td>USCAADDR, REOIV</td>
<td>1024</td>
<td>512</td>
<td>TMRSAVE</td>
<td>1124</td>
</tr>
<tr>
<td>13</td>
<td>RSGT1</td>
<td>1026</td>
<td>513</td>
<td>CTRLBYTE, QFDIDX</td>
<td>1126</td>
</tr>
<tr>
<td>14</td>
<td>RSGT2</td>
<td>1030</td>
<td>514</td>
<td>CMDBYTE, SCR14L</td>
<td>1130</td>
</tr>
<tr>
<td>15</td>
<td>RECC1</td>
<td>1032</td>
<td>515</td>
<td>EXP32</td>
<td>1132</td>
</tr>
<tr>
<td>16</td>
<td>RECC2</td>
<td>1034</td>
<td>516</td>
<td>SSN</td>
<td>1134</td>
</tr>
<tr>
<td>17</td>
<td>TEMPCACH</td>
<td>1036</td>
<td>517</td>
<td>SWITCHES, PICSTAT</td>
<td>1136</td>
</tr>
<tr>
<td>20</td>
<td>ONE32</td>
<td>1040</td>
<td>520</td>
<td>WWADTR</td>
<td>1140</td>
</tr>
<tr>
<td>21</td>
<td>PBSAVE</td>
<td>1042</td>
<td>521</td>
<td>ADDRREG2</td>
<td>1142</td>
</tr>
<tr>
<td>22</td>
<td>RDMX3</td>
<td>1044</td>
<td>522</td>
<td>ADDRREG</td>
<td>1144</td>
</tr>
<tr>
<td>23</td>
<td>RDMX4</td>
<td>1046</td>
<td>523</td>
<td>LIGHTS, INTVEC</td>
<td>1146</td>
</tr>
<tr>
<td>24</td>
<td>C377</td>
<td>1050</td>
<td>524</td>
<td>QPTR, BYTFLG</td>
<td>1150</td>
</tr>
<tr>
<td>25</td>
<td>MINUS1</td>
<td>1052</td>
<td>525</td>
<td>WSLFLG</td>
<td>1152</td>
</tr>
<tr>
<td>26</td>
<td>LREGSET, CHKREG</td>
<td>1054</td>
<td>526</td>
<td>RDMX5</td>
<td>1154</td>
</tr>
<tr>
<td>27</td>
<td>DSWPARITY</td>
<td>1056</td>
<td>527</td>
<td>UMASK1, SCR27L</td>
<td>1156</td>
</tr>
<tr>
<td>30</td>
<td>PSWPB</td>
<td>1060</td>
<td>530</td>
<td>UMASK2, SCR30L</td>
<td>1160</td>
</tr>
<tr>
<td>31</td>
<td>PSWKEYS</td>
<td>1062</td>
<td>531</td>
<td>URDRXH, SCR31L</td>
<td>1162</td>
</tr>
<tr>
<td>32</td>
<td>PPA</td>
<td>1064</td>
<td>532</td>
<td>BFR04</td>
<td>1164</td>
</tr>
<tr>
<td>33</td>
<td>PPB</td>
<td>1066</td>
<td>533</td>
<td>DSSW</td>
<td>1166</td>
</tr>
<tr>
<td>34</td>
<td>DSWRMA</td>
<td>1070</td>
<td>534</td>
<td>RSTLB1</td>
<td>1170</td>
</tr>
<tr>
<td>35</td>
<td>DSWSSTAT</td>
<td>1072</td>
<td>535</td>
<td>RSTLB2</td>
<td>1172</td>
</tr>
<tr>
<td>36</td>
<td>DSWPB</td>
<td>1074</td>
<td>536</td>
<td>RSTLB3</td>
<td>1174</td>
</tr>
<tr>
<td>37</td>
<td>RSAVPTR</td>
<td>1076</td>
<td>537</td>
<td>RSTLB4</td>
<td>1176</td>
</tr>
</tbody>
</table>
**Other 50 series:**

<table>
<thead>
<tr>
<th>Rfile addr</th>
<th>Contents</th>
<th>Crash addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>TR0</td>
<td>300</td>
</tr>
<tr>
<td>1</td>
<td>TR1</td>
<td>302</td>
</tr>
<tr>
<td>2</td>
<td>TR2</td>
<td>304</td>
</tr>
<tr>
<td>3</td>
<td>TR3</td>
<td>306</td>
</tr>
<tr>
<td>4</td>
<td>TR4</td>
<td>310</td>
</tr>
<tr>
<td>5</td>
<td>TR5</td>
<td>312</td>
</tr>
<tr>
<td>6</td>
<td>TR6</td>
<td>314</td>
</tr>
<tr>
<td>7</td>
<td>TR7(PB)</td>
<td>316</td>
</tr>
<tr>
<td>10</td>
<td>RDMX1</td>
<td>320</td>
</tr>
<tr>
<td>11</td>
<td>RDMX2</td>
<td>322</td>
</tr>
<tr>
<td>12</td>
<td>USCADDR (750,850)/REOIV</td>
<td>324</td>
</tr>
<tr>
<td>13</td>
<td>RSGT1</td>
<td>326</td>
</tr>
<tr>
<td>14</td>
<td>RSGT2</td>
<td>330</td>
</tr>
<tr>
<td>15</td>
<td>RECC1</td>
<td>332</td>
</tr>
<tr>
<td>16</td>
<td>RECC2</td>
<td>334</td>
</tr>
<tr>
<td>17</td>
<td>-RATMPL</td>
<td>336</td>
</tr>
<tr>
<td>20</td>
<td>ZERO/ONE</td>
<td>340</td>
</tr>
<tr>
<td>21</td>
<td>PBSAVE</td>
<td>342</td>
</tr>
<tr>
<td>22</td>
<td>RDMX3</td>
<td>344</td>
</tr>
<tr>
<td>23</td>
<td>RDMX4</td>
<td>346</td>
</tr>
<tr>
<td>24</td>
<td>C377</td>
<td>350</td>
</tr>
<tr>
<td>25</td>
<td>MINUS1/MINUS2</td>
<td>352</td>
</tr>
<tr>
<td>26</td>
<td>WWADTR</td>
<td>354</td>
</tr>
<tr>
<td>27</td>
<td>DSWPARITY(&gt;750)</td>
<td>356</td>
</tr>
<tr>
<td>30</td>
<td>PSWPB</td>
<td>360</td>
</tr>
<tr>
<td>31</td>
<td>PSWKEYS</td>
<td>362</td>
</tr>
<tr>
<td>32</td>
<td>PPA/PCBA</td>
<td>364</td>
</tr>
<tr>
<td>33</td>
<td>PBB/PCCB</td>
<td>366</td>
</tr>
<tr>
<td>34</td>
<td>DSWRMA</td>
<td>370</td>
</tr>
<tr>
<td>35</td>
<td>DSWSTAT</td>
<td>372</td>
</tr>
<tr>
<td>36</td>
<td>DSWPB</td>
<td>374</td>
</tr>
<tr>
<td>37</td>
<td>RSAVPTR</td>
<td>376</td>
</tr>
</tbody>
</table>

RFIL ADDR = Address in Register File
CRASH ADDR = Disp in hardware register save area.

TR7          PB at machine halt
PSWPB        PB at last interrupt
PSWKEYS      Keys at last interrupt
PPA          Current level/current PCB
PPB          Next level/next PCB
RSAVPTR      Reg save area ptr. 0: regs saved.
'33          850 only: 41004 - this ISU; 102010 - other ISU

**Register set**

<table>
<thead>
<tr>
<th>Reg num</th>
<th>I-mode</th>
<th>V-mode</th>
<th>R-mode</th>
<th>Rel crash addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>GR0</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>GR1</td>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>GR2</td>
<td>L_A/B</td>
<td>A(1)^5/B(2)</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>GR3</td>
<td>E</td>
<td></td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>GR4</td>
<td>-Y</td>
<td>-S(3)</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>GR5</td>
<td></td>
<td></td>
<td>12</td>
</tr>
</tbody>
</table>

^o(n) indicate P900 address mapping
<table>
<thead>
<tr>
<th>Reg num</th>
<th>I-mode</th>
<th>V-mode</th>
<th>R-mode</th>
<th>Rel crash addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>GR6</td>
<td>-/X</td>
<td>-/X(3)</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>GR7</td>
<td></td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>10</td>
<td>FAR0,FAC0L</td>
<td>FAR0</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>11</td>
<td>FLR0</td>
<td></td>
<td>('13/-')</td>
<td>22</td>
</tr>
<tr>
<td>12</td>
<td>FAR1</td>
<td></td>
<td>(4/5)</td>
<td>24</td>
</tr>
<tr>
<td>13</td>
<td>FLR1</td>
<td></td>
<td>(6/-)</td>
<td>26</td>
</tr>
<tr>
<td>14</td>
<td>PB</td>
<td></td>
<td>PB</td>
<td>30</td>
</tr>
<tr>
<td>15</td>
<td>SB</td>
<td></td>
<td>SB('14/15)</td>
<td>32</td>
</tr>
<tr>
<td>16</td>
<td>LB</td>
<td></td>
<td>('16/17')</td>
<td>34</td>
</tr>
<tr>
<td>17</td>
<td>XB</td>
<td></td>
<td>XB</td>
<td>36</td>
</tr>
<tr>
<td>20</td>
<td>DTAR3</td>
<td></td>
<td>DTAR3('10/-)</td>
<td>40</td>
</tr>
<tr>
<td>21</td>
<td>DTAR2</td>
<td></td>
<td>DTAR2</td>
<td>42</td>
</tr>
<tr>
<td>22</td>
<td>DTAR1</td>
<td></td>
<td>DTAR1</td>
<td>44</td>
</tr>
<tr>
<td>23</td>
<td>DTAR0</td>
<td></td>
<td>DTAR0</td>
<td>46</td>
</tr>
<tr>
<td>24</td>
<td>KEYS/MODALS</td>
<td>KEYS/MODALS</td>
<td>KEYS/MODALS</td>
<td>50</td>
</tr>
<tr>
<td>25</td>
<td>OWNER</td>
<td></td>
<td>OWNER</td>
<td>52</td>
</tr>
<tr>
<td>26</td>
<td>FCODE</td>
<td></td>
<td>FCODE('11/-)</td>
<td>54</td>
</tr>
<tr>
<td>27</td>
<td>FADDR</td>
<td></td>
<td>FADDR(-'12)</td>
<td>56</td>
</tr>
<tr>
<td>30</td>
<td>TIMER</td>
<td></td>
<td>TIMER</td>
<td>60</td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td>62</td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
<td></td>
<td>64</td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
<td></td>
<td>66</td>
</tr>
<tr>
<td>34</td>
<td></td>
<td></td>
<td></td>
<td>70</td>
</tr>
<tr>
<td>35</td>
<td></td>
<td></td>
<td></td>
<td>72</td>
</tr>
<tr>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td>74</td>
</tr>
<tr>
<td>37</td>
<td></td>
<td></td>
<td></td>
<td>76</td>
</tr>
</tbody>
</table>

See the *System Architecture Reference Guide* [65] for additional information.

### 3.19. RSAV format
Registers as saved/restored by the RSAV/RRST instructions.
### Save Mask:

<table>
<thead>
<tr>
<th>Mmnr</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>Unused</td>
<td>170000</td>
<td>F000</td>
</tr>
<tr>
<td>5</td>
<td>FRN1</td>
<td>004000</td>
<td>0800</td>
</tr>
<tr>
<td>6</td>
<td>FR1</td>
<td>002000</td>
<td>0400</td>
</tr>
<tr>
<td>7</td>
<td>FRN0</td>
<td>001000</td>
<td>0200</td>
</tr>
<tr>
<td>8</td>
<td>FR0</td>
<td>000400</td>
<td>0100</td>
</tr>
<tr>
<td>9</td>
<td>GR7</td>
<td>000200</td>
<td>0060</td>
</tr>
<tr>
<td>10</td>
<td>GR6</td>
<td>000100</td>
<td>0040</td>
</tr>
<tr>
<td>11</td>
<td>GR5</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>12</td>
<td>GR4</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>13</td>
<td>GR3</td>
<td>000010</td>
<td>0008</td>
</tr>
<tr>
<td>14</td>
<td>GR2</td>
<td>000004</td>
<td>0004</td>
</tr>
<tr>
<td>15</td>
<td>GR1</td>
<td>000002</td>
<td>0002</td>
</tr>
<tr>
<td>16</td>
<td>GR0</td>
<td>000001</td>
<td>0001</td>
</tr>
</tbody>
</table>

The XB is always saved.
3.20. Segment descriptor word (SDW)

<table>
<thead>
<tr>
<th>Mnem</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAL</td>
<td>Bits 7-22 of physical addr of a PMT or HMAP entry; bits 17-22 must be zero</td>
<td>177700</td>
<td>FFC0</td>
</tr>
<tr>
<td>F</td>
<td>Fault, 1 = No segment or missing pagemap</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>Ac1</td>
<td>Access controls for Ring 1: 000 No access 001 Gate 010 Read 011 Read/Write 100 Reserved 101 Reserved 110 Read/XEQ 111 R/W/XEQ</td>
<td>070000</td>
<td>7000</td>
</tr>
<tr>
<td>Ac2</td>
<td>Access controls for Ring 2 (not used)</td>
<td>007000</td>
<td>0E00</td>
</tr>
<tr>
<td>Ac3</td>
<td>Access controls for Ring 3</td>
<td>000700</td>
<td>01C0</td>
</tr>
<tr>
<td>PAU</td>
<td>Bits 1-6 of physical addr of a PMT or HMAP entry.</td>
<td>000077</td>
<td>003F</td>
</tr>
</tbody>
</table>

3.21. Semaphores

<table>
<thead>
<tr>
<th>Word</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wait counter</td>
<td>Number of outstanding waits: 0 - empty list &lt;0 - uneventful notifies &gt;0 - PCBs waiting</td>
</tr>
<tr>
<td>Location</td>
<td>Location of first PCB in OWNERH segment</td>
</tr>
</tbody>
</table>
### 3.22. Stack frame

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0:PCL 1:CALF</td>
</tr>
<tr>
<td>1</td>
<td>SN of stack root</td>
</tr>
<tr>
<td>2</td>
<td>PB for return</td>
</tr>
<tr>
<td>3</td>
<td>Caller's SB</td>
</tr>
<tr>
<td>4</td>
<td>Caller's LB</td>
</tr>
<tr>
<td>5</td>
<td>Caller's KEYS</td>
</tr>
<tr>
<td>6</td>
<td>PBCL</td>
</tr>
<tr>
<td>7</td>
<td>FCODE if CALF</td>
</tr>
<tr>
<td>8</td>
<td>FADDR if CALF</td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

### 3.23. Stack Headers

**Stack root header (word 0 of stack segment)**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Free pointer</td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>First extension</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**Stack extension header (word 0 of stack segment)**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Next extension</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

---

6Start of automatic storage if PCL. See section 4.10

---

Prime Restricted 3-27
3.24. STLB

9750, 9950, 9955:

\[
\begin{array}{cccccc}
VMS & Ac1 & Ac3 & ProclID & Seg & Phys Addr \\
\hline
1 & 2 & 3 & 4 & 6 & 7 & 9 & 10 & 21 & 22 & 33 & 34 & 46 \\
\end{array}
\]

2550, 9650:

\[
\begin{array}{cccccc}
VMS & Ac1 & Ac3 & ProclID & Seg & Phys Addr \\
\hline
1 & 2 & 3 & 4 & 6 & 7 & 9 & 10 & 19 & 20 & 28 & 29 & 40 \\
\end{array}
\]

All others:

\[
\begin{array}{cccccc}
VMS & Ac1 & Ac3 & ProclID & Seg & Phys Addr \\
\hline
1 & 2 & 3 & 4 & 6 & 7 & 9 & 10 & 21 & 22 & 33 & 34 & 45 \\
\end{array}
\]

<table>
<thead>
<tr>
<th>Mnem</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>STLB has valid data.</td>
</tr>
<tr>
<td>M</td>
<td>Page has been modified.</td>
</tr>
<tr>
<td>S</td>
<td>This memory is shared.</td>
</tr>
<tr>
<td>Ac1</td>
<td>Ring 1 access rights.</td>
</tr>
<tr>
<td>Ac3</td>
<td>Ring 3 access rights.</td>
</tr>
<tr>
<td>ProclID</td>
<td>ID of the process referencing this memory.</td>
</tr>
<tr>
<td>Seg</td>
<td>Segment number of the virtual address.</td>
</tr>
<tr>
<td>Phys Addr</td>
<td>The physical page address.</td>
</tr>
</tbody>
</table>
4. PRIMOS

4.1. ABORT FLAGS
PCB+4, ABSAVE at 6000/20

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINALM</td>
<td>One minute abort flag</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>SMLALM</td>
<td>SLMC alarm</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>LGIALM</td>
<td>Login alarm</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>WRMALM</td>
<td>Warm start alarm</td>
<td>004000</td>
<td>0800</td>
</tr>
<tr>
<td>MSGALM</td>
<td>User 1 message alarm</td>
<td>002000</td>
<td>0400</td>
</tr>
<tr>
<td>CHKALM</td>
<td>Check alarm</td>
<td>001000</td>
<td>0200</td>
</tr>
<tr>
<td>SWIALM</td>
<td>Software interrupt alarm</td>
<td>002200</td>
<td>0800</td>
</tr>
<tr>
<td>IOALM</td>
<td>I/O completed alarm</td>
<td>001000</td>
<td>0400</td>
</tr>
<tr>
<td>IOMALM</td>
<td>I0$MSG alarm</td>
<td>000080</td>
<td>0020</td>
</tr>
<tr>
<td>DISALM</td>
<td>Disconnect alarm</td>
<td>000010</td>
<td>0008</td>
</tr>
<tr>
<td>TMOALM</td>
<td>Timeout alarm</td>
<td>000004</td>
<td>0004</td>
</tr>
<tr>
<td>AIOALM</td>
<td>Async. I/O alarm</td>
<td>000002</td>
<td>0002</td>
</tr>
<tr>
<td>TSEALM</td>
<td>Time slice end (firmware)</td>
<td>000001</td>
<td>0001</td>
</tr>
</tbody>
</table>

4.2. EPF formats

VCIB:

0
1
2 Type Version
3 Resume segs
4 Linkage areas
5 Debugger segs
6 CIB
7 ERP

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Program.</td>
</tr>
<tr>
<td>3</td>
<td>Process class library (initialized once per process).</td>
</tr>
<tr>
<td>5</td>
<td>Program class library (initialized once per program invocation).</td>
</tr>
<tr>
<td>6</td>
<td>Registrable program (may be registered).</td>
</tr>
<tr>
<td>8</td>
<td>Registrable process class library.</td>
</tr>
<tr>
<td>10</td>
<td>Registrable program class library.</td>
</tr>
<tr>
<td>CIB:</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>0</td>
<td>Version</td>
</tr>
<tr>
<td>1</td>
<td>Starting</td>
</tr>
<tr>
<td>2</td>
<td>ECB ERP</td>
</tr>
<tr>
<td>3</td>
<td>LTD list</td>
</tr>
<tr>
<td>4</td>
<td>ERP</td>
</tr>
<tr>
<td>5</td>
<td>LIB</td>
</tr>
<tr>
<td>6</td>
<td>ERP</td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Map</td>
</tr>
<tr>
<td>11</td>
<td>ERP</td>
</tr>
<tr>
<td>12</td>
<td>DBG info</td>
</tr>
<tr>
<td>13</td>
<td>ERP</td>
</tr>
<tr>
<td>14</td>
<td>Merge segs</td>
</tr>
<tr>
<td>15</td>
<td>Merge info</td>
</tr>
<tr>
<td>16</td>
<td>ERP</td>
</tr>
<tr>
<td>17</td>
<td>CP Flags</td>
</tr>
<tr>
<td>20</td>
<td>Name gen pos</td>
</tr>
<tr>
<td>21</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>AIB</td>
</tr>
<tr>
<td>24</td>
<td>ERP</td>
</tr>
<tr>
<td>25</td>
<td>Module body</td>
</tr>
<tr>
<td>26</td>
<td>ERP</td>
</tr>
</tbody>
</table>

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

<table>
<thead>
<tr>
<th>W</th>
<th>T</th>
<th>I</th>
<th>V</th>
<th>D</th>
<th>S</th>
<th>F</th>
<th>A</th>
<th>R</th>
<th></th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>Wildcarding enabled.</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>T</td>
<td>Treewalking enabled.</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>I</td>
<td>Iteration enabled.</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>V</td>
<td>Verification of wildcaring enabled.</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>D</td>
<td>Directories.</td>
<td>000200</td>
<td>0080</td>
</tr>
<tr>
<td>S</td>
<td>Segment directories.</td>
<td>000100</td>
<td>0040</td>
</tr>
<tr>
<td>F</td>
<td>Files.</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>A</td>
<td>Access categories (ACATs).</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>R</td>
<td>Recovery based file types only.</td>
<td>000010</td>
<td>0008</td>
</tr>
</tbody>
</table>

4-2 Prime Restricted
### 4.3. FIGCOM
Starts at 14/700

<table>
<thead>
<tr>
<th>Name</th>
<th>Dft</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOUTQM</td>
<td>1000</td>
<td>Inactive minutes to auto logout</td>
</tr>
<tr>
<td>LOTLIM</td>
<td>3</td>
<td>Inactive logout time during login</td>
</tr>
<tr>
<td>DONSTP</td>
<td>0</td>
<td>Phantom restart flag for warmstart</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 =&gt; Logout phantoms on warm start</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 =&gt; Continue phantoms on warm start</td>
</tr>
<tr>
<td>CSSEGS</td>
<td></td>
<td>Number of concealed stack segs.</td>
</tr>
<tr>
<td>DEFFERA</td>
<td>242</td>
<td>Default erase character (*)</td>
</tr>
<tr>
<td>DEFKIL</td>
<td>277</td>
<td>Default kill character (?)</td>
</tr>
<tr>
<td>PR1500</td>
<td></td>
<td>1 =&gt; P500</td>
</tr>
<tr>
<td>VERSIO</td>
<td></td>
<td>PRIMOS revision id (char(16)var)</td>
</tr>
<tr>
<td>NLGPLRT</td>
<td>1</td>
<td>1 =&gt; Inhibit login messages at console</td>
</tr>
<tr>
<td>LOGOVF</td>
<td>0</td>
<td>1 =&gt; Can’t login while logged in (login-over-login)</td>
</tr>
<tr>
<td>LRQUTR</td>
<td>10000</td>
<td>LOGREC quota (obsolete at 21.0)</td>
</tr>
<tr>
<td>DMQMSK</td>
<td></td>
<td>=157777777777 to disable DMQ-AMLC logic</td>
</tr>
<tr>
<td>CPUID</td>
<td></td>
<td>CPU model number</td>
</tr>
<tr>
<td>INSTLB</td>
<td></td>
<td>1 =&gt; Can use LIOT and PTLB instructions</td>
</tr>
<tr>
<td>APCCNF</td>
<td></td>
<td>1 =&gt; P850 cpu in use</td>
</tr>
<tr>
<td>UPSSW</td>
<td>-1</td>
<td>UPS config</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-1 =&gt; No UPS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 =&gt; Halt on warm-start</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt;0 =&gt; n seconds delay after warmstart</td>
</tr>
<tr>
<td>CPUREV</td>
<td></td>
<td>CPU U-CODE REV</td>
</tr>
<tr>
<td>STAMP</td>
<td></td>
<td>((15)bin)</td>
</tr>
<tr>
<td>RWLOCK</td>
<td>1</td>
<td>System read/write lock:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - 1 reader or 1 writer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - n readers or 1 writer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 - n readers and 1 writer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5 - n readers and n writers</td>
</tr>
<tr>
<td>ABBRSW</td>
<td>1</td>
<td>Abbreviation enabled flag</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 =&gt; Abbrevs enabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 =&gt; Abbrevs disabled</td>
</tr>
<tr>
<td>SLVRUN</td>
<td></td>
<td>1 =&gt; P850 slave CPU running</td>
</tr>
<tr>
<td>DTRDRP</td>
<td>0</td>
<td>1 =&gt; Drop DTR at logout</td>
</tr>
<tr>
<td>ZCPU</td>
<td></td>
<td>1 =&gt; can use ZMOV and ZFIL</td>
</tr>
<tr>
<td>STHMCN</td>
<td></td>
<td>1 =&gt; can use STTM</td>
</tr>
<tr>
<td>MAPREV</td>
<td></td>
<td>Rev of page map format</td>
</tr>
<tr>
<td>RGSETS</td>
<td></td>
<td>Number of user reg sets</td>
</tr>
<tr>
<td>RGSET0</td>
<td></td>
<td>Number of ucode reg set</td>
</tr>
<tr>
<td>ECCTRL</td>
<td></td>
<td>Memory controller ECC</td>
</tr>
<tr>
<td>BCLK</td>
<td></td>
<td>Battery clock with cpu</td>
</tr>
<tr>
<td>SENSOR</td>
<td></td>
<td>Environmental sensors</td>
</tr>
<tr>
<td>MMRHLT</td>
<td>1</td>
<td>Halt on ECCU</td>
</tr>
<tr>
<td>DISPCH</td>
<td></td>
<td>'in dispatcher' bit always set.</td>
</tr>
<tr>
<td>LOGBAD</td>
<td>0</td>
<td>Monitor failed logins</td>
</tr>
<tr>
<td>DEFMEL</td>
<td></td>
<td>Default minimum extent length</td>
</tr>
<tr>
<td>RDBG_ON</td>
<td>0</td>
<td>Ring 0 debugger configuration flag</td>
</tr>
<tr>
<td>SUSPEND_SLAVE</td>
<td></td>
<td>Debugger variable</td>
</tr>
<tr>
<td>TPDUMP</td>
<td></td>
<td>Take tape dumps</td>
</tr>
</tbody>
</table>

**FIGCME:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Dft</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>HIOSEG</td>
<td></td>
<td>Highest I/O window segment</td>
</tr>
<tr>
<td>STBCLK</td>
<td></td>
<td>PRIMOS can set the battery clock</td>
</tr>
<tr>
<td>MIRROR</td>
<td>0</td>
<td>Mirroring enabled</td>
</tr>
</tbody>
</table>

**Prime Restricted**  

4-3
4.4. LOCKS, LCKCOM

Locks are semaphores used to control access to serially reusable resources. Located in LCKCOM (SEG 6), source file N1LOCK.

<table>
<thead>
<tr>
<th>Lock</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSLOK</td>
<td>File system lock</td>
</tr>
<tr>
<td>UFDOCK</td>
<td>Directory lock</td>
</tr>
<tr>
<td>BLKLOK</td>
<td>Block I/O lock (21.0)</td>
</tr>
<tr>
<td>SDLK</td>
<td>Segment directory lock</td>
</tr>
<tr>
<td>TRNLK</td>
<td>Transaction locks</td>
</tr>
<tr>
<td>UTLOK</td>
<td>Unit table lock</td>
</tr>
<tr>
<td>RATLOK</td>
<td>DISKRAT lock</td>
</tr>
<tr>
<td>DEVLCK</td>
<td>Device lock</td>
</tr>
<tr>
<td>NSSLCK</td>
<td>NSS database lock (21.0)</td>
</tr>
<tr>
<td>NETLCK</td>
<td>Network lock</td>
</tr>
<tr>
<td>NMMCLK</td>
<td>Network memory mapping lock</td>
</tr>
<tr>
<td>SLCCLK</td>
<td>SMLC lock</td>
</tr>
<tr>
<td>MOVLCK</td>
<td>Segment movement lock</td>
</tr>
<tr>
<td>EPFLCK</td>
<td>REPF database lock</td>
</tr>
<tr>
<td>SEGLCK</td>
<td>GETSN$/RTN$G$ lock</td>
</tr>
<tr>
<td>PAGLCK</td>
<td>Page fault lock</td>
</tr>
<tr>
<td>LOCEMK</td>
<td>Mutual excl. semaphore for locate</td>
</tr>
<tr>
<td>SEMSEMK</td>
<td>Mutual excl. semaphore for name semaphores</td>
</tr>
<tr>
<td>BCBLK</td>
<td>ROAM BCB lock</td>
</tr>
<tr>
<td>WCBLOK</td>
<td>ROAM WCB lock</td>
</tr>
<tr>
<td>SALSEM</td>
<td>Semaphore for system class storage</td>
</tr>
<tr>
<td>LON$SEM</td>
<td>Logout notification semaphore</td>
</tr>
<tr>
<td>IPC$SEM</td>
<td>IPC mutual excl. lock</td>
</tr>
<tr>
<td>LON$STA</td>
<td>Logout notification area pointer</td>
</tr>
<tr>
<td>SYSTHCB</td>
<td>Heap control block for system class storage</td>
</tr>
<tr>
<td>LOCUSR</td>
<td>Owner of LOCEMK</td>
</tr>
<tr>
<td>SEMUSR</td>
<td>Owner of SEMSEMK</td>
</tr>
<tr>
<td>BCBSMR</td>
<td>Owner of BCBSMK</td>
</tr>
<tr>
<td>WCBUSR</td>
<td>Owner of WCBSEMK</td>
</tr>
<tr>
<td>SALSEMR</td>
<td>Owner of SAL_SEM</td>
</tr>
<tr>
<td>ISCSEM</td>
<td>Mutual excl. semaphore for ISC database</td>
</tr>
<tr>
<td>ISCUSR</td>
<td>Owner of ISCSEM</td>
</tr>
<tr>
<td>REGSEM</td>
<td>Semaphore for ISC registration database</td>
</tr>
<tr>
<td>REGUSR</td>
<td>Owner of REGSEMK</td>
</tr>
<tr>
<td>SHSENM</td>
<td>SHSEG'T lock (ISC/SNA shared segments)</td>
</tr>
<tr>
<td>SHURUSR</td>
<td>Owner of SHRSENM</td>
</tr>
</tbody>
</table>

4.5. PTUSEG

PTUSEG(2,KSEG) (SEG 14)

PTUSEG(1,N) Owner of Page Map N
PTUSEG(2,N) Segment Number for Page Map N

4.6. PUDCOM

Starts at 6000/0.

<table>
<thead>
<tr>
<th>Name</th>
<th>Dflt</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>PGFFRE</td>
<td>Next frame pointer (ptr)</td>
</tr>
<tr>
<td>2</td>
<td>PGFEXT</td>
<td>Stack extension pointer (ptr)</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Reserved</td>
</tr>
<tr>
<td>6</td>
<td>PGFSB</td>
<td>Saved return PB on page fault stack (ptr)</td>
</tr>
<tr>
<td>10</td>
<td>CUSR</td>
<td>Current user number</td>
</tr>
<tr>
<td>11</td>
<td>PCBUSR</td>
<td>PCB index</td>
</tr>
</tbody>
</table>

Prime Restricted
<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>UTLBPT Pointer to unit table (ptr)</td>
</tr>
<tr>
<td>14</td>
<td>VRTSSW Virtual sense switches</td>
</tr>
<tr>
<td>15</td>
<td>LITE User virtual lights</td>
</tr>
<tr>
<td>16</td>
<td>DSKUSE Current disk request</td>
</tr>
<tr>
<td>17</td>
<td>INHPRF Inhibit-process-fault counter</td>
</tr>
<tr>
<td>20</td>
<td>ABSAVE Saved abort flags: see 4.1.</td>
</tr>
<tr>
<td>21</td>
<td>LOKOWN N1-locks owner table</td>
</tr>
<tr>
<td>22</td>
<td>OWNFS Owner count for FSLOK</td>
</tr>
<tr>
<td>23</td>
<td>R3QUIT Ring 3 quit inhibit count (&gt; 0 - quits inhibited)</td>
</tr>
<tr>
<td>24</td>
<td>R1QUIT Ring 1 quit inhibit count (&gt; 0 - quits inhibited)</td>
</tr>
<tr>
<td>25</td>
<td>PRVL Master privilege word</td>
</tr>
<tr>
<td>26</td>
<td>ASRCWD ASR controls</td>
</tr>
<tr>
<td>27</td>
<td>COMSWI Command input switch</td>
</tr>
<tr>
<td></td>
<td>100000 On (1 - on)</td>
</tr>
<tr>
<td></td>
<td>060000 Read state</td>
</tr>
<tr>
<td></td>
<td>00 - read left character next</td>
</tr>
<tr>
<td></td>
<td>10 - read right character next</td>
</tr>
<tr>
<td></td>
<td>01 - tab expans. in progress read left character next</td>
</tr>
<tr>
<td></td>
<td>11 - tab expans. in progress read right character next</td>
</tr>
<tr>
<td></td>
<td>010000 Last character was LF</td>
</tr>
<tr>
<td></td>
<td>007400 Unused</td>
</tr>
<tr>
<td></td>
<td>000377 Character saved</td>
</tr>
<tr>
<td>30</td>
<td>COMUNI Command input unit</td>
</tr>
<tr>
<td>31</td>
<td>COUSWI Command output switch</td>
</tr>
<tr>
<td>32</td>
<td>COUPTR COU LIN character pointer</td>
</tr>
<tr>
<td>33</td>
<td>COULIN Command output buffer (char(20))</td>
</tr>
<tr>
<td>45</td>
<td>ERRVEC Error vector (9) bin</td>
</tr>
<tr>
<td>56</td>
<td>SWITYP Pending software interrupts (See 4.9)</td>
</tr>
<tr>
<td>57</td>
<td>MSGCTL Message control</td>
</tr>
<tr>
<td></td>
<td>100000 Global message pending</td>
</tr>
<tr>
<td></td>
<td>040000 Personal message pending</td>
</tr>
<tr>
<td></td>
<td>020000 Rejected messages</td>
</tr>
<tr>
<td></td>
<td>010000 Deferred messages only</td>
</tr>
<tr>
<td></td>
<td>001000 Message was sent by a process</td>
</tr>
<tr>
<td>60</td>
<td>MPDB Pointer to process data block (ptr)</td>
</tr>
<tr>
<td>62</td>
<td>HMAPSK Pointer to stack page map (ptr)</td>
</tr>
<tr>
<td>64</td>
<td>BUFNEW Index of current locate buffer</td>
</tr>
<tr>
<td>65</td>
<td>XSAVE Temp save for X in fault entrances</td>
</tr>
<tr>
<td>66</td>
<td>LSAVE Temp save for L in process fault (bin(31))</td>
</tr>
<tr>
<td>70</td>
<td>RHDBUF Record header buffer (16)bin</td>
</tr>
<tr>
<td>110</td>
<td>USRETM Remainder of eligibility time-quantum</td>
</tr>
<tr>
<td>111</td>
<td>USRTS Default user timeslice</td>
</tr>
<tr>
<td>112</td>
<td>CURRTS Current user timeslice</td>
</tr>
<tr>
<td>113</td>
<td>LOUTCK Auto logout clock</td>
</tr>
<tr>
<td>114</td>
<td>CPLKCT Master CPU preference count</td>
</tr>
<tr>
<td>115</td>
<td>WTTIME WAIT$T time</td>
</tr>
<tr>
<td>116</td>
<td>IOUSED I/O time used</td>
</tr>
<tr>
<td>120</td>
<td>TIMDOG Real time watchdog timer (bin(31))</td>
</tr>
<tr>
<td>122</td>
<td>CPUUDOG CPU time watchdog timer (bin(31))</td>
</tr>
<tr>
<td>124</td>
<td>SLNODE Slave's master's node id</td>
</tr>
<tr>
<td>125</td>
<td>SLMAST Slave's master's user number</td>
</tr>
</tbody>
</table>
4.7. Shared Segments

<table>
<thead>
<tr>
<th>Segment</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>ED (Editor)</td>
</tr>
<tr>
<td>2001-2003</td>
<td>DBMS</td>
</tr>
<tr>
<td>2004-2011</td>
<td>SPSS, Scicards</td>
</tr>
<tr>
<td>2012</td>
<td>DBMS</td>
</tr>
<tr>
<td>2013</td>
<td>BASIC/VM</td>
</tr>
<tr>
<td>2014</td>
<td>Shared libraries (obsolete):</td>
</tr>
<tr>
<td></td>
<td>Area</td>
</tr>
<tr>
<td></td>
<td>100-277</td>
</tr>
<tr>
<td></td>
<td>300-377</td>
</tr>
<tr>
<td></td>
<td>1000-37777</td>
</tr>
<tr>
<td></td>
<td>40000-177777</td>
</tr>
<tr>
<td>2015</td>
<td>DPTX</td>
</tr>
<tr>
<td>2016</td>
<td>COBOL</td>
</tr>
<tr>
<td>2017</td>
<td>BASIC/VM</td>
</tr>
<tr>
<td>2020</td>
<td>MIDASPLUS writable shared segment</td>
</tr>
<tr>
<td></td>
<td>(177762-177777)</td>
</tr>
<tr>
<td>2021</td>
<td>FORMS library</td>
</tr>
<tr>
<td>2022-2023</td>
<td>PLP</td>
</tr>
<tr>
<td>2024-2025</td>
<td>POWERPLUS</td>
</tr>
<tr>
<td>2026-2027</td>
<td>FTS</td>
</tr>
<tr>
<td>2030-2037</td>
<td>Reserved for customers.</td>
</tr>
<tr>
<td>2040-2042</td>
<td>DBG</td>
</tr>
<tr>
<td>2043</td>
<td>SPSS, Scicards</td>
</tr>
<tr>
<td>2044</td>
<td>DSM</td>
</tr>
<tr>
<td>2045-2056</td>
<td>-</td>
</tr>
<tr>
<td>2057-2065</td>
<td>OAS (until rev 6.0)</td>
</tr>
<tr>
<td>Segment</td>
<td>Contents</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>2066-2067</td>
<td>DBMS</td>
</tr>
<tr>
<td>2070</td>
<td>OAS (until rev 6.0)</td>
</tr>
<tr>
<td>2071</td>
<td>SPSS, Scicards</td>
</tr>
<tr>
<td>2072</td>
<td>DISCOVER</td>
</tr>
<tr>
<td>2073-2077</td>
<td>EDMS</td>
</tr>
<tr>
<td>2100</td>
<td>OAS (until rev 6.0)</td>
</tr>
<tr>
<td>2102-2114</td>
<td>DBG</td>
</tr>
<tr>
<td>2115</td>
<td>SPL (until rev 20.2)</td>
</tr>
<tr>
<td>2122-2125</td>
<td>MIDASPLUS</td>
</tr>
<tr>
<td>2126-2127</td>
<td>FTS</td>
</tr>
<tr>
<td>2130-2137</td>
<td>MEDUSA</td>
</tr>
<tr>
<td>2140</td>
<td>EDMS, BP99</td>
</tr>
<tr>
<td>2141-2150</td>
<td>FED</td>
</tr>
<tr>
<td>2151-2153</td>
<td>CBL</td>
</tr>
<tr>
<td>2154-2161</td>
<td>EDMS, BP99</td>
</tr>
<tr>
<td>2162-2163</td>
<td>SPICE</td>
</tr>
<tr>
<td>2164-2166</td>
<td>SPOOL</td>
</tr>
<tr>
<td>2170-2177</td>
<td>Reserved for Customers, SCICARDS</td>
</tr>
<tr>
<td></td>
<td>2170-2171 INFO</td>
</tr>
<tr>
<td></td>
<td>2172-2175 MDS</td>
</tr>
<tr>
<td></td>
<td>2176 IFPS</td>
</tr>
<tr>
<td></td>
<td>2177 X.ED</td>
</tr>
<tr>
<td>2200-2203</td>
<td>ROAM</td>
</tr>
<tr>
<td>2204-2207</td>
<td>PRISAM</td>
</tr>
<tr>
<td>2210-2215</td>
<td>ESCAPE34, TAPS</td>
</tr>
<tr>
<td>2216</td>
<td>TAPS</td>
</tr>
<tr>
<td>2217-2220</td>
<td>ROAM</td>
</tr>
<tr>
<td>2221</td>
<td>MAGLIB (until 21.0)</td>
</tr>
<tr>
<td>2222</td>
<td>ROAM</td>
</tr>
<tr>
<td>2223-2224</td>
<td>DBMS</td>
</tr>
<tr>
<td>2225</td>
<td>ESCAPE34</td>
</tr>
<tr>
<td>2226</td>
<td>PRISAM</td>
</tr>
<tr>
<td>2230-2267</td>
<td>PRIMEWAY</td>
</tr>
<tr>
<td>2270-2276</td>
<td>INFORMATION</td>
</tr>
<tr>
<td>2277</td>
<td>PRISAM/DISCOVER</td>
</tr>
<tr>
<td>2300-2317</td>
<td>Reserved for customers</td>
</tr>
<tr>
<td>2310-2317</td>
<td>THEMIS</td>
</tr>
<tr>
<td>2320-2321</td>
<td>MIDASPLUS</td>
</tr>
<tr>
<td>2322</td>
<td>PRIMEWAY</td>
</tr>
<tr>
<td>2323</td>
<td>C (CC)</td>
</tr>
<tr>
<td>2330-2337</td>
<td>INFORMATION</td>
</tr>
<tr>
<td>2340</td>
<td>EMACS</td>
</tr>
<tr>
<td>2341-2347</td>
<td>INFORMATION/PDGS</td>
</tr>
<tr>
<td>2350-2355</td>
<td>PDGS</td>
</tr>
<tr>
<td>2356-2367</td>
<td>THEMIS</td>
</tr>
<tr>
<td>2370-2372</td>
<td>SNA</td>
</tr>
<tr>
<td>2377</td>
<td>PDMS</td>
</tr>
<tr>
<td>2400-2427</td>
<td>SNA RJE</td>
</tr>
<tr>
<td>2430-2442</td>
<td>EDMS</td>
</tr>
<tr>
<td>2443</td>
<td>PRIMEWAY</td>
</tr>
<tr>
<td>2444-2447</td>
<td>INFORMATION/CONNECTION</td>
</tr>
<tr>
<td>2476</td>
<td>SNA RJE</td>
</tr>
<tr>
<td>2477</td>
<td>ORACLE</td>
</tr>
<tr>
<td>2500-2521</td>
<td></td>
</tr>
<tr>
<td>2522-2534</td>
<td></td>
</tr>
<tr>
<td>Segment</td>
<td>Contents</td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
</tr>
<tr>
<td>2535</td>
<td>CBL</td>
</tr>
<tr>
<td>2536-2547</td>
<td>-</td>
</tr>
<tr>
<td>2550-2556</td>
<td>C</td>
</tr>
<tr>
<td>2557-2564</td>
<td>PDGS</td>
</tr>
<tr>
<td>2565-2567</td>
<td>-</td>
</tr>
<tr>
<td>2570-2573</td>
<td>ESCAPE34</td>
</tr>
<tr>
<td>2574-2575</td>
<td>-</td>
</tr>
<tr>
<td>2576</td>
<td>DBG</td>
</tr>
<tr>
<td>2577</td>
<td>-</td>
</tr>
<tr>
<td>2600-2601</td>
<td>ROAM/DDM</td>
</tr>
<tr>
<td>2602-2665</td>
<td>-</td>
</tr>
<tr>
<td>2666-2765</td>
<td>EPFs</td>
</tr>
<tr>
<td>6001</td>
<td>Per user linkage segment:</td>
</tr>
<tr>
<td>0-32777</td>
<td>Allocated</td>
</tr>
<tr>
<td>33000-40777</td>
<td>Product</td>
</tr>
<tr>
<td>41000-66777</td>
<td>VCOb LB (obsole te)</td>
</tr>
<tr>
<td>67000-67767</td>
<td>MIDAS (obsole te)</td>
</tr>
<tr>
<td>67770-67777</td>
<td>SPOOL (obsole te at 21)</td>
</tr>
<tr>
<td>70000-105777</td>
<td>BATCH</td>
</tr>
<tr>
<td>106000-112777</td>
<td>FORMS</td>
</tr>
<tr>
<td>113000-117777</td>
<td>ED</td>
</tr>
<tr>
<td>120000-131777</td>
<td>NPX</td>
</tr>
<tr>
<td>132000-177777</td>
<td>ABBREV</td>
</tr>
<tr>
<td>6006</td>
<td>Allocated</td>
</tr>
<tr>
<td>0-37777</td>
<td>Product</td>
</tr>
<tr>
<td>40000-70000</td>
<td>FTS (QPAKS)</td>
</tr>
<tr>
<td>70001-77777</td>
<td>MIDASPLUS</td>
</tr>
<tr>
<td>100000-177777</td>
<td>ROAM/DDM</td>
</tr>
<tr>
<td>6007</td>
<td>Allocated</td>
</tr>
<tr>
<td>0-47777</td>
<td>Product</td>
</tr>
<tr>
<td>50000-122777</td>
<td>ROAM</td>
</tr>
<tr>
<td>123000-137777</td>
<td>PRISAM</td>
</tr>
<tr>
<td>140000-177777</td>
<td>MAGLIB (until 21.0)</td>
</tr>
<tr>
<td>6010</td>
<td>ORACLE, EMACS, PRIMEWAY</td>
</tr>
<tr>
<td>6011</td>
<td>Per user data:</td>
</tr>
<tr>
<td>0-177777</td>
<td>ROAM</td>
</tr>
</tbody>
</table>

### 4.8. Semaphore allocation

-1 to -10  DBMS
-12 to -15  QPAKS
-16  MIDAS
-63 to -64  SPOOL
### 4.9. Software interrupt flags

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUTINT</td>
<td>Terminal quit</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>CPUINT</td>
<td>CPU watchdog timer</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>TIMINT</td>
<td>Real time watchdog timer</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>LOGINT</td>
<td>Forced logout</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>LONINT</td>
<td>Logout notification</td>
<td>004000</td>
<td>0800</td>
</tr>
<tr>
<td>CPSINT</td>
<td>Cross process signalling</td>
<td>002000</td>
<td>0400</td>
</tr>
<tr>
<td>IPCMWI</td>
<td>IPC message waiting</td>
<td>001000</td>
<td>0200</td>
</tr>
<tr>
<td>WRMINT</td>
<td>Warmstart software interrupt</td>
<td>000400</td>
<td>0100</td>
</tr>
<tr>
<td>PXCPSINT</td>
<td>Primix cps interrupt</td>
<td>000200</td>
<td>0080</td>
</tr>
</tbody>
</table>
### 4.10. Software Stack Frame

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 0 | Flags |
| 1 | Root Segment |
| 2 | Return |
| 3 | PB |
| 4 | Return |
| 5 | SB |
| 6 | Return |
| 7 | LB |
| 10 | Keys |
| 11 | Word after PCL |
| 12 | Reserved |
| 21 |   |
| 22 | Owner |
| 23 | ECB Ptr |
| 24 | Shortcall |
|   | Temps |
| 33 |   |
| 34 | OnUnit |
| 35 | Ptr |
| 36 | Cleanup |
| 37 | OnUnit |
| 40 | Extension |
| 41 | Headers |
| 42 | SPL library |
|   | scratch area |
| 47 |   |
| 50 | PL/I conditions |

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>C</td>
<td>D</td>
<td>X</td>
<td>U</td>
<td>S</td>
<td>L</td>
<td>E</td>
<td></td>
<td>FF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4-10 Prime Restricted
### 4.11. SVC interlude

**ENTRY DAC **

** SVC **

** OCT **

** CODE **

1 100000 1 => interlude call
2 040000 1 => bounce
3, 4 030000 Unused
5-16 007777 SVC number

### 4.12. UPCOM

User Profile COMmon.

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>projid</td>
<td>Project id (char (32))</td>
</tr>
<tr>
<td>0</td>
<td>collsize</td>
<td>Number of command levels</td>
</tr>
<tr>
<td>1</td>
<td>epfinvoc</td>
<td>Number of EPF invocations</td>
</tr>
<tr>
<td>2</td>
<td>rset2</td>
<td>Number static dtar2 segs</td>
</tr>
<tr>
<td>3</td>
<td>dyns2</td>
<td>Number of dynamic dtar2 segs</td>
</tr>
<tr>
<td>25</td>
<td>dyns2</td>
<td>Start of dynamic dtar segment ranges((0:3) bin)</td>
</tr>
<tr>
<td>30</td>
<td>diifus(16)</td>
<td>Remote id information</td>
</tr>
<tr>
<td></td>
<td>nodnam_ptr</td>
<td>Pointer to the node name in nnt of this entry (ptr)</td>
</tr>
<tr>
<td></td>
<td>user_id</td>
<td>The user id for this node (char(32) var)</td>
</tr>
<tr>
<td></td>
<td>password</td>
<td>The password for this user id (char(16) var)</td>
</tr>
<tr>
<td></td>
<td>project_id</td>
<td>The project to login under (char(32) var)</td>
</tr>
<tr>
<td>1350</td>
<td>dlns_count</td>
<td>Number of entries in use</td>
</tr>
<tr>
<td>1351</td>
<td>vcdata(16)</td>
<td>State info for each active NPX virtual circuit</td>
</tr>
<tr>
<td></td>
<td>void</td>
<td>Virtual circuit id for use with IPCF</td>
</tr>
<tr>
<td></td>
<td>node</td>
<td>Node number</td>
</tr>
<tr>
<td></td>
<td>spare</td>
<td>Save it for namtab ptr</td>
</tr>
<tr>
<td></td>
<td>vcstat</td>
<td>Virtual circuit status words for IPCF ((2) bin)</td>
</tr>
<tr>
<td></td>
<td>ns</td>
<td>Npx message sequence number to send next (mod 8)</td>
</tr>
<tr>
<td></td>
<td>nr</td>
<td>Npx message seq number last received over this VC</td>
</tr>
<tr>
<td></td>
<td>allocnt</td>
<td>Allocation count for this slave</td>
</tr>
<tr>
<td></td>
<td>slavno</td>
<td>For slave's id (char(6))</td>
</tr>
<tr>
<td></td>
<td>flags</td>
<td></td>
</tr>
<tr>
<td></td>
<td>000004</td>
<td>firsttime: shared by R$ALOC R$CALL R$BEGIN</td>
</tr>
<tr>
<td></td>
<td>000002</td>
<td>receive_posted: set when there is 1 rcv pending</td>
</tr>
<tr>
<td></td>
<td>000001</td>
<td>inprog: set while a RPCL is still pending</td>
</tr>
<tr>
<td>1651</td>
<td>npxv</td>
<td>NPX VC active in TRNRCV</td>
</tr>
<tr>
<td>1652</td>
<td>npxvany</td>
<td>Store any_handler entry (entry variable)</td>
</tr>
<tr>
<td>1654</td>
<td>upend</td>
<td>End of upcom</td>
</tr>
</tbody>
</table>

Prime Restricted
5. File System
The following describes the internal formats of all disk records for both the old and new file system partitions. Where possible, field names are the same as those used by the internal file system routines.

5.1. Diskrat Formats
Beginning Record Address (bra) = 2

5.1.1. 21
Rev 21:

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Len (= 31)</td>
</tr>
<tr>
<td>1</td>
<td>Rec_size</td>
</tr>
<tr>
<td>2</td>
<td>Disk_size</td>
</tr>
<tr>
<td>3</td>
<td>Heads</td>
</tr>
<tr>
<td>4</td>
<td>Spec_bits</td>
</tr>
<tr>
<td>5</td>
<td>Cylinders</td>
</tr>
<tr>
<td>6</td>
<td>Disk_vers</td>
</tr>
<tr>
<td>7</td>
<td>Npertk</td>
</tr>
<tr>
<td>8</td>
<td>block_alloc</td>
</tr>
<tr>
<td>9</td>
<td>disk_model</td>
</tr>
<tr>
<td>10</td>
<td>dts</td>
</tr>
<tr>
<td>11</td>
<td>first_free</td>
</tr>
<tr>
<td>12</td>
<td>DBS_address</td>
</tr>
<tr>
<td>13</td>
<td>Reserved</td>
</tr>
<tr>
<td>14</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>RAT</td>
</tr>
</tbody>
</table>

\[n\]
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Len</td>
<td>Diskrat header length</td>
</tr>
<tr>
<td>Rec_size</td>
<td>Physical record size (448 or 1040)</td>
</tr>
<tr>
<td>Disk_size</td>
<td>Number of records in partition</td>
</tr>
<tr>
<td>Heads</td>
<td>Number of heads in partition</td>
</tr>
<tr>
<td>Spec_bits</td>
<td>See section 5.1.3.</td>
</tr>
<tr>
<td>Cylinders</td>
<td>Number of cylinders</td>
</tr>
<tr>
<td>Disk_vers</td>
<td>Disk version</td>
</tr>
<tr>
<td>Nperrk</td>
<td>Number of sectors/track</td>
</tr>
<tr>
<td>block_alloc</td>
<td>Block allocation method</td>
</tr>
<tr>
<td></td>
<td>1 2</td>
</tr>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>Interleave</td>
</tr>
<tr>
<td></td>
<td>D Allocation direction:</td>
</tr>
<tr>
<td></td>
<td>0 forward</td>
</tr>
<tr>
<td></td>
<td>1 reverse</td>
</tr>
<tr>
<td></td>
<td>Interleave direction:</td>
</tr>
<tr>
<td></td>
<td>3 forward</td>
</tr>
<tr>
<td></td>
<td>1 reverse</td>
</tr>
<tr>
<td>Disk_model</td>
<td>Disk model type</td>
</tr>
<tr>
<td>Dts</td>
<td>Date/time shut down (for mirroring)</td>
</tr>
<tr>
<td>First_free</td>
<td>First record after RMA</td>
</tr>
<tr>
<td>DBS_address</td>
<td>Pointer to DBS (Dynamic Bad Spot):</td>
</tr>
<tr>
<td></td>
<td>1 8 9</td>
</tr>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>cylinder</td>
</tr>
<tr>
<td></td>
<td>head sector</td>
</tr>
<tr>
<td>RAT</td>
<td>Record Availability Table (Disk_size/16) (one bit/record)</td>
</tr>
</tbody>
</table>
### 5.1.2. Rev 19 and 20

#### Rev 20:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Len</td>
<td>Diskrat header length</td>
</tr>
<tr>
<td>Rec_size</td>
<td>Physical record size (448 or 1040)</td>
</tr>
<tr>
<td>Disk_size</td>
<td>Number of records in partition</td>
</tr>
<tr>
<td>Heads</td>
<td>Number of heads in partition</td>
</tr>
<tr>
<td>Spec_bits</td>
<td>See section 5.1.3.</td>
</tr>
<tr>
<td>Cylinders</td>
<td>Number of cylinders</td>
</tr>
<tr>
<td>Disk_vers</td>
<td>Disk version</td>
</tr>
<tr>
<td>Nperrtk</td>
<td>Number of sectors/track</td>
</tr>
<tr>
<td>RAT</td>
<td>Record Availability Table (Disk_size/16) (one bit/record)</td>
</tr>
</tbody>
</table>

#### Rev 19:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Len</td>
<td>Diskrat header length</td>
</tr>
<tr>
<td>Rec_size</td>
<td>Physical record size (448 or 1040)</td>
</tr>
<tr>
<td>Disk_size</td>
<td>Number of records in partition</td>
</tr>
<tr>
<td>Heads</td>
<td>Number of heads in partition</td>
</tr>
<tr>
<td>Spec_bits</td>
<td>See section 5.1.3.</td>
</tr>
<tr>
<td>Cylinders</td>
<td>Number of cylinders</td>
</tr>
<tr>
<td>Disk_vers</td>
<td>Disk version</td>
</tr>
<tr>
<td>RAT</td>
<td>Record Availability Table (Disk_size/16) (one bit/record)</td>
</tr>
</tbody>
</table>

#### 5.1.3. RAT specifier bits

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Crash; disk not shut down previous time</td>
</tr>
<tr>
<td>D</td>
<td>DOS modified or permanently broken</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>Crash; disk not shut down previous time</td>
<td>0000002</td>
<td>0002</td>
</tr>
<tr>
<td>D</td>
<td>DOS modified or permanently broken</td>
<td>0000001</td>
<td>0001</td>
</tr>
</tbody>
</table>
5.2. Record Header Formats
NOTE: record header formats are the same for all partitions. The format of a record header is a function of the physical record size.

1040-Word Records:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Rekrca</td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Rekbra</td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Rekdct</td>
</tr>
<tr>
<td>5</td>
<td>Rektyp</td>
</tr>
<tr>
<td>6</td>
<td>Rekpt</td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Rekbpt</td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Reklvl</td>
</tr>
<tr>
<td>11</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

| 15 |   |

448-Word Records:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Rekrca</td>
</tr>
<tr>
<td>1</td>
<td>Rekrca</td>
</tr>
<tr>
<td>2</td>
<td>Rekbra</td>
</tr>
<tr>
<td>3</td>
<td>Rekpt</td>
</tr>
<tr>
<td>4</td>
<td>Rekdct</td>
</tr>
<tr>
<td>5</td>
<td>Rektyp</td>
</tr>
<tr>
<td>6</td>
<td>Rekpt</td>
</tr>
<tr>
<td>7</td>
<td>Reserved</td>
</tr>
</tbody>
</table>

| 15 |   |

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rekrca</td>
<td>Record address of this record</td>
</tr>
<tr>
<td>Rekbra</td>
<td>Beginning Record Address (BRA of directory if first record)</td>
</tr>
<tr>
<td>Rekdct</td>
<td>Number data words in this record</td>
</tr>
<tr>
<td>Rektyp</td>
<td>Type of this file</td>
</tr>
<tr>
<td>Rekpt</td>
<td>RA next sequential record (0 if last)</td>
</tr>
<tr>
<td>Rekbpt</td>
<td>RA of previous record (0 if first)</td>
</tr>
<tr>
<td>Reklvl</td>
<td>Index level for dam files</td>
</tr>
<tr>
<td>Rekcnt</td>
<td>Number data words in this record</td>
</tr>
</tbody>
</table>
5.2.1. Rektyp

Rektyp is valid only in the first record of a file.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>File type:</td>
<td>000377</td>
<td>00FF</td>
</tr>
<tr>
<td>type</td>
<td>0 - SAM file</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - DAM file</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - SAM segment directory</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 - DAM segment directory</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 - Directory</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 - ACL directory</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 - ACAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 - CAM file</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If the file is the record zero bootstrap (BOOT) or the disk record availability table (DSKRAT or volume name) and the disk has a 1040 record size (Storage Module), bit 1 (:100000) of FILTYP will be set.
### 5.2.2. DBS Record Headers

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBS_rec_hdr_size</td>
<td>Size of DBS Record_header</td>
</tr>
<tr>
<td>num_entries</td>
<td>Size of DBS Record Header</td>
</tr>
<tr>
<td>next_record_addr</td>
<td>Number of DBS entries in record</td>
</tr>
<tr>
<td>bs_rm</td>
<td>Pointer to next record</td>
</tr>
</tbody>
</table>

**Field 1**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>num_entries</td>
<td>Number of DBS entries in record</td>
</tr>
<tr>
<td>next_record_addr</td>
<td>Pointer to next record</td>
</tr>
<tr>
<td>bs_rm</td>
<td>Array of badspot/remap matched pairs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cylinder</td>
<td></td>
</tr>
<tr>
<td>head</td>
<td></td>
</tr>
<tr>
<td>sector</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BS_cylinder</td>
<td>BSHead, BSsector</td>
</tr>
<tr>
<td>A</td>
<td>Already mapped by controller</td>
</tr>
<tr>
<td>RM_cylinder</td>
<td>RMHead, RMsector</td>
</tr>
<tr>
<td></td>
<td>Address of remap</td>
</tr>
</tbody>
</table>

**Field 2**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>num_entries</td>
<td>Number of DBS entries in record</td>
</tr>
<tr>
<td>next_record_addr</td>
<td>Pointer to next record</td>
</tr>
<tr>
<td>bs_rm</td>
<td>Array of badspot/remap matched pairs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cylinder</td>
<td></td>
</tr>
<tr>
<td>head</td>
<td></td>
</tr>
<tr>
<td>sector</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BS_cylinder</td>
<td>BSHead, BSsector</td>
</tr>
<tr>
<td>A</td>
<td>Already mapped by controller</td>
</tr>
<tr>
<td>RM_cylinder</td>
<td>RMHead, RMsector</td>
</tr>
<tr>
<td></td>
<td>Address of remap</td>
</tr>
</tbody>
</table>
5.3. UFD Header and Entry Formats

5.3.1. UFD header formats

<table>
<thead>
<tr>
<th>Rev 20, 21:</th>
<th>Rev 19:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ECW</td>
<td>ECW</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Owner_password</td>
<td>Owner_password</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Non_owner_password</td>
<td>Non_owner_password</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Reserved</td>
<td>Reserved</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Max_quota</td>
<td>Max_quota</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Dir_used</td>
<td>Dir_used</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Tree_used</td>
<td>Tree_used</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>Rec_time_prod</td>
<td>Rec_time_prod</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Prod_dtm</td>
<td>Prod_dtm</td>
</tr>
<tr>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td>Free_pos</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
</tr>
<tr>
<td>Hash_version</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Hash_tbl_size</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
</tr>
<tr>
<td>Hash_table</td>
<td></td>
</tr>
</tbody>
</table>

Prime Restricted
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECW</td>
<td>Entry control word. See 5.3.3.</td>
</tr>
<tr>
<td>Owner_password</td>
<td>Owner password (6 chars)</td>
</tr>
<tr>
<td>Non_owner_password</td>
<td>Non-owner password (6 chars)</td>
</tr>
<tr>
<td>Max_quota</td>
<td>Maximum quota</td>
</tr>
<tr>
<td>Dir_used</td>
<td>Quota used in this directory</td>
</tr>
<tr>
<td>Tree_used</td>
<td>Quota used in entire tree including subdirectories</td>
</tr>
<tr>
<td>Rec_time_prod</td>
<td>Record-time product</td>
</tr>
<tr>
<td>Prod_dtm</td>
<td>DTM-record product (FS date format)</td>
</tr>
<tr>
<td>Free_pos</td>
<td>Free pointer</td>
</tr>
<tr>
<td>Hash_version</td>
<td>Version of hash function</td>
</tr>
<tr>
<td>Hash_tbl_size</td>
<td>Number of entries in hash table</td>
</tr>
<tr>
<td>Hash_table</td>
<td>The hash table</td>
</tr>
</tbody>
</table>
### 5.3.2. UFD Entry Formats

#### 5.3.2.1. File entries

<table>
<thead>
<tr>
<th>Rev 20:</th>
<th>19.0:</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ECW</td>
<td>ECW</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>BRA</td>
<td>BRA</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Log_type</td>
<td>Log_type</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>DTB</td>
<td>DTB</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Protec</td>
<td>Protec</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>ACL_pos</td>
<td>ACL_pos</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>DTM</td>
<td>DTM</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>File_info</td>
<td>File_info</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Name_length</td>
<td>Name_length</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Name</td>
<td>Name</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>n+1</td>
<td>n+1</td>
</tr>
<tr>
<td>DTC</td>
<td>DTC</td>
</tr>
<tr>
<td>n+3</td>
<td>n+3</td>
</tr>
<tr>
<td>DTL</td>
<td>DTL</td>
</tr>
<tr>
<td>n+5</td>
<td>n+5</td>
</tr>
<tr>
<td>Link</td>
<td>Link</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>ECW</td>
<td>Entry Control Word</td>
</tr>
<tr>
<td>BRA</td>
<td>Beginning Record Address</td>
</tr>
<tr>
<td>Log_type</td>
<td>Logical type</td>
</tr>
<tr>
<td>DTB</td>
<td>Date/Time Backed-up</td>
</tr>
<tr>
<td>Protec</td>
<td>Protection keys</td>
</tr>
<tr>
<td>ACL_pos</td>
<td>ACL position</td>
</tr>
<tr>
<td>DTM</td>
<td>Date/Time Modified</td>
</tr>
<tr>
<td>File_info</td>
<td>See sect 5.3.2.4</td>
</tr>
<tr>
<td>Name_length</td>
<td>Length of name</td>
</tr>
<tr>
<td>Name</td>
<td>Name of file entry (32 characters)</td>
</tr>
<tr>
<td>DTC</td>
<td>Date/Time Created</td>
</tr>
<tr>
<td>DTL</td>
<td>Date/Time Last accessed</td>
</tr>
<tr>
<td>Link</td>
<td>Link to next entry on chain</td>
</tr>
</tbody>
</table>
### 5.3.2.2. ACAT entries

**Rev 20:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECW</td>
<td>Entry Control Word</td>
</tr>
<tr>
<td>DTB</td>
<td>Date/Time Backed-up</td>
</tr>
<tr>
<td>ACL_pos</td>
<td>ACL position</td>
</tr>
<tr>
<td>DTM</td>
<td>Date/Time Modified</td>
</tr>
<tr>
<td>File_info</td>
<td>See sect 5.3.2.4</td>
</tr>
<tr>
<td>Name_length</td>
<td>Length of name</td>
</tr>
<tr>
<td>Name</td>
<td>Name of file entry (32 characters max)</td>
</tr>
<tr>
<td>DTC</td>
<td>Date/Time Created</td>
</tr>
<tr>
<td>DTL</td>
<td>Date/Time Last accessed</td>
</tr>
<tr>
<td>Link</td>
<td>Link to next entry on chain</td>
</tr>
</tbody>
</table>

**Rev 19:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECW</td>
<td>Entry Control Word</td>
</tr>
<tr>
<td>DTB</td>
<td>Date/Time Backed-up</td>
</tr>
<tr>
<td>ACL_pos</td>
<td>ACL position</td>
</tr>
<tr>
<td>DTM</td>
<td>Date/Time Modified</td>
</tr>
<tr>
<td>File_info</td>
<td>See sect 5.3.2.4</td>
</tr>
<tr>
<td>Name_length</td>
<td>Length of name</td>
</tr>
<tr>
<td>Name</td>
<td>Name of file entry (32 characters max)</td>
</tr>
<tr>
<td>DTC</td>
<td>Date/Time Created</td>
</tr>
<tr>
<td>DTL</td>
<td>Date/Time Last accessed</td>
</tr>
<tr>
<td>Link</td>
<td>Link to next entry on chain</td>
</tr>
</tbody>
</table>
5.3.2.3. DBS entries

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file_hd_size</td>
<td>Size of DBS file header</td>
</tr>
<tr>
<td>version</td>
<td>Version number</td>
</tr>
<tr>
<td>DBS_entry_size</td>
<td>Size of a DBS entry</td>
</tr>
<tr>
<td>number_recs</td>
<td>Number of records in file</td>
</tr>
<tr>
<td>num_badspots</td>
<td>Number of badspots in file</td>
</tr>
<tr>
<td>num_remaps</td>
<td>Number of remap records</td>
</tr>
<tr>
<td>control_bits</td>
<td>Various flags:</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

P  Primos modified this last
C  The controller modified this last
### 5.3.2.4. File Information Bits

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>Long RAT header</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>D</td>
<td>Dumped; file has been backed up</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>M</td>
<td>File has been modified under DOS</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>S</td>
<td>Special file</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>RW</td>
<td>Read/Write lock: 00 - system default 01 - n</td>
<td>006000</td>
<td>0C00</td>
</tr>
<tr>
<td></td>
<td>readers</td>
<td>1 writer 10 - n readers &amp; 1 writer 11 - n</td>
<td></td>
</tr>
<tr>
<td></td>
<td>readers &amp; n writers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>Truncated by FIX_DISK</td>
<td>001000</td>
<td>0200</td>
</tr>
<tr>
<td>File type</td>
<td>File type:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 - SAM file</td>
<td>000377</td>
<td>00FF</td>
</tr>
<tr>
<td></td>
<td>1 - DAM file</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - SAM segment directory</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 - DAM segment directory</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 - Directory</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 - ACL directory</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 - ACAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 - CAM file</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 5.3.3. Entry Control Word (ECW)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>Type of entry:</td>
<td>177400</td>
<td>FF00</td>
</tr>
<tr>
<td></td>
<td>0 - old dir header</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - directory header</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - vacant entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 - file entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 - access category (named ACL)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 - ACL</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 - directory index block</td>
<td></td>
<td></td>
</tr>
<tr>
<td>size</td>
<td>Size of the entry</td>
<td>000377</td>
<td>00FF</td>
</tr>
</tbody>
</table>
5.4. File system date format

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>Year + 1900 (100-127 = 2000-2027)</td>
<td>177000</td>
<td>FE00</td>
</tr>
<tr>
<td>Month</td>
<td>Month (1 = Jan)</td>
<td>000740</td>
<td>01E0</td>
</tr>
<tr>
<td>Day</td>
<td>Date</td>
<td>000037</td>
<td>001F</td>
</tr>
<tr>
<td>Time</td>
<td>Quad-seconds since midnight</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6. SUBROUTINES
For additional information, see the Subroutines Reference Guide [58], [59], [60], [61] or the Primenet Guide [45]. Volume and page numbers follow the documented routines (P indicates Primenet Guide reference). Any routine not marked is not released. Use of unreleased routines must be cleared with the owning group. Users of unreleased routines should realize that those routines may be modified or removed without notice.

6.1. System routines - Supervisor Calls
Primos ring 0 gates and ring 3 entries.

**AB$WS()** returns(bin) ill-2-3
  if substr(ab$sw$(i) = 1) then abbrev enabled;
  Returns the cold start setting of the global abbrev enable switch.

**AC$CAT**(char(128)var, char(32)var, bin) ll-2-3
  call ac$cat(object_pathname, acat_name, code);
  Add a file to an access category.

**AC$CAT0**(char(32)var, char(32)var, bin) [NOT RELEASED]
  call ac$cat0(object_name, category_name, code);
  Place an object into an access category.

**AC$CHG**(char(128)var, ptr, bin) ll-2-5
  call ac$chg(object_pathname, addr(acl_struct), code);
  Modify an existing ACL.

**AC$DEV**(bit(1), bin) [NOT RELEASED]
  call ac$dev(on_or_off, code);
  Enable/Disable device ACLs.

**AC$DFT**(char(128)var, bin) ll-2-7
  call ac$dft(object_pathname, code);
  Set default protection.

**AC$DFT0**(char(32)var, bin) [NOT RELEASED]
  call ac$dft0(object_name, code);
  Protect an object with default access rights.

**AC$LKS**(char(128)var, char(128)var, bin) ll-2-9
  call ac$lik(target_object, reference_object, code);
  Protect one file like another.

**AC$LST**(char(128)var, ptr, bin, char(128), bin, bin) ll-2-11
  call ac$lst(object_pathname, addr(acl_struct), max_acl_entries,
             acl_name, acl_type, code);
  Read an ACL.

**AC$LST0**(char(32)var, ptr, bin, char(128)var, bin, bin) [NOT RELEASED]
  call ac$lst0(object_name, logical_acl_ptr, max_entry_count,
              acl_name, acl_type, code);
  Return the contents of an ACL in logical format.

**AC$RVT**(bin) ll-2-13
  call ac$rvt(code);
  Convert current ACL dir to password dir.
AC$SET (bin, char(128)var, ptr, bin) II-2-15
   call ac$set(key, object_pathname, addr(acl_struc), code);
   Create or replace an ACL.

AC$SET0 (bin, char(32)var, ptr, bin) [NOT RELEASED]
   call ac$set0(key, object_name, acl_ptr, code);
   Create an ACL for an object, given the object's entryname.

ACCOM$ (bit(16) aligned, fixed bin, char(*) var) [NOT RELEASED]
   call accom$(switch, unit, action);
   Saves or restores cominput switch and file unit. action is "save" or "restore".

ADBSP$ (bin, bin(31), bin) [NOT RELEASED]
   call adbsp$(work_pdev, physical_badspot, error_code);
   Add a badspot to the Rev 21 (or greater) disk specified.

ADISK$ (ptr, bin, bin) [NOT RELEASED]
   call adisk$(struc_ptr, list_max, code)
   Return a list of the locally ASSIGNED disks.

AD_CMD (char(256)var, bin) [NOT RELEASED]
   call ad_cmd(cam_args, com_status);
   ADDISK command.

ALC$RA (fixed bin(31), ptr options(short)) III-4-16
   call alc$ra(words_to_allocate, rtn_function_ptr);
   Allocate space in process class storage for return function data.

ALLOC (fixed bin(31)) [NOT RELEASED]
   call alloc(size);
   Allocates size bytes on the callers stack.

ALOC$S entry (fixed bin (15), ptr, bit (1) aligned) options (shortcall (4)) III-4-3
   call alloc$ (size_to_allocate, pointer_to_space, contiguous);
   Allocates size_to_allocate half-words on stack.

ALS$RA (char(*), fixed bin(31), fixed bin(31)); III-4-21
   call als$ra(function_result_str, char_size_of_str,
               rtn_function_addr);
   Allocate space and set return data for return function.

AMLC$ (char(32) nonvarying, (3) fixed bin, bit(16) aligned, fixed) [NOT RELEASED]
   call amlc$(protocol, line_config_word, arg_flag, status);
   Set the line configuration for an amlic line.

AMT$DTR3 (bin) returns(bin(31)) [NOT RELEASED]
   DTAR3_storage_used = amt$dtr3(code);
   Find amount of DTAR 3 storage used by the caller.

APPEND (char(*)var, bin, char(*)var, bin, bin[]) returns(bit(1)) [NOT RELEASED]
   full = append(string, max_size_of_string, new,
                new_start, new_length);
   Appends new to end of string.

APROTO (bin, char(6), code) [NOT RELEASED]
   call aproto(line, protocol, code)
   Select protocol for an async line. (OBSCOLETE. Removed at 22.0)
APFX$ (char(128) var, char(128) var, char(32) var, bin) II-4-4
    call apfx$ (in_pathname, out_pathname, suffix, code);
    Append a suffix to a pathname. code = -1 -> suffix already present.

AR$ALC entry (ptr, fixed bin (31)) returns (ptr) [NOT RELEASED]
    storage_ptr = ar$alc (area_ptr, size_to_allocate);
    Allocates storage in area previously defined by AR$IN.

AR$FRE (ptr, ptr) [NOT RELEASED]
    call ar$fre (area_ptr, storage_ptr);
    Frees storage from a previously defined area.

AR$IN (ptr, fixed bin (31)) [NOT RELEASED]
    call ar$in (area_ptr, area_size);
    Initializes area for use by the area manager package.

AR$SIZ (ptr, ptr) returns(bin(31)) [NOT RELEASED]
    area_size = ar$siz(area_ptr, block_ptr);
    Return size of allocated area.

ARW$ALC (ptr, bin(31), bin) returns(ptr) [NOT RELEASED]
    alloc_ptr = arw$alc(area_ptr, size, code);
    Allocate in area.

ARW$FRE (ptr, ptr, bin) [NOT RELEASED]
    call arw$fre(area_ptr, block_ptr, code);
    Free block in area.

ARW$IN (ptr, bin(31), bin) [NOT RELEASED]
    call arw$in(area_ptr, area_size, code);
    Initialize area header.

ARW$SIZ (ptr, ptr, bin) returns(bin(31)) [NOT RELEASED]
    block_size = arw$siz(area_ptr, block_ptr, code);
    Return size of allocated area.

AS$GET (bin, bin, ptr, bin, bin) [NOT RELEASED]
    call as$get(line_number, version, par_list_ptr, par_list_len,
                error_code)
    Returns async line information.

AS$LIN (bin, bin) [NOT RELEASED]
    call as$lin(line_number, error_code);
    Returns the current user's line number.

ASDXE$ (bin, char(80), bin, bin) [NOT RELEASED]
    call asdxe$ (key, line, state, code);
    Assign disk and other peripheral devices except magtape.

ASNLN$ (bin, bin, char(*), bin, bin, bin) IV-8-21
    call asnln$ (key, amlc_line, protocol, amlc_config, lword, code);
    Assign AMLC line. Key = 0 - unassign; 1 - assign; 2 - unassign all.

ASNM$ (bit(16), char(256) var, bin(15)) [NOT RELEASED]
    call asnm$ (no_mags, user_assign_cmd_line, return_status);
    Assign magnetic tape drive.
ASSUR$ (bin) returns(bit(1)) III-2-17
   enough_time = assur$(desired_mseconds);
   Allow a user process to assure it has a certain amount of cpu time left.

AT$ (bin, char(128)var, bin) II-3-3
   call AT$(key, path_name, code);
   Attach by pathname. Key = K$SETH, K$SETC.

AT$ABS (bin, char(32)var, char(39)var, bin) II-3-6
   call AT$ABS(key, partition_name, dir_name, code);
   Attach to top-level dir on given partition. Dir_name includes password. Key = K$SETH, K$SETC.

AT$ANY (bin, char(39)var, bin) II-3-8
   call AT$ANY(key, dir_name, code);
   Attach to top-level dir. See AT$ABS for notes.

AT$ANY0 (bin, char(39)var, bin) [NOT RELEASED]
   call AT$ANY0(key, dir_name, code);
   Do an old-style attach scan.

AT$HOM (bin) II-3-10
   call AT$HOM(code);
   Return to home dir.

AT$INV (bin, bin) [NOT RELEASED]
   call AT$INV(key, code);
   Invalidates specified attach point(s). Key = K$KURA, K$HOMA, K$MINIA, K$ALL.

AT$LDEV (bin, bin, char(39)var, char(32)var, bin) II-3-11
   call AT$LDEV(key, ldev, dir_name, partition, code);
   Attach to top-level directory given the ldev of the partition.

AT$OR (bin, bin) II-3-13
   call AT$OR(key, code);
   Return to origin dir. Key = K$SETH, K$SETC.

AT$REL (bin, char(39)var, bin) II-3-15
   call AT$REL(key, dir_name, code);
   Attach relative to current dir. Key = K$SETH, K$SETC.

AT$TMP returns(bit(1)) [NOT RELEASED]
   swap_completed = AT$TMP();
   Save or restore the current attach point.

ATCH$$ (char(32), bin, bin, char(6), bin, bin) (svc = 1500) II-A-2
   call ATCH$$ (ufd_name, name_len, ldisk num, password, key);
   Attach to UF D. (Obsolete; use AT$, AT$ABS, AT$ANY, AT$HOM, AT$OR, AT$REL)

ATLIST (fixed bin, (12) char(32) var, fixed bin, char(32), fixed bin, char(6), fixed bin, fixed bin) [NOT RELEASED]
   call ATLIST (key, disk list, disk_count, dir_name,
                   dir_name_len, password, found_index, code);
   Search a list of dirs on a given system (NPX only).

AT$SHRS (bin(31), bin, ptr, ptr, bin) [NOT RELEASED]
call atshr$(unique_seg_id, req_accesses, true_seg_ptr, 
detz2_seg_ptr, code);
  Attach to a segment allocated by gtsnr$.

AUSCUR (bin, char(256), bin) [NOT RELEASED]
call au$cur(user, dest, code);
  Access current log entry for a given user.

AUSRDR (bin, bin) [NOT RELEASED]
call au$drn(context, code)
  Shut down an OS_LOG phantom.

AUSGET (ptr, bin, bin) [NOT RELEASED]
call au$get(dest, mpage, code);
  Return copy of current log buffers for LOGANAL utility of OS_LOG.

AUSSTART (bin, bin) [NOT RELEASED]
call au$start(return_phantom, code);
  Start up the OS_LOG utility phantom.

AUSSTAT (bin, char(128)var)returns(bit(16)) [NOT RELEASED]
  status = au$stat(user, file);
  Show current status of OS_LOG phantom.

AUSSTRTO (bin, bin) [NOT RELEASED]
call au$strt0(rtn_phy, code);
  Start up the OS_LOG utility phantom.

AUSTSK (bin, bin, char(160)var, bin, char(32), bin) [NOT RELEASED]
call au$stsk(type, task_type, command, status, caller_id, 
cpl_taskno);
  Assembles OS_LOG login/logout message types before logging.

AUSWRT (bin, bin, bin) [NOT RELEASED]
call au$wrt(log_file, prwf_rtn_code, status);
  Write to OS_LOG log file & wait for a data buffer.

BATCH$ (char(""), fixed bin, fixed bin, char(""), fixed bin, fixed bin, fixed bin) [NOT RELEASED]
call batch$(filename, name_length, unit, user_name,
user_name_length, user_num, status)
  Spawns a phantom under any user id. Privileged. (OBSCOLETE; use SPAWNS$)

BCKUPB$ (ptr) [NOT RELEASED]
call bckupb$(target_sb);
  Back Up Return PB For Ring 0 Restart.

BDSATT (char(""), bin, bin, bin) [NOT RELEASED]
call bd$att(name, length, dev, code);
  Block device 'ATTACH' subroutine. (DPTX)

BDSDET (bin, bin) [NOT RELEASED]
call bd$det(device, code);
  Block device detach subroutine. (DPTX)

BDSINF (bin, bin, char(""), bin, (10)bin, bin) [NOT RELEASED]
call bd$inf(device, key, buffer, buf_len, stat_protocol, code);
  Block device information & status subroutine. Key = k$inf, k$infld, k$infs. (DPTX)
BD$INP (bin, bin, char(4), bin, (10)bin, bin, bin) [NOT RELEASED]
call bd$inp(device, key, buffer, buf_len, status_protocol, code, 
        wait_period);
    Block device input subroutine. Key = k$wait, k$snowa, k$s watt. (DPTX)

BD$LST (bin, char(8), bin, ('*')bin, bin, bin) [NOT RELEASED]
call bd$lst(key, name, name_len, data_buffer, data_len, code);
    Block device interface description routine. Key = k$infid, k$inin, k$itlt, k$ipat, k$plat,
        k$patd, k$bsys. (DPTX)

BD$OUT (bin, bin, char(4), bin, (10)bin, bin) [NOT RELEASED]
call bd$out(device, key, buffer, len, status_protocol, code);
    Block device output subroutine. Key = k$xmtf, k$xmtd, k$msk, k$rawd. (DPTX)

BD$SET (bin, bin, bin) [NOT RELEASED]
call bd$set(device, key, code);
    Block device attribute-setting subroutine. Key = k$inwt, k$iwof, k$tabl, k$abt, k$spdi,
        k$spdo, k$srm, k$srm, k$spa2p, k$spa2q. (DPTX)

BIN$SR (char(4) var, fixed bin, ptr, ptr, ptr, fixed bin) [NOT RELEASED]
call bin$sr( entry, entry_size, start_ptr, end_rel, spot_ptr, 
        code);
    Binary search on ordered table (one segment restriction).

BM$GET (bin, char(4), bin, bin, bin) [NOT RELEASED]
call bm$get(key, buffer, buffer_length, chars_returned, 
        return_code);
    Gets data from the user's TFLIO input buffer to the user buffer.

BM$MOD (bin, char(1), bin) [NOT RELEASED]
call bm$mod(key, end_char, return_code);
    Switch user's terminal line between character and block mode.

BM$QRY (bin) [NOT RELEASED]
call bm$qry(mode);
    Queries whether a user's terminal is in block mode.

BM$RDY returns (bin) [NOT RELEASED]
buf_sem_count = bm$rdy();
    Returns the count field of a user's terminal buffer semaphore.

BM$SCAN (char(4), bin, char(1), bin) [NOT RELEASED]
call bm$scan(buffer, length_of_scan_areas, scan_char, offset);
    Search for a single character in a string from a offset.

BREAK$ (bin) (svc = 0507) III-3-50
        call break$(value);
    Inhibits or enables quits. Value = 1 to inhibit breaks.

BSCMAN (fixed bin(15), fixed bin(15), (256) fixed bin(15)) [NOT RELEASED]
call bscman (error_line, option, protocol_table);
    Initiate the bisynchronous communications. Handler for the IBM 3270 protocols (DPTX).

C1IN (char) (svc = 0601) III-3-5
   call c1in(character);
    Get one char (right justified) from terminal or command file.
C1NS (char(2), bit(1), bit(1)) III-3-7
  call clin$(retchar, echo, termonly);
  Single character command input.

C1NE$ (char(2)) III-3-9
  call clne$(rtnc_char)
  Input single character with no echo.

CALAC$ (char(128)var, ptr, char(80), char(80)var, bin) II-2-17 returns(bit(1))
  have_access = calac$(pathname, addr(id_struc), access_required,
                     access_gotten, code);
  Calculate access available.

CALAC$0 (char(32)var, ptr, char(47)var, char(47)var, bin) returns (bit (1)) [NOT RELEASED]
  have_access = calac$0(name, id_ptr, access_needed,
                        access_gotten, code);
  Calculate accesses available on a named object.

CALFC$ (ret_pb, not_in_range) options (shortcall (4)) [NOT RELEASED]
  call calfc$ (PB_in_question, not_in_range);
  Does magnitude check on a return PB to see if it is within the ring 3 pointer fault table.

CAT$DL (char(128)var, bin) II-2-19
  call cat$dl(acat_pathname, code);
  Delete an access category.

CAT$DL0 (char(32)var, bin)
  call cat$dl0(category_name, code);
  Delete an access category.

CE$BRD returns(bin) II-6-3
  maximum_command_env_breadth = ce$brd();
  Return maximum command level breadth for this user.

CE$DPT returns(bin) II-6-4
  maximum_command_env_depth = ce$dpt();
  Return maximum command level depth for this user.

CF$EXT (bin, bin(31), bin(31), bin) [NOT RELEASED]
  call cf$ext(unit, req_peof, act_peof, code);
  Moves physical end of file for a contiguous file.

CF$REM (bin, bin, bin, bin) [NOT RELEASED]
  call cf$rem(unit, buffer, length, code);
  Returns a copy of the on-disk extent map.

CF$SME (bin, bin, bin) [NOT RELEASED]
  call cf$sme(unit, min_ext_len, code);
  Sets minimum extent length for a contiguous file.

CFI returns(bit(16)) [NOT RELEASED]
  char_avail = cfi();
  Program to check if there is a character in the terminal buffer.

CH$FX1 (char (*) var, fixed bin(15) [], fixed bin(15)) III-6-3
  call ch$fx1 (string_to_convert, result [], nonstandard_code));
  Convert character varying string to fixed bin(15).
CHSFX2 (char(*) var, fixed bin(31) [, fixed bin(15)]) III-6-5
  call chsfx2 (string_to_convert, result [, nonstandard_code]);
  Convert character varying string to fixed bin(31).

CHSFX2 (char(*) var, fixed bin (31) [, fixed bin (15)]) III-6-7
  call chsfx2 (string_to_convert, result [, nonstandard_code]);
  Convert character varying string to fixed bin(31) as hex.

CHS$MOD (fixed bin, fixed bin, fixed bin) II-4-6
  call chsmod (key, unit, code);
  Change the open mode of an open file. Key = K$READ, K$WRIT, K$RDWR.

CHS$OC2 (char(*) var, fixed bin(31) [, fixed bin(15)]) III-6-9
  call chsoc2 (string_to_convert, result [, nonstandard_code]);
  Convert character varying string to fixed bin(31) as octal.

CHBK$ (bin, bin(31), bin, bin) [NOT RELEASED]
  call chbk$ (key, uri, unit, status);
  Routine to check status of asynchronous writes.

CHG$PW (char(16)var, char(16)var, code) III-2-18
  call chg$pw(old_password, new_password, code);
  Change login password.

CHG$SA (char(32) var, fixed bin) [NOT RELEASED]
  call chg$sa (new_administrator_id, code);
  Changes the user ID of the system administrator. Privileged.

CIRLOG (bin, bin, bin, bin, bin, bin, bin, bin, bin, bin) [NOT RELEASED]
  call cirlog(entry_type, subroutine_name, arg1, arg2, arg3,
              arg4, arg5, arg6, arg7);
  Debug routine for NPX.

CKDyn$ (char(32)var, bin) III-2-4
  call ckdyn$(routine_name, code)
  Check for the existence of a dynamic entrypoint.

CKNDNM (char(32)var, bin, bin) [NOT RELEASED]
  call ckndnm(node_name, vcix, code);
  Subroutine to check the validity of node name on the name table.

CL$F0 (char(32)var, bin) [NOT RELEASED]
  call cl$fn0(entryname, code);
  Close an open file by name.

CL$FN (char(128)var, 1, 2 bin, 2 (*)bit(16), bin, bin) II-4-7
  call cl$fnr(pathname, rtn_list, first_file_unit, code);
  Close a file by name and return a bit varying indicating closed units.

CL$FR (char(32)var, 1, 2 bin, 2 (*) bit (16), bin, bin) [NOT RELEASED]
  call cl$fr0(entryname, rtn_list, first_file_unit, code);
  Close a file by name and return a bit varying indicating closed units.

CL$GET (char(*)var, bin, bin) III-3-10
  call cl$get(buffer, max_buffer_len, code);
  Read a line of text from terminal or command file.
CL$GET_EV (bin, entry, bin) [NOT RELEASED]
    call cl$get_ev(key, routine, code)
    Command loop get entry variable. Key = K$COMMAND_PROCESSOR,
    K$COMMAND_LINE_READER, K$COMMAND_PROMPT

CL$PAR (bit(16) aligned, char(*) var, char(*) var, bin, 1 ..., bin, bin) [NOT RELEASED]
    call cl$par (keys, source_str, token_str, token_str_size, info,
                   next_ch, status);
    Parse source_str according to basic command line rules.

CL$PIX (bit(16), char(*)var, ptr, bin, char(*)var, ptr, bin, bin, bin, ptr) II-6-5
    call cl$pix(keys, caller_name, addr(picture), pixel_size,
                 input_line, addr(com_line_struct), pic_error_loc,
                 bad_index, code);
    Parse command line. See Subroutine Ref Guide for code values.

CL$SET_EV (bin, entry, bin) [NOT RELEASED]
    call cl$set_ev (key, routine, code);
    Command line set entry variable. Same keys as CL$GET_EV.

CLO$FN (char(128) var, fixed bin) II-4-9
    call clo$fn(pathname, code);
    Close an open file by name.

CLO$FU (fixed bin, fixed bin) II-4-10
    call clo$fu(unit, code);
    Close an open file by unit.

CLRLV$ [NOT RELEASED]
    call clrlv$;
    Clear the existing command level.

CMD_POST (ptr options(short)) [NOT RELEASED]
    call cmd_post_invk (epf_smt_ptr);
    Post-program invocation cleanup.

CMD_PRE_(bin) [NOT RELEASED]
    call cmd_pre_ (code);
    Perform pre-program invocation initialization.

CMLV$E II-5-5
    call cmlv$e;
    Call a new command level with error prompt. (see comlv$)

CMREAD$ (char(80), bin, (59)bin, bin, bin) [NOT RELEASED]
    call cmread$ (com_line, com_state, ucompar, maxlen, code);
    Old style command line parser.

CNAME$ (char(*), bin, char(*), bin, bin) (svc = 1515) II-4-11
    call cnam$ (old_name, old_name_len, new_name, new_name_len,
                code);
    Change the name of a file.

CNINS (char(*), bin, bin) (svc = 0604) III-3-13
    call cin$ (buffer, char_count, rtn_char_count);
    Input char_count characters.
CNSIG$ (bin) III-7-19
   call CNSIG$(code);
   Call more on-units; continue to signal condition.

CO$GET (bin, bin) III-3-52
   call CO$GET (comoutput_funit, command_stream_sw);
   Retrieve the comoutput unit number and value of COUSW.

COM$AB (char(1024) var, bin, bin) III-2-20
   call COM$AB(command, command_size, code);
   Interlude to invoke command abbreviation processor.

COMANL (svc = 0600) III-3-15
   call COMANL;
   Read a line of text. (Obsolete; use CL$GET)

COMIS$ (char(32), bin, bin, bin) (svc = 1516) III-3-53
   call COMIS$(file_name, file_name_len, file_unit, code);
   Switch between terminal and command file for input.

COMLV$ III-5-6
   call COMLV$;
   Call a new command level.

COMOS$ (bit(16), char(32), bin, bin, bin) (svc = 1523) III-3-55
   call COMOS$(key, file_name, file_name_len, reserved, code);
   Change terminal output to terminal or file. Key Bits: 01 - TTY off; 02 - TTY on; 10 - file off;
   20 - file on; 40 - append if file on, close if file off; 100 - truncate if file on.

CPS$ (char(*) var, bin, bin, 1, 2 bit(1), 2 bit(1), 2 bit(14), ptr, ptr) II-6-9
   call CPS$(command_line, status, com_status, flags, local_variable_ptr, rtn_function_ptr);
   Invoke the user's currently specified command processor.

CP$ITR(char(1024) var, entry, bin, bit(5), bit(3), bit(1), bin) [NOT RELEASED]
   call CP$ITR (com_line, executer, eq_position, default_types, exp_wild, vy_default, status)
   Command language iteration handler.

CP$TW (char(128) var, ptr, bit(1), entry(char*) var) var, bin, bin, char(128) var, bit(1), char(32))
   [NOT RELEASED]
   call CP$TW (wildpath, a_optp, exp_wildcards, executer, status,
   a_level, a_tame_path, tame_wild, a_tree_wildcard);
   Perform command language Treewalk iteration.

CP$WLD (char(128), var, ptr, entry, fixed) [NOT RELEASED]
   call CP$WLD (wildcard_path, options_ptr, executer, status);
   Invoke executer for every file in wildcard_path.

CP$S (fixed bin(15), fixed bin(15)) [NOT RELEASED]
   call CP$S(user, code);
   Invoke cross process signal on-unit set up by another user process.

CP$SCN (fixed bin(15)) [NOT RELEASED]
   call CP$SCN(key);
   Enable or disable registered cross process signal condition.
CPS$NA (char(\*) var) [NOT RELEASED]
call cps$cn(name);
Retrieve name of the on-unit currently registered for cross process signalling.

CPS$RC (fixed bin(15), fixed bin(15)) [NOT RELEASED]
call cps$ (user, code);
Check on receipt of a cross process signal by another user.

CPS$RG (char(32) var, (\*) bin, bin, bin) [NOT RELEASED]
call cps$rg(name, acl, len, key);
Register process with the cross process signalling mechanism.

CPS$SN ((128) bin, bin, bin) [NOT RELEASED]
call cps$sn(u81, len, rtrnlen);
Gets list of users who have signalled during disabled CPS state.

CPS$ST (fixed bin(15)) [NOT RELEASED]
call cps$st (code);
Check CPS status of a process.

CPUID$ (ptr, bin) III-2-5
call cpuid$ (cpuid$ _struc_ptr, ercode);
Return the CPU Id and microcode revision numbers.

CRAWL_ (entry, entry, ptr, ptr, fixed bin, fixed bin) [NOT RELEASED]
call crawl_ (crawl_fim, crawl_fim_backup, crawl_frame, 
regs_frame, cs_depth, defer_crawl);
Crawls from an inner ring and rejoin sign$ or fim_.

CREA$$ (char(32), bin, char(6), char(6), bin) (svc = 1501) II-A-5
call crea$$ (file_name, fiel_name_len, owner_pw, non_owner_pw, 
code)
Create sub-UFD of same type as containing UFD (ACL or non-ACL).

CREPW$ (char(32), bin, char(6), char(6), bin) II-A-7
call crepw$ (new_dir_name, dir_name_len, owner_pw, non_owner_pw, 
code)
Create a password dir.

CSTAK$ (fixed bin, 1 ..., bit(1) aligned, ptr) [NOT RELEASED]
call cstak$ (depth, cs_data, eog, pb_value);
Manipulate/examine the calling process' concealed stack.

CUCPY$ (bin) [NOT RELEASED]
call cucpy$ (ldrev);
Perform a catch up copy on a mirrored pair of disks.

CV$DQS (bin(31), bin(31)) III-6-12
call cv$dqs (binary_date, quadseconds);
Convert binary date to quadseconds.

CV$DTB (char(128) var, bin(31), bin) III-6-13
call cv$dtb (ascii_date, binary_date, code);
Convert formatted date to binary.

CV$FDA (bin(31), bin, char(21)) III-6-15
call cv$fda (binary_date, day_of_week, ascii_date)
Convert binary date to ISO format.
CV$FDV (bin(31), bin, char(28)var) III-6-17
  call cv$fdv(binary_date, day_of_week, ascii_date)
  Convert binary date to visual format.

CV$QSD (bin(31), 1, 2, 3 bit(7), 3 bit(4), 3 bit(5), 2 bin) III-6-19
  call cv$qsd(quad_secs, fs_date);
  Convert quadseconds since January 1, 1901 to date.

DATES$ returns(bin(31)) III-2-8
  binary_date = date$();
  Return current date/time in binary format.

DATE_A (char(1024) var, bin, char(32) var, ptr, char(1024) var, bin) [NOT RELEASED]
  call date_af (arguments, code, af_name, vcb_ptr, answer, result_max);
  CPL date function.

DB$MOD (bit(1) aligned, ptr) [NOT RELEASED]
  call db$mod (dbg_in_use, dbg_onunit_ptr);
  Set/reset debugger-mode switch and static on-unit.

DB$BR (bin, bit(1)) [NOT RELEASED]
  call db$br(fault_fr_hdr, do_signal);
  Notify the ring zero debugger in the event of a breakpoint.

DELAY (bin, bin, bin) [NOT RELEASED]
  call delay (min, max, margin)
  Define delay times for printing characters after new line.

DELAY_ (char(*) var, fixed, char(*) var) [NOT RELEASED]
  call delay_ (com_args, com_status, com_name);
  Processes command arguments for DELAY command.

DET$GET (char(128)var, bin, char(32)var, char(1024)var, bin, bin) [NOT RELEASED]
  call det$get(et_path, error_code, error_name, message, msg_size, code);
  Get msg from a Diagnostic Error Table.

DH3270 [NOT RELEASED]
  call dh3270;
  Initiate the data handler for IBM 3270 terminals (DPTX).

DIR$CR (char (128) var, ptr, bin) II-4-15
  call dir$cr(dir_path, attribute_ptr, code);
  Create a directory.

DIR$CRC0 (char(32)var, ptr, bin) [NOT RELEASED]
  call dir$cro(dir_name, attribute_ptr, code);
  Create a directory.

DIR$LS (bin, bin, bit(1), bit(4), ptr, bin, ptr, bin, bin, bin, (4)bin, (31), bin(31), bin) II-4-17
  call dir$ls(dir_unit, dir_type, initialize, rtn_file_types, addr(wild_card_array), wca_len, addr(rtn_struct),
  max_entries, entry_size, rtn_num_entries, num_types, before_binary_date, after_binary_date, code)
  Search directory.

6-12  Prime Restricted
DIR$RD (bin, bin, ptr, bin, bin) II-4-23
    call dir$rd(key, dir unit, addr(buffer), buf len, code)
    Read dir entries. Key = K$INIT, K$READ.

DIR$SE (bin, 1, 2 bit(13), 2 bit(2), 2 bit(1), bit(1), ptr, bin, bin, bin, bin, bin) II-4-27
    call dir$se(dunit, dtype, re wind, sel_ptr, arg_outptr, outnum,
          out len, out_filled_in, totals, max_type, code);
    Retrieve info about selected entries in a given directory.

DIRSER (bin, bin, bit(1), ptr, bin, ptr, bin, ptr, bin, bin, bin, bin) [NOT RELEASED]
    call dirser(dunit, dtype, re wind, sel_ptr, sel len _rem,
          wild_ptr _rem, wild_len _rem, arg_outptr, outnum, out len,
          out_filled_in, totals, max_type, code);
    Remote interlude to DIR$SE (NPX only).

DGEO (bin, ptr, bin) [returns(bin)] IV-5-18
    ldev = dgeo$ (pdev, geo_ptr, code);
    Register disk geometry with the disk driver.

DLSCMCGC (1, 2 bin, 2 bin, 2 bin, 1, ..., bin) [NOT RELEASED]
    call dl$cmgc(search_list, status, error_code);
    Gate to get CCPAT entry for a specified controller.

DLSCMGC (1, 2 bin, 2 bin, 2 bin, 2 bin, bin) [NOT RELEASED]
    call dl$cmgc(search_list, cntl_index, error_code);
    Get controller index for a specified LAN or ICS controller.

DLSCMGLS (1, 2 bin, 2 bin, 1, ..., bin) [NOT RELEASED]
    call dl$cmgls(search_list, status, error_code);
    Get poc_load_Parms state for specified controller index.

DLSCMGRE (1, 2 bin, 2 bin, 1, 2 bin, 2 (7)bin, 2 (16)bin, bin) [NOT RELEASED]
    call dl$cmgre(search_list, status, error_code);
    Get a PCCRSTDT entry for a specific ICS controller.

DLSCMUC (1, ..., bin) [NOT RELEASED]
    call dl$cmuc(update_list, error_code);
    Update CCPAT entry for a specified controller index.

DLSCMUL (1, ..., bin) [NOT RELEASED]
    call dl$cmul(update_list, error_code);
    Update PCC_LOAD_PARMS state for any controller.

DLSCMURE (1, 2 bin, 2 bin, 2 bin, 2 (7)bin, 2 (16)bit(16), bin) [NOT RELEASED]
    call dl$cmure(update_list, error_code);
    Update a PCCRSTDT entry for a specific ICS controller.

DLSCIAIO (1, 2 bin, 2 (*), 2 bin, 2 bin, 2 bin, 2 (31), 2 bin, 2 bin) [NOT RELEASED]
    call dl$ciaio(alloclist, error);
    Allocating SEG0 area and phantom interrupt code.

DLSCASY (1, 2 bin, 2 (*), 1 (*), 2 bin, 2 bin, 2 bin) [NOT RELEASED]
    call dl$csay(start_list, error);
    Start ASYNC support for an ICS controller.

DLSCCIDIO (1, 2 bin, 2 (*), 2 bin) [NOT RELEASED]
    call dl$ccidio(deallocate_list, error);
    Deallocating SEG0 area and phantom interrupt code.
DL$CDLL (1, 2 bin, 2 (*)bin, 2(*))bin, 2 char(128)var, 1(,), 2 bin, 2 bin, 2 bin) [NOT RELEASED]
call dl$cdll(load_list, error);
Down line load a DMT file into specified ICS controllers.

DL$CISIV (1, 2 bin, 2 (*)bin, 1(,), 2 bin, 2 (*)bin, 1 (,), 2 bin, 2 bin, 2 bin) [NOT RELEASED]
call dl$cisiv(verify_list, status, error);
Issue self verify to a specified ICS controllers.

DL$CNCR (1, 2 bin, 2 (*)bin, ()bin) [NOT RELEASED]
call dl$cncr(reset_list, error_code);
Perform normal reset for ICS controllers.

DL$CSCR (1, 2 bin, 2 (*)bin, ()bin) [NOT RELEASED]
call dl$cscr(reset_list, error_code);
Perform special reset for ICS controllers.

DL$CSDC (1, 2 bin, 2 (*)bin, 1 (), 2 bin, 2 bin, 2 bin) [NOT RELEASED]
call dl$csdc(shut_list, error);
Shut down a specified ICS controller.

DL$CSR (1, 2 bin, 2 (*)bin, 1 (), 2 bin, 2 bin, 2 bin) [NOT RELEASED]
call dl$csrt(start_list, error);
Starting IPQNM for given ICS controllers.

DL$CHD (1, 2 bin, 2 (*)bin, char(128)var) [NOT RELEASED]
call dl$hd(load_list, error);
Initiate downline load of an LHC controller.

DL$HLISV (1, 2 bin, 2 (*)bin, 1 (,), 2 bit(16), 2 bit(16), 1 (,), 2 bin, 2 bin) [NOT RELEASED]
call dl$hisv(verify_list, status, error);
Initiate self verify of an LHC controller.

DL$HLSP (1, 2 bin, 2 (*)bin, 2 ptr, 2 bin, 1 (,), 2 bin, 2 bin) [NOT RELEASED]
call dl$hls(start_list, error);
Perform load start packet function for the LAN controller.

DL$LHNCR (1, 2 bin, 2 (*)bin, ()bin) [NOT RELEASED]
call dl$lnr(reset_list, error_code);
Normal reset of a LAN controller.

DL$HULD (1, 2 bin, 2 bin, 2 char(128)var, 1, 2 bin, 2 bin) [NOT RELEASED]
call dl$huld(dump_list, error);
Initiate upline dump from a LAN controller.

DMP$LS (bin, bin, bin, bin) [NOT RELEASED]
call dmp$ls(index, low_seg, high_seg, err_code);
Display entries from the DMP_SEG array for partial tape dump.

DMP$LU (bin, char(32), bin) [NOT RELEASED]
call dmp$lu(index, user_name, err_code);
Display entries from the DMP_USR array for partial tape dump.

DMP$RS (bin) [NOT RELEASED]
call dmp$rs(err_code);
Reset the DMP_SEG and DMP_USR arrays to the default values.

DMP$SG (bin, bin, bin) [NOT RELEASED]
call dmp$sg(low_seg, high_seg, err_code);
  Add new entries to DMP_SEG array for partial tape dump.

DMP$US (char(32), bin) [NOT RELEASED]
call dmp$us(user_name, err_code);
  Routine to add new entries to DMP_USR array for partial tape dump.

DOSSUB (char(80), bin) [NOT RELEASED]
call dossub(command_line, code);
  Old internal command processor.

DPT$QM (1, 2 (8)bin, 2 (8)bin, 2 (8)bin, 2 (32)bin, 2 (32)bin, 2 (32)bin, 2 (32)bin, 2 bin,
  2 bin, 2 bin, bin) [NOT RELEASED]
call dpt$qm(queue_length, code);
  Queue monitor subroutine for DPTX queues.

DPT$ST (bin, bin, (*, 19)bin, bin) [NOT RELEASED]
call dpt$st(key, line, info, code);
  Retrieve ring 0 information for DPTX monitor.

DPTINI (bin, bin) [NOT RELEASED]
call dptini (file_unit, code);
  Initialize all of the DPTX databases.

DPTOFF (bin) [NOT RELEASED]
call dptoff (code);
  Deallocates all of the DPTX databases and shuts down the DPTX phantoms.

DS$ACC (char(32)var, ptr, bin) [NOT RELEASED]
call ds$acc(node_name, sptr, code);
  Return Primenet nodal access configuration.

DS$ASY (bin, bin, ptr, bin) [NOT RELEASED]
call ds$asy(key, line_no, sptr, code);
  Retrieve asynchronous line information.

DS$AVL (ptr, bin, bin) [NOT RELEASED]
call ds$avl(list_p, ldav, code);
  Return disk size and date of last backup.

DS$CFG (ptr, bin(31), bin) [NOT RELEASED]
call ds$cfg(loc_ptr, storage_size, code);
  Return config directive values.

DS$COM (ptr, bin) [NOT RELEASED]
call ds$com(bufptr, code);
  List communications controller status.

DS$ENV (bin, ptr, bin) [NOT RELEASED]
call ds$env(user_no, lptr, code);
  Return general information about a user's process environment.

DS$HST (bin, char(32)var, pointer, bin) [NOT RELEASED]
call ds$hst(version_num, lan_name, user_bufr_p, error_code);
  Retrieve configured HOST information from the NSS database.

DS$LANT (pointer, bin) [NOT RELEASED]
call ds$lan(user_bufr_p, error_code);
            Retrieve configured LAN information from the NSS database.

DS$LTS (bin, char(32)var, pointer, bin) [NOT RELEASED]
          call ds$lts(version_num, lan_name, user_bufr_p, error_code);
          Retrieve LTS information from the NSS database.

DS$LTU (bin, char(32)var, char(16)var, bin, bin) [NOT RELEASED]
          call ds$ltau(user_num, LAN_name, LTS_name, LTS_line, error_code);
          Retrieve LTS User Information from Primos User Number.

DS$PNC (bin, ptr, bin) [NOT RELEASED]
          call ds$pnc(pnc_id, sptr, code);
          Return the IDs of all configured nodes on a specified ring.

DS$POR (ptr, bin(31), bin) [NOT RELEASED]
          call ds$por(sptr, size, code);
          Return the system port assignments.

DS$RECK (1, ..., bin) [NOT RELEASED]
          call ds$rechk(resus_switch_data, return_code);
          Interrogate the REMote System USe r switch.

DS$REENA (1, ..., bin) [NOT RELEASED]
          call ds$reena(r3, return_code);
          Enable REMote System USe r switch.

DS$RERD (char(1), bin) [NOT RELEASED]
          call ds$rerd(in_char, return_code);
          Read character from User-1 output queue.

DS$RERST (bin) [NOT RELEASED]
          call ds$rerst(return_code);
          Reset REMote System USe r switch.

DS$RESSW (char(32), bin, char(32), bin, bin) [NOT RELEASED]
          call ds$ressw(user_id, user_no, user_node, synchroniser_id,
                         return_code);
          Set REMote System USe r switch.

DS$REWR (char(1), bin) [NOT RELEASED]
          call ds$rewr(out_char, return_code);
          Put character on REMote System USe r input queue.

DS$RNG (bin, bin, ptr, bin) [NOT RELEASED]
          call ds$rng(pnc_id, ring_id, sptr, code);
          Return the status of a specified ring node.

DS$SYN (ptr, bin) [NOT RELEASED]
          call ds$syn(sptr, code);
          Return synchronous line configuration.

DS$UNI (bin, bin, bin, char(128)var, ptr, bin) [NOT RELEASED]
          call ds$uni(key, user_no, unit_no, full_path, struc_ptr, code);
          Returns data on specified user's open file unit, attach point or open file.

DUPLX$ (bin) returns(bin) (svc = 0705) III-3-57
prev_config = duplx$(new_term_config)
Set terminal configuration word (bits 1-8 only).

DY$SGS returns(bin) III-4-25
maximum_private_dynamic_segs = dy$sgs();
Returns maximum number of dynamic segments for this user.

EBK$$ (bin, bin(31), bin(31), bin) [NOT RELEASED]
call ebk$$ (unit, leof, peof, code);
Returns physical and logical eof for file open in block mode.

EF$RELOCATE (ptr, ptr) returns(ptr) [NOT RELEASED]
real_address = ef$relocate(smt_ptr, erp);
Relocate an ERP for an EPF.

EM3270 (bin, (1920)bin) [NOT RELEASED]
call em3270 (line_number, virtual_buffer_temporary);
Initiate the emulator handler for the IBM 3270 terminals (DPTX).

ENC$RP, ENCRYPT$(char(16), char(16) var) III-6-24
call encrypt$ (encrypted_password, unencrypted_password);
Encrypts login passwords (one-way).

ENT$RD (bin, char(32)var, ptr, bin, bin) II-4-35
call ent$rd(dir_unit, entry_name, addr(rtn_struct), rs_len, code);
Read a dir entry by a given name.

ENTDIR$(char(128)var, char(128)var, char(32)var, bin) returns(bin(1)) [NOT RELEASED]
attached = entdir$(pathname, pathname, entry_name, code);
Attach to parent of object in pathname, return entry name of object.

EPF$AL, EPF$ALLC(ptr, fixed bin) II-5-3
call epf$allc(smt_ptr, status);
Allocate EPF linkage.

EPF$CHAIN (ptr, ptr, bin, bin, struc, struc) [NOT RELEASED]
call epf$chain(smt_ptr, data_ptr, data_len, code, com_state,
com_proc_flags);
Support routine to allow chaining.

EPF$CINF (char(128)var, (8)char(32) var, ptr, bin, ptr, bin) [NOT RELEASED]
call epf$cinf(epf_treename, alias_list, smt_ptr,
epf_database, erroraliases_ptr, status);
Copy EPF information to the registered EPFs database.

EPF$CP, EPF$CPF (ptr, struc, bin, bin) II-5-5
call epf$cpf (smt_ptr, com_state_structure, status);
Get command processor flags from an EPF.

EPF$DBG (pointer, bin, pointer, bin) [NOT RELEASED]
call epf$dbg(smt_ptr, requested_version, epf_dbg_info_ptr,
status);
Obtain EPF information from the PRIMOS command environment for DBG.

EPF$DEL, EPF$DL (ptr, bin) II-5-7
call epf$del(smt_ptr, status);
Terminate EPF, de-allocating storage.
EPFSGETI (char(32)var, bin, ptr, char(32)var, bin) [NOT RELEASED]
  call epf$sgeti(epf_name, epf_database, smt_ptr, bad_lib, code);
  Get information (in an SMT) about a registered EPF.
EPFSGTLI (char(32)var, bin, bin, ptr, ptr, char(32)var, bin) [NOT RELEASED]
  call epf$sgtli(epf_name, epf_database, num_smts,
                 current_SMTs_ptr, smt_ptr, bad_lib, code);
  Get limited information about a registered EPF.
EPFSINFO (ptr, struc, bin) [NOT RELEASED]
  call epf$info(smt_ptr, epf_info, status);
  Return info about a desired epf file.
EPFSINIT (bin, ptr, bin) II-5-9
  call epf$init (key, smtp, status) options(nocopy);
  Initialize EPF static data.
EPFSINVK (ptr, bin[, char(*) var, bin, 1, ..., ptr]) II-5-11
  call epf$invk(smt_ptr, status, com_args, com_status, com_state,
                flags, rtn_function_ptr);
  Start execution of an EPF.
EPFSLENT (char(32)var, char(32)var, ptr, bin) [NOT RELEASED]
  call epf$lent(entryname, libname, liberp, code);
  Search registered EPF libraries in order for specific entrypoint.
EPFSMAP, EPFSMP (bin, bin, bin, bin) returns(ptr) II-5-15
  smtp_ptr = epf$map (key, vmfa_unit, access_rights, status);
  Map an EPF file to virtual memory. Key = K$COPY, K$DBG.
EPFSNF. See EPFSINFO.
EPFSNT. See EPFSINIT.
EPFSREG (1, 2 char(128)var, 2 (8)char(32)var, 2 char(1024)var, 2 bit(1), ptr, bin) [NOT
RELEASED]
  call epf$reg (register_info, error_aliases_ptr, status);
  Register an EPF.
EPFSRELCP, EPFSRL (ptr, ptr) returns (ptr) [NOT RELEASED]
  real virtual address = epf$relc(epf_relative_ptr, smt_ptr);
  Relocate EPF relative pointer(ERP).
EPFSRN, EPFSRUN(bin, bin, bin[, char(*) var, bin, struc, struc, ptr]) II-5-19
  smtp = epf$run (key, src_unit, status [, , com_args, com_status,
                         com_state, flags, rtn_function_ptr]);
  Run an EPF.
EPFSSMAL [NOT RELEASED]
  call epf$smal;
  Permit linking to in-use static mode library.
EPFSSMDL [NOT RELEASED]
  call epf$smdl;
  Disallow linking to in-use static mode library.
EPFSRCH (ptr, char(32)var, ptr) [NOT RELEASED]
    call epfsrch(epf_smt_ptr, faulted_entryname, lib_entry_erp);
    Search an EPF library to resolve a faulted entrypoint.

EPFSREG (bin, char(128)var, bin, bin, bin) [NOT RELEASED]
    Call epf$reg(remove_key, epf_pathname, epf_smt, first_proc_seg,
               epf_database, status);
    Un-register EPF.

EPFSVK. See EPFSINVK.

EPF_ERR (fixed bin(15), char(1024) var) [NOT RELEASED]
    call epf_err(err_code, info_str);
    Print diagnostic error message on terminal.

EPF_RL (ptr) [NOT RELEASED]
    call epf_rl(epf_smt_ptr);
    Pop volatile EPF smt data for program and library EPFs.

EQUALS (char(32)var, char(32)var, char(32)var, bin) II-4-37
    call equal$(obj_name, pattern, generated, code);
    Generate (equal) name from a source name and a pattern.

ERKLS$ (bin, char(1), char(1), code) (svc = 1524) II-3-60
    call erkl$(key, erase_char, kill_char, code)
    Read or set erase and kill chars. Chars are right justified, zero filled. Key = K$WRIT, K$READ.

ERRPR$ (bin, bin, char(*), bin, char(*), bin) (svc = 1402) III-3-30
    call errpr$(key, error_code, message, message_len, file_name,
               file_name_len)
    Interpret a return code. Key = K$NRTN, K$SRTN, K$IRTN.

ERTXT$ (bin, char(1024)var) III-2-9
    call ertxt$(error_code, error_text);
    Return the text of a specified error code.

EVAL_A (char(*), var, bit(1) aligned, ptr, ptr, fixed bin, fixed bin, fixed bin) [NOT RELEASED]
    call eval_a (expression, op_switch, local_var_ptr,
               global_var_ptr, expr_size, error_code, com_status)
    Evaluate all CPL vars in a character string.

EX$CLR III-7-35
    call ex$clr;
    Disable signalling of the EXITS condition upon program termination.

EX$RD (bin) III-7-36
    call ex$rd (transmit_exit_setting);
    Return value of the TRANSMIT_EXIT counter.

EX$SET III-7-37
    call ex$set;
    Enable signalling of EXIT$ on program termination.

EXIT (svc = 0105) III-5-7
    call exit
    Return to PRIMOS.
EXTRA$A (char('*') var, char('*') var, bin, char(32) var, bin) II-4-39
   call extr$A (full_path, parent_path, max_length, entryname, code);
   Return parent tree and entryname from treename.

EXTRAC (bin, pointer, bin, char('*') varying, bin) [NOT RELEASED]
   call extrac(caller_key, xp, xtype, xstr, xarglen);
   Extract a spare data field from a string.

FATAL$ (bin) [NOT RELEASED]
   call fatal$(code);
   Fatal error handler.

FILE$D0 (char(32)var, bin) [NOT RELEASED]
   call file$d0 (obj_name, code);
   Delete a file or directory.

FILE$DL (char(128)var, bin) II-4-41
   call file$dl (object_pathname, code);
   Delete a file.

FIND$BKT (ptr, char(32) var, bin) returns (ptr) [NOT RELEASED]
   data_address = find$bk t (table_address, name, code);
   Search a standard hash table for a bucket address.

FIND_U, FIND_UID (char(32) var, bin, ptr, ptr, bin(31), bin) returns (bin(1)) [NOT RELEASED]
   id_found = find_uid (user_id, vf_unit, addr(vf_header),
                      addr(uvf_entry), entry_pos, code);
   Search system validation files for an entry.

INFO$ (bin, ptr, bin) II-4-43
   call info$ (unit, addr(info_struc), code);
   Return information about specified file unit.

FNCHK$ (bin, char('*') var) returns(bit(1)) II-4-45
   name_ok = fnchk$ (key, file_name)
   Check a filename for valid format. Key = K$UPRC, K$WLC, K$NULL, K$NUM.

FNONU$ (ptr, char(32) var, ptr, ptr, ptr) returns(bit(1)) [NOT RELEASED]
   cond_was_found = fnonu$ (frame_ptr, condition_name,
                           onunit_or_last_ptr, catch_all_ptr, spec_ptr);
   Find on-unit in specified stack frame.

FNSID$ (fixed bin, ptr, fixed bin, fixed bin) [NOT RELEASED]
   call fnsid$ (key, addr(remote_id), max_entries, code);
   Search and add entries to user's remote id database. (NPX) Key = K$ADD, K$LIST, K$SRCH.

FORCEW (bin, bin [], bin]) (svc = 0115) II-4-47
   call forcew (key, file_unit [], code)
   Force write to disk immediately. Key = 0.

FORK$ (char(8), bin) returns(bit(1)) [NOT RELEASED]
   I_am_child = fork$ (unique_id, code);
   Creates a child process from within a program.

FPLEN$ (bin(31)) [NOT RELEASED]
length = fplen$(free_poll_id);
   Return the length of the free pool queue.

FRES$RA (ptr) III-4-23
   call fres$ra (rtn_function_ptr);
   De-allocate space used for return info from command functions.

FRK$CP returns(bit(1)) [NOT RELEASED]
   foo = frk$cp;
   Address copy routine for Forked processes.

G$METR (bin, ptr, bin, bin, bin, bin) [NOT RELEASED]
   call g$metr(key, bufptr_arg, buf_size, user_arg, revision,
                code);
   Get metering data of various sorts and flavors. See also: GMETR$.

GEM$PB (bin, bin, bin, bin, bin, bin, bin, bin, bin, bin, bin) [NOT RELEASED]
   call gem$pb(seg_code, eventid, nwords, len1, arg1,
              [len2, arg2, ..., len6, arg6]);
   Probe to monitor ring3 activities.

GEM$R3 returns(bit(1)) [NOT RELEASED]
   monitoring_enabled = gem$r3();
   Indicates whether ring 3 monitoring is enabled.

GEM$ST (bin, pointer, bin) [NOT RELEASED]
   call gem$st(assign_buffer, addr(init structure), code);
   Control procedure for General Event Monitor (GEM).

GEM$WT (bin, pointer) [NOT RELEASED]
   call gem$wt(lost_count, buffer_pointer);
   Gate routine to wait for and dump General Event Monitor buffers.

GET$DPT (bin) [NOT RELEASED]
   call get$dpt(program_session_depth);
   Get the depth of the program_session.

GET$DTR3 (bin, bin(31), bin) returns(ptr [NOT RELEASED]
   block_pointer = get$dtr3(storage_type, block_size, code);
   Allocates given amount of storage in DTAR 3 according to storage type.

GETAT$ (1, 2 bin, 2 bin, 2 bin, 2 bin, 2 bin, 2 bin) [NOT RELEASED]
   call getat$(system_defaults)
   Reads system defaults and passes them to Edit_Profile.

GETID$ (ptr, bin, bin) II-2-21
   call getid$(addr(id_struc), max_groups, code);
   Get full user id.

GETREG ((*)bin) [NOT RELEASED]
   call getreg(svec)
   Sets tvec from svec.

GETSN$ (bin, bin, bin, (*), bin, bin, bin) [NOT RELEASED]
   call getsn$(key, start_segno, num_segs, segno_array,
            num_segs_found, code);
   Allocates a set of dynamic segments. Key = K$UP, K$DOWN, K$UPC, K$DWNC.
GET_REPL (bit(1)) returns(bit(2)) NOT RELEASED

reply = get_reply(all_options);
    Fetch a yes/no/null/next reply from command input stream.

GINFO ((6)bin, bin) (svc = 0112) II-2-10

call ginfo(xer_vec, xer_vec_len)
    Return operating system info (PRIMOS II).

GMETRS$ (bin, ptr, bin, bin) NOT RELEASED

call gmets$(key, addr(buffer), buf_size, code, user_number)
    Get metering data of various sorts and flavors. Key = GM_SYS, GM_FS, GM_INT,
    GM_USER, GM_MEM, GM_DISK. See also: G$METR.

GNUSR$ (bin) NOT RELEASED

call gnusr$(network_user_number);
    Gets the network process's user number.

GPASS$ (char(32), bin, char(6), char(6), bin) (svc = 1504) II-2-23

call gpass$(ufd_name, ufd_name_len, owner_pw, non_owner_pw,
    code);
    Return passwords of sub-UFD.

GPATHS$ (bin, bin, char(128), bin, bin, bin) II-4-49

call gpath$(key, file_unit, path_name, path_name_len,
    rtn_path_len, code)
    Find pathname for file unit or current home or attach point. Key = K$UNIT, K$CURA,
    K$HOMA, K$INIA.

GT$PAR (bit(16), char(*)var, char(*)var, char(*)var, char(*)var, char(*)var, bin, struc, bin) III-6-27

call gt$par(key, white, quote, break, source_str, token_str,
    token_str_size, info, next_char);
    Parse a character string into tokens separated by white space, quotes, and break characters.

GT$HRS$ (bin(31), bin, ptr, ptr, bin) NOT RELEASED

call gt$hrs$(unique_seg_id, req_acceses, seg_to_share,
    dtar2_seg_ptr, code);
    Map a DTAR2 segment onto a DTAR0 segment.

GV$GET (char(32)var, char(*)var, bin, bin) II-6-12

call gv$get(gvar_name, gvar_value, gvar_value_len, code);
    Retrieve value of a global variable.

GV$SET (char(32)var, char(*)var, bin, bin) II-6-14

call gv$set(gvar_name, gvar_value, bin);
    Set the value of a global variable.

HASH_U, HASH_UID (char(32) var, bin) returns (fixed bin) NOT RELEASED

table_index = hash_uid (user_id, table_size);
    Performs the current hashing function on the passed user ID.

HS$DRAIN NOT RELEASED

call hs$drain;
    Drain the caller's per-user semaphore.

HS$NTFY (bin, bin) NOT RELEASED

call hs$ntfy(user_number, code);
    Notify the specified user's per-user semaphore.
HS$WAIT returns(bin) [NOT RELEASED]  
    notified = hs$wait();  
    Wait on the caller's per-user semaphore.

I$GCLB (ptr, ptr) [NOT RELEASED]  
    call igclb(callers_sb, callers_lb);  
    Get EXIT_LB and EXIT_SB from OLDATA. (INFORMATION only).

I$ON Nonstandard. [NOT RELEASED]  
    FAR0 = addr(condition name (char(*)var))  
    FAR1 = addr(on-unit_ecb)  
    GR2 = Specifier ptr (0 => null())  
    GR5H = 1 if snap option is on, else 0.  
    JSXB I$ON  
    Make PL/I on-unit.

ICE$ III-5-8  
    call ice$;  
    Initialize command environment.

ICMTB_  
    Internal command table. Not a procedure. [NOT RELEASED]

ICPL_ (char(*) var, char(*) var, bin, bin, 1, 2 bit(1), 2 bit(1), 2 bit(14), ptr) [NOT RELEASED]  
    call icpl_(arg_source, args, com_status, src_unit,  
        flags, rtn_function_ptr);  
    Invoke CPL interpreter on given file, processing suffix.

ICS2CT (bin, bin, bin)returns(bin) [NOT RELEASED]  
    success = ics2ct(key, device_address, data);  
    Allow OTA and INA from eagle monitor to controller. Key = 1 (INA), 2 (OTA).

IDCHKS$ (bin, char(*)var) returns(bit(1)) III-2-22  
    id_ok = idchks$(key, id);  
    Check an id for valid format. Key = K$UPRC, K$WLDC, K$NULL, K$GRP.

IG$ABUF (ptr, ptr, bin) [NOT RELEASED]  
    call ig$abuf(lcptr, xcb, status);  
    Add a buffer to RTNQ. Simulates buffers being returned by controller.

IG$AWIR (ptr, bin) [NOT RELEASED]  
    call ig$awir(address, status);  
    Ring3 gate to wire a page.

IG$COLD (bin, bin, bin, bin) [NOT RELEASED]  
    call ig$cold(device, num_connections, num_windows, status);  
    Initialize database for a controller. If first time called for any controller, initialize over-all  
    IGUANA database.

IG$DEQ (bin(31), ptr, bin, bin) [NOT RELEASED]  
    call ig$deq(lcid, buffer, size, status);  
    Dequeue an item (command or XCB) from INQ.

IG$ENQ (bin(31), ptr, bin, bin) [NOT RELEASED]  
    call ig$enq(lcid, buffer, size, status);  
    Enqueue an item (command or XCB) on an OUTQ.
IG$FIND (bin, bin, bin, bin, bin(31), bin(31), bin(31), bin) [NOT RELEASED]
call ig$find(device, lcn, q_array, buf_size, i_sem, o_sem, lc_id, status);
Find a specific per-connection database for a particular controller

IG$GBUF (bin(31), ptr, bin, bin) [NOT RELEASED]
call ig$gbuf(lcid, buffer, size, status);
Obtain a buffer from RTNQ.

IG$RBUF (bin(31), ptr, bin) [NOT RELEASED]
call ig$rbuf(lcid, buffer, status);
Add a buffer to BUFQ. Becomes available for controller to write input data and put on INQ.

IG$RMV (bin(31), bit(16), ptr, bin, bin) [NOT RELEASED]
call ig$rmv(lcid, bit_mask, buffer, size, status);
Routines to remove buffers from all queues of a connection.

IG$SWIR (ptr, bin) [NOT RELEASED]
call ig$swir(address, status);
Ring3 gate to unwire a page.

IG$WAIT (bin(31), bin) [NOT RELEASED]
call ig$wait(lcid, sem);
Ring3 gate to wait on a semaphore (input done, output done).

IN$LO returns(bit(1)) III-2-23
in_grace_period = in$lo();
Return state of PPMD.IN_GRACE_PERIOD (i.e., force logout in progress).

INIT$3 (bin, bin, char (') var, bin) [NOT RELEASED]
call init$3(key, user_num, login_comline, cpl_unit);
Initialize ring 3 environment.

INSON$ (bin) [NOT RELEASED]
call inson$(key)
Initialize static on units. Key = 0 (ring 0), 3 (ring 3), 2 (both).

INTCM_ (char(32) var, entry, bit(3), bin, bit(1), bit(5)) returns (bit(1)) [NOT RELEASED]
Is_internal = intcm_ (command_name, entry_var, exp_wildcards,
eq_position, vfy_default, default_types);
Fetch local command table entry if any, else check system's table.

IO$GET_MSG (1, 2 bin, 2 bin, 1, 2 bin, 2 ..., 1, ..., bin) [NOT RELEASED]
call io$get_msg(wait_info, msg_sender, message, status);
Return a stored I/O related message for DSM/SM to log.

IO$PUT_MSG (1, 2 bin, 2 bin, 1, ..., bin) [NOT RELEASED]
call io$put_msg(wait_info, message, status);
Put a I/O related message into the queue of message for DSM/SM to log.

IOA$ (char(') bin, [arg1, ... arg99]) III-3-32
   call ioa$(control_string, control_string_len
   [, arg1, ... , arg99]);
Write formatted string to terminal. See D for control string format.

IOA$ER (char(') bin, [arg1, ... arg99]) III-3-38
call ioa$er(control_string, control_string_len [, arg1, ..., arg99]);
   Write formatted string to terminal after forcing on terminal output. See D for control string format.

IOAS$ (char(*), bin, bin, char(*), bin,[arg1, ..., arg99]) III-6-32
   call ioa$s(rtn_string, rtn_str_size, rtn_str_rtned_len, 
   control_string, control_string_len [, arg1, ..., arg99]);
   Return formatted string according to control string. See D for control string format.

IOAF$ ((101)ptr (long), char(*), bin, bin) [NOT RELEASED]
   call ioa$f$ (arg: pointers, buffer, buffer_max_size, rtn_len);
   Process control format string. (IOA$)

IPC$C(blin, bin) [NOT RELEASED]
   call ipc$0 (mbx_id, code);
   Close a IPC mailbox using the mbx_id specified.

IPC$CA [NOT RELEASED]
   call ipc$ca;
   Close all mailboxes the current user owns.

IPC$CM (bin, bin, bin) [NOT RELEASED]
   call ipc$cm(mode_key, mbx_id, code);
   Change mailbox access mode from read/write to specified mode.

IPC$GU (bin, bin, ptr, bin, bin, bin) [NOT RELEASED]
   call ipc$gu(key, mbx_id, buf_ptr, buf_size, returned_size, code);
   Get the desired mailbox user ID specified by key. Key = K$READ, K$WRT, K$RDWR, 
   K$MINE.

IPC$NC (bin, bin) [NOT RELEASED]
   call ipc$nc (mbx_id, code);
   Close a IPC mailbox with notification using the mbx_id specified.

IPC$O(bin, bin, char(128) var, bin, bin) [NOT RELEASED]
   call ipc$o(access_key, notification_key, pathname, mbx_id, code);
   Open an IPC mailbox for specified access using pathname. Access_key = K$READ, 
   K$WRT, K$RDWR. Notification_key = K$NFIN, K$NFSN.

IPC$O0 (bin, bin, char(*)var, bin, [1, 2 char(6), 2 bin, char(*)var], bin) [NOT RELEASED]
   call ipc$o0(access_key, notification_key, entry_or_pathname, 
   mbx_id, [usrid, my_node], code);
   Open an IPC mailbox for specified access using entryname for access control. Access_key 
   = {k$read, k$writ, k$rdwr}; notification_key = {k$nfin, k$nfsm}.

IPC$R (bin, bin, ptr, bin, bin, bin) [NOT RELEASED]
   call ipc$r(read_key, mbx_id, buf_ptr, buf_size, msg_size, 
   mbx_send_uid, code);
   Receive a message from specified IPC mailbox waiting if specified. Read_key = K$READ, 
   K$RDWT.

IPC$RA (bin, bin, ptr, bin, bin, bin, bin, bin) [NOT RELEASED]
   call ipc$ra(read_key, buf_ptr, buf_size, mbx_id, msg_size, 
   mbx_send_uid, code);
   Receive a message from any IPC mailbox owned by the user. Read_key = K$READ, 
   K$RDWT.
IPC$SA (bin, ptr, bin, bin) [NOT RELEASED]
call ipc$sa(mbx_id, msg_ptr, msg_size, code);
  Send a message to any IPC user attach to specified mailbox.

IPC$SB (bin, ptr, bin, bin) [NOT RELEASED]
call ipc$sb(mbx_id, msg_ptr, msg_size, code);
  Send a message to all IPC users attach to specified mailbox.

IPC$SS (bin, bin, ptr, bin, bin) [NOT RELEASED]
call ipc$ss(mbx_id, mbx_uid, msg_ptr, msg_size, code);
  Send a message to any IPC user attach to specified mailbox.

IPC$SSA (bit[1], bin, ptr, bin, bin) [NOT RELEASED]
call ipc$ssa(mbx_id, msg_ptr, msg_size, code);
  Send a message to a specific IPC user.

IPC$SSB (bit[1], bin, ptr, bin, bin) [NOT RELEASED]
call ipc$ssb(mbx_id, msg_ptr, msg_size, code);
  Send a message to all IPC users attach to specified mailbox and notify the caller.

IPC$ST (bin, bin, bin, bin) [NOT RELEASED]
call ipc$st(key, mbx_id, value, code);
  Return various IPC statuses determined by user specified key. Key = K$NMSG, K$MRMOM,
  K$ROOM, K$NUSR, (K$NFYS).

ISSAB (bin, bin, bin) returns(ptr) [NOT RELEASED]
call is$ab(session_id, buffer_length, code)
  AllocateBuffer - allocate an ISC data buffer.

ISSAS (bin, ptr, ptr, ptr, bin) [NOT RELEASED]
call is$as(SessionID, ConnectMessage, ConfigInfo_ptr,
  SessionSyncs, ReturnCode);
  AcceptSession - accept an ISC Session Request.

ISSCE (bin, bin) [NOT RELEASED]
call is$ce(session_id, code);
  ClearException - clear an an outstanding exception

ISSEPFU, ISSEPU (char[128])var, bin, bin) returns(bit(1)) [NOT RELEASED]
ing_use = is$epfu(target_tree, target_type, code);
  Determine if an EPF is in use.

ISSFB (bin, ptr, bin) [NOT RELEASED]
call is$fb(session_id, buffer, code);
  FreeBuffer - free an ISC data buffer.

ISSGE (bin, bin, ptr, bin) [NOT RELEASED]
call is$ge(session_id, exception Raised, message, code);
  GetException - get details of an outstanding exception.

ISSGRQ (ptr, ptr, bin, bin, bin) [NOT RELEASED]
call is$grq(TargetLN, ConnectMessage, ConfigInfo_ptr, SessionID,
  AuthInfo_ptr, ReturnCode);
  GetSessionRequest - get an incoming session request.

ISSGRS(bin, ptr, ptr, bin, bin) [NOT RELEASED]
call is$gra(SessionID, TargetLLN, AuthInfo_ptr, ResponseCode, 
              ConnectMessage, ReturnCode);
             GetSessionResponse - get response to ISC session request.

IS$GSA (bin, ptr, ptr, ptr, bin) [NOT RELEASED]
     call is$gsa(SessionID, ConfigInfo_ptr, SessionSyncs_ptr, 
              AuthInfo_ptr, ReturnCode);
             GetSessionAttributes - provide attributes of a session.

IS$GSO (bin, ptr, bin, bin) [NOT RELEASED]
     call is$gso(ArrayLength, SessionsOwned_ptr, SessionCount 
              ReturnCode);
             GetSessionsOwned - get a list of sessions owned by caller.

IS$GSS (bin, ptr, bin) [NOT RELEASED]
     call is$gss(SessionID, StatusInfo_ptr, ReturnCode);
             GetSessionStatus - provide session status information.

IS$PAS (bin, ptr, ptr, ptr, ptr, bin) [NOT RELEASED]
     call is$pas(SessionID, ConnectMessage, ConfigInfo_ptr, 
            InternalAuthInfo_ptr, SessionSyncs, ReturnCode);
             AcceptSession - accept an ISC Session Request (privileged process).

IS$PRS (ptr, ptr, ptr, ptr, bin, ptr, bin) [NOT RELEASED]
     call is$prs(TargetLLN, ConnectMessage, ConfigInfo_ptr, 
            AuthInfo_ptr, SessionID, SessionSyncs, ReturnCode);
             RequestSession - request an ISC session (privileged process).

IS$PTS (bin, bin, ptr, ptr, bin, bin) [NOT RELEASED]
     call is$pts(SessionId, ReasonCode, Message, 
            InternalAuthInfo_ptr, ReturnCode);
             TerminateSession - terminate an ISC session

IS$R (char(12), bin, bin) [NOT RELEASED]
     call is$r(ServerUID, SessionRequestPending, ReturnCode);
             RegisterProcessAsServer - register as an ISC Server.

IS$RE (bin, bin, bin) [NOT RELEASED]
     call is$re(SessionId, ExceptionRaised, ReturnCode);
            RaiseException - raise an exception on an ISC session.

IS$RM (bin, ptr, bit(1), bin) [NOT RELEASED]
     call is$rm(SessionId, Message, IsExpedited, ReturnCode);
             ReceiveMessage - receive a message on an ISC session.

IS$RS (ptr, ptr, ptr, ptr, bin, ptr) [NOT RELEASED]
     call is$rs(TargetLLN, ConnectMessage, ConfigInfo_ptr, 
            SessionID, SessionSyncs, ReturnCode);
            RequestSession - request an ISC session.

IS$SM (bin, ptr, bit(1), bin) [NOT RELEASED]
     call is$sm(SessionId, Message, IsExpedited, ReturnCode);
            SendMessage - send a message on an ISC session.

IS$STA (bin, ptr, bin) [NOT RELEASED]
     call is$sta(SessionID, StatisticsInfo_ptr, ReturnCode);
             GetSessionStatistics - provide ISC session statistics.
IS$TS (bin, bin, ptr, bin) [NOT RELEASED]
call is$ts(SessionId, ReasonCode, Message, ReturnCode);
      TerminateSession - terminate an ISC session.

IS$U (bin) [NOT RELEASED]
call is$u(ReturnCode);
      UnregisterProcessAsServer - unregister as an ISC Server.

ISACLS (bin, bin) returns(bit(1)) II-2-25
      is_acl_directory = isacl$(file_unit, code);
      Get directory type (ACL or non-ACL).

ISFEPF () returns(bit(1)) [NOT RELEASED]
      parent_is_epf = isepf();
      Determine if parent is an EPF.

ISPRIV$, ISPRV$ (bit(16), bin, char(128)var, char(32)var) returns (bit(1)) [NOT RELEASED]
      user_is_priv = ispriv$( privilege_definition, user_type, operation, ck_group);
      Check user privilege.

ISREM$(bin, char(128)var, bin, char(32)var, bin) returns(bit(1)) II-4-52
      file_is_remote = isrem$(key, filename, unit, system_name, code);
      Return information on remoteness of a filesystem object. Key = k$name, k$unit.

ISUREM$(bin, char(32)var, bin) returns(bit(1)) [NOT RELEASED]
      unit_is_remote = isurem$(unit, sysnam, code);
      Return information on remoteness of a filesystem object open on a unit.

JOB$0 (bin, bin, bin, (entry_length) bin, (entry_length) bin, bin, bin) [NOT RELEASED]
call job$0(key, queue_index, priority, old_entry, new_entry, entry_length, code);
      Operate on batch queue control file in a secure manner. (JOB only)

JOB$1 (bin, ptr, ptr, bin) [NOT RELEASED]
call job$1(key, addr(qinfo), addr(job_entry), code);
      Queue control gate for BATCH subsystem.

KLM$ES (struc, bin) [NOT RELEASED]
call klm$es(klm_struc, code);
      Return serialization information on an EPF.

KLM$MV (ptr, bin) [NOT RELEASED]
call klm$mv(klm_ptr, status);
      Move klm info from invokers buffer into level class storage.

KLM$PR (bin) [NOT RELEASED]
call klm$pr(code);
      Output copyright notice.

KLM$RT (struc, bin) [NOT RELEASED]
call klm$rt(klm_struc, code);
      Return klm information.

KTRAN$ (char(*) var, bin) returns (bin) [NOT RELEASED]
      hash_key = ktran$(name, modulus);
      Provides simple hash on name.
LDISK$ (bin, char(32) var, ptr, bin, bin) II-4-54
    call ldisk$ (key, system_name, addr(disk_list), max_entries, 
        code);
    Return information on the system's disk list. Key = K$ALL, K$LOCL, K$REM, K$SYS.

LDISKUS (fixed bin, (128) bit(1), fixed bin) [NOT RELEASED]
    call ldsku$ (logical_device, user_list, code);
    Returns bit-encoded list of users using a specified logical device.

LOGIN$ [NOT RELEASED]
    call login$ (key, code)
    Turn on and off OS and network logging.

LIBTBL - Library tables; not a routine. [NOT RELEASED]

LIMIT$ (1, 2 bit(8), 2 bit(8), bin(31), bin, bin) III-8-36
    call limit$ (key, limit, reserved, code);
    Set/read cpu, realtime, and login time limits. KeyI = 1 (read), 2 (set). KeyH = 1 (cpu sec), 2 
        (login min), 5 (cpu watchdog sec), 6 (real-time watchdog min), 7 (real-time watchdog sec)

LIST$CMD (char(32) var, bin) II-6-16
    call list$cmd (wildcard_match, status);
    List internal mini-level commands by wildcard match.

LIST$EN (char(128) var, (8) char(32) var, bin, bin, ptr, bin) [NOT RELEASED]
    call list$en (pathname, entrynames, num_total, num_found, 
        rtn_list_ptr, error);
    Return library entrynames in an EPF library.

LN$SET (pointer, bin) [NOT RELEASED]
    call ln$set (smtp, status);
    Sets a library already mapped in into a user's search list.

LOGIN$ (char(256) var, fixed bin) [NOT RELEASED]
    call login$ (com_args, com_status);
    Parsing and routing routine for the LOGIN command.

LOGO$$ (bin, bin, char(*), bin, bin(31), bin) III-2-24
    call logo$$ (key, user_number, user_name, user_name_len, reserved, 
        code)
    Log out a process or user. Key = -1 - all; 0 - self; 1 - user_number; 2 - user_name.

LOGOUS [NOT RELEASED]
    call logou$;
    Initial processor for the LOGOUT command.

LON$CN (bin) III-5-20
    call lon$cn (key);
    Enable or disable logout notification. Key = 0 - off; 1 - on.

LON$PR (bin, (6)bin) [NOT RELEASED]
    call lon$pr (code, msginfo);
    Print phantom logout notification message.

LON$R (ptr, bin, bit(1), bin) III-5-21
    call lon$r (addr(message), message_len, more_waiting, code);
    Retrieve logout info.

Prime Restricted
LOV$SW returns (bit 1)) [NOT RELEASED]
    login_over_login_not_allowed = lov$sw();
    Checks to see if login over login is allowed.

LSRS$DLAY (bin, bin, bin, bin, bin) [NOT RELEASED]
    call lsr$dlay(min, max, margin, who, status);
    Set slope of delay curve for terminal of specified user.

LSRS$ERR (char(*), bin, bin) [NOT RELEASED]
    call lsr$err(message, message_length, status);
    Gives the Login Server a way to log to the console.

LSRS$GETC (bin, char(1), bin) [NOT RELEASED]
    call lsr$getc(line_number, retchar, status);
    Special Login Server gate to let it get characters from its lines.

LSRS$GLSE (bin(31), bin, bin) [NOT RELEASED]
    call lsr$glse(Timeout, NewEvent, Status);
    Routine to return Login Server Event.

LSRS$GTLL ("*"bin, bin, bin, bin) [NOT RELEASED]
    call lsr$gtll(WhichLines, ArraySize, HowMany, Status);
    Get list of loginable "lines" (buffer indices).

LSRS$GTLO (bin, bin, char(*), var, bin) [NOT RELEASED]
    call lsr$gtlo(Who, Why, Command, Status);
    Manage logout information for the Login Server.

LSRS$GTNM (bin, bin) [NOT RELEASED]
    call lsr$gtnm(NewMaxusr, Status);
    Retrieve the maxusr value for the Login Server.

LSRS$GTPR (bin, bin, bin) [NOT RELEASED]
    call lsr$gtpr(who, newprocess, status);
    Obtain a process number for use with a given line.

LSRS$KLR (bin) [NOT RELEASED]
    call lsr$klr(status)
    Post a suicide event for the Login Server.

LSRS$LSR (bin) [NOT RELEASED]
    call lsr$lsr(code);
    Procedure to start up the Login Server.

LSRS$RLI (bin, bin, struc, bin) [NOT RELEASED]
    call lsr$rli(Who, ProcessNo, Attr, Status);
    Start up a user process to be used for logged-out user going remote.

LSRS$STPR (bin, bin, struc, bin) [NOT RELEASED]
    call lsr$stpr(Who, ProcessNo, Attr, Status);
    Routine to start up a local user's process.

LSRS$TNOA (bin, char(*), bin) [NOT RELEASED]
    call lsr$tnoa(user, string, count);
    Login Server terminal output (Login Server only).

LSRS$TRBC (bin, bin, bin) [NOT RELEASED]
call lsr$trbc(line, toWhom, Status);
    Transfer line (buffer) control from one process to another.

LSR$USR (char(80), bin, bin, bin) [NOT RELEASED]
call lsr$user(line, status, for_whom, code);
    USRASR command processor for Login Server.

LUDEV$ (bin, ptr, bin, bin); [NOT RELEASED]
call ludev$(user, addr(rtn_struc), max_devs, code);
    List a user's assigned devices.

LUDSK$ (fixed bin, ptr, fixed bin, fixed bin); II-4-57
call ludsk$ (user, addr(disk_list), max_entries, code);
    Returns list of all disks currently in use by a given user.

LUID$ (bin, bin(31), bin) [NOT RELEASED]
call luid$(unit, uid, code);
    Return a unique ID consisting of the ldev and BRA.

LV$GET (ptr, char(32)var, char(1024)var, bin, bin) II-6-18
call lv$get(vcb_arg, var_name, var_value, var_size, code);
    Get local variable.

LV$SET (ptr, char(32)var, char(1024)var, bin, bin) II-6-20
call lv$set(vcb_ptr, variable, value, code);
    Set local user variables.

M2MSA$(bin, bin) returns(bin) [NOT RELEASED]
    runit = m2msa$(unit, code);
    Returns the master-to-slave mapping for the remote file unit.

MAXUS$ (char(80), bin) [NOT RELEASED]
call maxus$(line, status);
    Carry out the MAXUSR operator command.

MESSG$ (char(32), char(*), bin, bin, bin) [NOT RELEASED]
call messg$(user(*), comline, message, msg_code, code);
    Handle message command.

MGSET$ (bin, bin) III-9-5
call mgset$(key, code);
    Set receiving state for messages. Key = K$ACPT, K$DEFR, K$RJCT.

MIR_OFF_CMD$(char(*),var, bin) [NOT RELEASED]
call mir_off_cmd$(CommandArgs, CommandStatus);
    Process MirrorOff command.

MIR_ON_CMD$(char(*),var, bin) [NOT RELEASED]
call mir_on_cmd$(CommandArgs, CommandStatus);
    Process MirrorOn command.

MKLB$F (int*2, real*8) III-7-20
call mklb$f(fortran_label, rtl_pl1_label)
    Make PL/I compatible label in fortran program.

MKON$F (int*2(*), int*2, external) III-7-21
call mkon$f(condition_name, condition_name_len, routine)
    Create an on-unit in FTN.
MKON$P (char(*), bin, entry) III-7-23
   call mkon$(condition_name, condition_name_len, handler);
   Create an on-unit in F77 or PL1G.

MKONUS (char(*), var, entry, options(shortcall 20) III-7-25
   call mkonus$(condition_name, handler);
   Create an on-unit in PMA, SPL, or PLP.

MKONXS (char(*), var, entry, ptr, bit(16)) options(shortcall(18)) [NOT RELEASED]
   call mkonx$ (condition_name, onunit_proc, specifier, flags);
   Make PL/I on-unit.

MKSH$ (bin(31), bin, bin, ptr, bin) [NOT RELEASED]
   call mksh$(unique_seg_id, req_accesses, limiting_accesses,
            dtar2_seg_ptr, code);
   Make a pure DTAR 2 shared area.

MKSON$ (entry, fixed bin) [NOT RELEASED]
   call mkson$ (sou_routine, code);
   Make a static on-unit in either ring 0 or ring 3.

MM$MLPA (bin, bin) [NOT RELEASED]
   call mm$mlpa(segment, status);
   Make an out of bounds last page available.

MM$MLPU (bin, bin) [NOT RELEASED]
   call mm$mlpu(segment, status);
   Make the last page of a segment unavailable.

MOV$(ptr, ptr, bin) [NOT RELEASED]
   call movb(from, to, number_of_bytes);
   Moves words ((number_of_bytes + 1)/2) from area pointed to by from to area pointed to by to.

MOVEWS (ptr, ptr, bin) III-6-34
   call movew$(from, to, count);
   Move count words from area pointed to by from to that pointed at by to.

MOVWDS (ptr, ptr, fixed bin(31)) [NOT RELEASED]
   call movwds(from, to, number_of_words);
   Moves number_of_words from from to to.

MSG$ (bin, char(*), bin, char(*), bin, bin, char(*), bin, (131) bin) [NOT RELEASED]
   call msg$ (key, from_name, from_user_num, to_name,
             to_user_num, name_len, from_system_name,
             system_name_len, time_sent, text, text_len,
             error_vector);
   Send message using specified banner information. NPX only.

MSG$ST (bin, bin, char(*), bin, char(*), bin, bin) III-9-3
   call msg$st(key, user_num, system_name, system_name_len,
                user_name, uname_len, status);
   Return receiving state of a user. Key = K$READ, 2 (read by user_num).

N$ADDR (char(16)var, char(32)var, bin) [NOT RELEASED]
   call n$addr(address, name, code);
   Add a node "addr block" to the network database.

6-32 Prime Restricted
N$AHCB (char(32)var, char(32)var, bin, bin, bin, bin, bin, bin, bin, bit(16), bin, bin, bin, bin) [NOT RELEASED]
    call n$ahcb(node_name, pdn_name, pdn_flag, maxvc, window,
                  packet_size, block_type, line_no, fdx_flag,
                  slcon, prdec, hcbid, lapfg, i_am_dte, code);
    Add an HCB block and a linedef block to the database.

N$ANAM (char(32)var, bit(16), bit(16), bit(16), char(32)var, char(32)var, char(32)var, char(32)var, bin, bin) [NOT RELEASED]
    call n$anam(node_name, netbits, rlthbits, fambits,
                 npxpsw, ihdxpas, ohdxpas, nodtype, code);
    Add a node "name block" to the network database.

N$APDN (bin, bin, bin, char(32)var, char(4)var, char(6)var, char(6)var, char(6)var, char(6)var, char(6)var, bin) [NOT RELEASED]
    call n$apdn(iii_typ, addr_typ, thru_key, pdn_name, dnic,
                 creq_fctys, cacpt_fctys, rit_fctys, rlg_fctys, code);
    Add a "pdn block" to the network database.

N$APTH (char(16)var, char(32)var, bit(16), bin, char(32)var, bin, bit(1), bin) [NOT RELEASED]
    call n$apth(address, name, access, hcbid, gate_name, pthid,
                 path_online, code);
    Add a "path block" to the network database.

N$ASAD (char(16)var, char(32)var, bin) [NOT RELEASED]
    call n$asad(passed_addr, pdn_name, code);
    Add an address to a source address chain.

N$CHCB(char(32)var, bit(3), bin, ptr, bin) [NOT RELEASED]
    call n$chcb(name, net, line, buffer_ptr, code);
    Modify an existing host block.

N$CHONE(char(32)var, bit(3), bin, ptr, bin) [NOT RELEASED]
    call n$chone(name, net, line, buffer_ptr, code);
    Return description of one host-block (packet-level).

N$INIT (bin) [NOT RELEASED]
    call n$init(code);
    Initialize all the network databases.

N$IPDN (bin) [NOT RELEASED]
    call n$ipdn(code);
    Fill the PDN table with known pdn values.

N$LALL (ptr, bin) [NOT RELEASED]
    call n$lall(buffer_ptr, error_code);
    Gathers statistics for all primenet synchronous lines.

N$LCFG (bin, ptr, bin) [NOT RELEASED]
    call n$lcfg(line_num, buffer_ptr, error_code);
    Gathers configuration statistics for one primenet synchronous line.

N$LDYN (bin, ptr, bin) [NOT RELEASED]
    call n$ldyn(line_num, buffer_ptr, error_code);
    Gathers dynamic statistics for one primenet synchronous line.

N$NETS (bin, bin, bin) [NOT RELEASED]
call n$snets(my_ring_id, ring_block_size, code);
Do final network configuration and setup.

N$SPNC (bin, pointer, bin, pointer) returns(bin) [NOT RELEASED]
status = n$spnc(pnc_number, traffic_buf, traffic_buf_size,
    trace_buffer);
Gather pnc statistics data.

N$RTRC (bit(1), bin) [NOT RELEASED]
call n$rttc(on_off_flag, error_code);
Turn network ring tracing on/off.

N$SPME (char(32)var, bin, bin, bin, char(16)var, bin, bin) [NOT RELEASED]
call n$spme(my_name, maxvc, window, packet, comp_addr,
    hcid, code);
Add all the "myself specific" data to the network databases.

N$VALL (ptr, bin) [NOT RELEASED]
call n$vall(buffer_ptr, error_code);
Gathers data for all virtual circuits.

N$VONE (bin, ptr, bin) [NOT RELEASED]
call n$vone(void, buffer_ptr, error_code);
Gathers statistics for one virtual circuit.

) logical = nameq$(file_name1, file_name1_len, file_name2,
    file_name2_len)
Compare two filenames for equivalence.

NETPRC [NOT RELEASED]
call netprc;
Network process running in ring 0.

NETSET (bin) [NOT RELEASED]
call netset(error_code);
Checks authorization of user starting network & init network segments.

NEWLV$ () [NOT RELEASED]
call newlv$;
Pushes a new command level.

NPX$RL returns(entry(ptr)var) [NOT RELEASED]
entry_point = npx$rl();
Called by SLAVE_CK to retrieve the entry point of any_handler.

NPX$SL (entry(ptr)) [NOT RELEASED]
call npx$sl(entry_point);
Called by SLAVE to store its any_handler in ring 0 data base.

NPXPRC (bin, *, *, *) [NOT RELEASED]
call npxprc(key, arg1, arg2, arg3);
Call random NPX routine. Key = CVTNAM(6), CVTNUM(7), RTICK(12), LOGMES(15),
    WNAME(17), RFMREV(18), CLUPSR(20), CLSbyn(21), RRPOW(24), CHKRO(25),
    RGROUP(26), WGROUP(27), LOGIN(28), LOG21(29), LOG22(30), LOG23(31),
    LOG24(32), LOG25(33), USRTYP(34) (also HBWAIT(22), LOGO5(23), XWAIT(35),
    LOG26(36), LOG44(44), LOG45(45)). Obsolete at 22.0.
NS$CRHOS(char(16)var, bin) [NOT RELEASED]
call ns$crhos(host_name, error_code);
Create a host on an extant LAN.

NS$CRLAN (char(32)var, bit(2), bit(2), (*) char(16)var, bin) [NOT RELEASED]
call ns$crlan(lan_name, unconfig_lts_ok, media_type, 
ntwk_mgmt_host, error_code);
Create a LAN node in the NSS database.

NS$CRLHC (char(32)var, char(16)var, bit(8), bin, bin, char(6), bin) [NOT RELEASED]
call ns$crlhc(lan_name, host_name, function, lhc_number, 
dev_addr, mac_addr, error_code);
Create an LHC on an extant host and LAN.

NS$CRLTS (char(32)var, char(16)var, bit(8), char(6), bin) [NOT RELEASED]
call ns$crlts(lan_name, lts_name, function, mac_addr, 
error_code);
Create an LTS on an extant LAN.

NS$DLTSA (char(6), bin) [NOT RELEASED]
call ns$dltsa(mac_addr, error_code);
Delete an LTS by address.

NS$DLTSN (char(16)var, bin) [NOT RELEASED]
call ns$dltsn(lts_name, error_code);
Delete an LTS by name.

NS$FLAG (bit(1), bin, bit(1), bin) [NOT RELEASED]
call ns$flag(write, flag_no, value, code);
Read or write NSS client visible flag.

NS$FLFUN (bit(8), bin) [NOT RELEASED]
call ns$flfun(functions, error_code);
Flush a function from the NSS database.

NS$RHA (char(6), pointer, bin) [NOT RELEASED]
call ns$ra(mac_addr, host_rec_p, error_code);
Read host description by address.

NS$RH (bin, char(*)var, char(10), pointer, bin) [NOT RELEASED]
call ns$rai(key, name, handle, host_rec_p, error_code);
Read host and LHC descriptions.

NS$RLA (char(6), pointer, bin) [NOT RELEASED]
call ns$ral(mac_addr, lts_rec_p, error_code);
Read LTS description by address.

NS$RLI (bin, char(32)var, char(10), pointer, bin) [NOT RELEASED]
call ns$rali(key, name, handle, lts_rec_p, error_code);
Read LTS description.

NS$RN (bin, char(32)var, char(10), pointer, bin) [NOT RELEASED]
call ns$rn(key, lan_name, handle, lan_rec_p, error_code);
Read LAN description.

NS$SEC (bin) [NOT RELEASED]
call ns$sec(code);
Ensure that caller is user 1 or ACL group member.
NSSFUNA (bin, char(6), bit(8), bin) [NOT RELEASED]
call ns$sfuna(key, mac_addr, function, error_code);
Set the function of an LHC or LTS based upon MAC address.

NSSFUNI (bin, char(16),var, bin, bit(8), bin) [NOT RELEASED]
call ns$sfuni(key, host_name, lhc_number, function,
error_code);
Set the function of an LHC based upon host name and lhc number.

NSSSSTAA (char(6), bit(3), bin) [NOT RELEASED]
call ns$sstaa(mac_addr, state, error_code);
Set the state of an LHC or LTS based upon MAC address.

NSSSSTAI (char(16), var, bin, bit(3), bin) [NOT RELEASED]
call ns$sstai(host_name, lhc_number, state, error_code);
Set the state of an LHC based upon host name and lhc number.

NSSXAN (char(6), char(16), var, bin, bin) [NOT RELEASED]
call ns$sxan(mac_addr, name, lhc_number, error_code);
Translate an address to a name (and, for hosts, an LHC number).

NSSXNA (char(16), var, bin, char(6), bin) [NOT RELEASED]
call ns$sxna(name, lhc_number, mac_addr, error_code);
Translate a name to an address.

NT$AS (bin, bin, char(16), var, bin, bit(1), bin) [NOT RELEASED]
call nt$as(primos_line, media_type, lts_name,
lts_line, permanent, error_code);
Associate an LTS line with a Primos line number.

NT$CHECK (bit(16), bin) [NOT RELEASED]
call nt$check(lhc_list, error_code);
Check for required LHCs configured anddownline loaded.

NT$CM [NOT RELEASED]
call nt$cm
NTS connection manager (part 1).

NT$CMODE (bin) [NOT RELEASED]
call nt$cmode(status);
Force the NTS terminal line of the current process back to LTS command mode.

NT$INIT (char(128), var, bin) [NOT RELEASED]
call nt$init(config_name, error_code);
Initialize NTS database.

NT$LTS (bin, bin, char(16), var, bin, char(6), bin) [NOT RELEASED]
call nt$lts(primos_line, media_type, lts_name,
lts_line, mac_address, error_code);
Return LTS line connection info.

NT$NAME (char(128), var, bin) [NOT RELEASED]
call nt$name(config_pathname, error_code);
Return NTS config file pathname.

NT$RAS (bin, bin, char(32), var, char(6), bin, bit(1), bin, char(32), var, bin) [NOT RELEASED]
call nt$xas(line, user_no, user_name, lts_address, lts_line, permanent, as_user_no, as_user_name, error_code);
Read an entry from the NTS associate table.

NT$START (bin, bin) [NOT RELEASED]
call nt$start(lhc_number, error_code);
Start NTS.

NT$STOP (bin, bin) [NOT RELEASED]
call nt$stop(lhc_number, error_code);
Stop NTS.

NT$UAS (bin, bin, char(16)var, bin, bin) [NOT RELEASED]
call nt$uas(pramos_line, media_type, lts_name, lts_line, error_code);
Dissociate an LTS line from a Pramos line number.

OERRTN (bin, bin, bin, char(*), bin, char(*), bin) [NOT RELEASED]
call oerrtn(alt_val, alt_rtn, code, text, text_len, name, name_len);
Old style error handling.

OPENSB (bin, char(*) var, bin, bin, bin) returns(bin(31)) [NOT RELEASED]
char_pos = open$sb(open_key, tree, unit, type, code);
Open a branch by tree name (nonstandard).

OPN$SR (char(32)var, char(128)var, char(128)var, bin, bit(5), char(128)var, bin, bin, bin) [NOT RELEASED]
call opn$sr(search_list, referencing_dir, file_path, open_mode, types, found_path, out_unit, out_type, code);
Open file using a search list. (Obsolete; will be removed; Use OPSR$).

OPN$SRSF (char(32)var, char(128)var, ptr, bin, bin, bit(5), char(128)var, char(128)var, bin, bin, bin) [NOT RELEASED]
call opn$srsf(search_list, file_path, suffix_list_ptr, n_suffixes, open_mode, types, referencing_dir, suffix_index, file_base_name, found_path, out_unit, out_type, code);
Open file using a search rule and suffixes. (Obsolete, will be removed; use OPSRSS$).

OPSR$ (char(32)var, char(128)var, bit(16), bin, char(128)var, bin, bin, char(128)var, var, bin) [NOT RELEASED]
call opsr$(list_name, referencing_dir, valid_types, open_key, file_path, unit, out_type, found_path, code);
Open a file system object using a search list.

OPSR$S (char(32)var, char(128)var, bit(16), bin, char(128)var, bin, bin, ptr, char(32)var, bin, char(128)var, var, bin) [NOT RELEASED]
call opsr$s(list_name, referencing_dir, valid_types, open_key, file_path, unit, out_type, n_suffixes, suffix_list_ptr, basename, suffix_index, found_path, code);
Open an object using search rules and suffix processing.

PA$DEL (char(32)var, bin) II-2-27
call pa$del(partition_name, code);
Delete a priority ACL.

PA$LST (char(128)var, ptr, bin, bin) II-2-28
call pa$lst(object_pathname, addr(acl_struc), max_acl_entries, code);
Read a priority ACL.

PA$LSTO (char(32)var, ptr, bin, bin) [NOT RELEASED]
call pa$lsto(object_name, logical_acl_ptr, max_entry_count, code);
Return the contents of a priority ACL in logical format.

PA$SET (char(32)var, ptr, bin) II-2-30
call pa$set(partition_name, addr(acl_struc), code);
Set a priority ACL.

PAR$RV (char(32)var, bin) returns(bin) II-4-59
rev_no = par$rv(partname, code);
Returns the partition rev. stamp of a named disk partition

PBH$GD ((1024)bin31), 1, 2 bin, 2 like pbhcom, bin) [NOT RELEASED]
call pbh$gd(arg_counters, arg_struc, code);
Get data for PB histogram.

PBH$ON (bin, bin, (max_num_segs)bin12), bin) [NOT RELEASED]
call pbh$on(arg_user_number, arg_num_segs, arg_seg_numbers, code);
PB Histogram Facility Startup/Access entries.

PHANT$ (char(*), bin, bin, bin, bin) III-10-8
call phant$(file_name, file_name_len, file_unit, user_num, code);
Start a phantom (Obsolete; use PHNTM$).

PHDBG (ptr, bin, bin) [NOT RELEASED]
call phdbg(free_store_area_ptr, length, code);
Returns addresses of common area for protocol handler. (RJE)

PHNTM$ (bit(16), char(32), bin, bin, bin, bin, char(128), bin) III-5-23
call phntm$(cpl_flag, file_name, file_name_len, file_unit, user_num, code, cpl_args, cpl_args_len)
Start a phantom.

PIDSCK (1, 2 char (8), 2 fixed bin) returns (bit(1) aligned) [NOT RELEASED]
id_is_valid = pid$sck (target_uusid);
Validates process unique id.

PIDSGET (char(8)) [NOT RELEASED]
call pid$sget(unique_id);
Get the PID of the current process.

PK2LDV (char(*)) var, bin, bin, bin) [NOT RELEASED]
call pk2ldv(packname, packlen, node, ldev)
Convert disk pack name, node number into a logical device number.

PMGS$ [NOT RELEASED]
call pmsg$;
Print messages on the caller's terminal.

PNM$CHK (bin, char(32)var, bin, bin) [NOT RELEASED]
call pnm$chk(hc_nbr, lan300_name, dev_addr, error_code)
Performs the consistency check for Ethernet Host Controller.

6-38 Prime Restricted
PNM$RLHB (bin, ptr, bin) [NOT RELEASED]
call pnm$rlhb(hctbl_number, hctbl_info, return_code);
Access data from the LHCTBL Data Structure.

PNM$RNMB (bin, bin, pointer, bin) [NOT RELEASED]
call pnm$rnmb(action_code, data_from_rmdb, sem_addr,
return_code);
Access data from the Ring0 Network Management Data Structure.

PNM$SEC (bin) [NOT RELEASED]
call pnm$sec(code);
Security check for Network Management gates.

PNM$WLHB (bin, bin, ptr, bin) [NOT RELEASED]
call pnm$wlhb(action_code, hctbl_number, hctbl_info,
return_code);
Update the LHCTBL data structure.

PNM$WNMB (bin, (2)bin, bin) [NOT RELEASED]
call pnm$wnmb(action_code, data_for_rmdb, return_code);
Update the Network Management Ring0 data structure

PRERR (bin) (svc = 0111) III-10-9
call prerr(user);
Print name and/or message from user’s ERRVEC (obsolete).

PRI$RV (char(16)var) III-2-12
call pri$rv(primos_rev);
Returns the Primos rev. stamp of the currently running operating system.

PRI$CH (bin, bin, bin) [NOT RELEASED]
call pri$ch(pdev_index, pratio, err_code);
Routine to change PRATIO values.

PRI$PD (bin, bin) [NOT RELEASED]
call pri$pd(pdev_count, err_code);
Routine to return the number of paging partitions on the system.

PRI$ST (bin, bin, bin, bin) [NOT RELEASED]
call pri$st(pdev_index, pratio, ldev, err_code);
Routine to return a specific pratio value.

PRJIDS (char(32)var) III-2-26
call prjids(project_id);
Return project ID of current user.

PRVSB_ (ptr, bit(1), bit(1), bin) returns (ptr) [NOT RELEASED]
prev_ptr = prvsb_ (crrr_ptr, crawl_flag, fix, cs_depth);
Find previous stack frame given pointer to current one.

PRWF$$ (bin, bin, ptr, bin31, bin, bin) (svc = 1506) II-4-61
call prwf$$ (key, file_unit, addr(buffer), num_words, position,
num_words_transferred, code);
Position, read or write to a file. Key = (K$READ, K$WRIT, K$POSN, K$TRNC, K$RPOS) +
(K$PREP, K$POSR, K$PREA, K$POSA) + (K$CONV, K$FRCW)

PTIME$ returns(bin(31)) III-2-27
process_time = ptime$();
Returns process time since logged in.

PTRAP$ (P3TRAP) [NOT RELEASED]
CALS PTRAP
FIM for restricted mode (RXM) and illegal instruction (ILL).

PWCHK$ (bin, char(*)var) returns(bit(1)) III-2-28
password_ok = pwchk(key, password)
Check a password for valid format. Key = K$UPRC, K$NULL.

PWDIR$ (bit(1), bin) [NOT RELEASED]
call pwdir$ (on_or_off, code);
Enable/Disable creation of password directories.

PX$BIRTH (bin, bin, char(34), bin) [NOT RELEASED]
call px$birth (my_id, parent_id, command, status_code);
Record the birth of a Primix process.

PX$CREA (char(128) var, ptr, bin) [NOT RELEASED]
call px$crea (dirname, info, code);
Special version of dir$cr -- presets ACL.

PX$CREA0 (char(32) var, bin) [NOT RELEASED]
call px$crea0 (dirname, code);
Special version of dir$cr -- presets ACL (ring 0 part, sets ACL).

PX$CWAIT (bin, bin) [NOT RELEASED]
call px$cwait (user_id, status_code);
Primix PM support for pause system call.

PX$DEATH (bin, bin, ptr, bin, bit(1), bin) [NOT RELEASED]
call px$death (my_id, child_status, snnode_ptr, snnode_count, parent_wait, status_code);
Record the death of a Primix process.

PX$DUMP (bin, bin, ptr, bin) [NOT RELEASED]
call px$dump (my_id, expected_version, ptr_dump_table, status_code);
Primix dump/who/write/wall commands support.

PX$EXEC (bin, char(34), bin) [NOT RELEASED]
call px$exec (my_id, command, status_code);
Record the name of the command being executed for Primix.

PX$INIT (bin, bin, bin) [NOT RELEASED]
call px$init (ver_num, lisc_number, status_code);
Initialize Primix.

PX$MXUSR (bin, bin) [NOT RELEASED]
call px$mxusr (max_users, status_code);
Handles the SET_PRIMIX_USERS command.

PX$PAUSP (bin, bin) [NOT RELEASED]
call px$pausp (user_id, status_code);
Primix PM support for pause system call.
PX$PDATA (bin, bin, bin, (*bin, bin)) [NOT RELEASED]
call px$pdada(user_id, expected_version, buf_size, buffer, 
status);
Return Primix process data for the indicated user.

PX$RDSIG (bin, bin, bin, (*bin(31), bin)) [NOT RELEASED]
call px$rdsig(user_id, num_expected, num_returned, signals, 
status_code);
Return current Primix signal.

PX$SGACT (bin, bin(31), bin) returns(ptr) [NOT RELEASED]
action = px$sgact(pid, signal, status_code);
Return current response to a Primix signal.

PX$SGSYS (bin, bin(31), ptr, ptr, bin) [NOT RELEASED]
call px$sgsys(user_id, signal, action, prev_action, status_code);
Primix PM support for the Signal System call function.

PX$SHDWN (bin) [NOT RELEASED]
call px$shdwn(code);
Shut down Primix.

PX$SIGNL (bin, bin(31), bin, bin) [NOT RELEASED]
call px$signl(user_id, signal_num, target, status_code);
Signal a process for Primix PM support.

PX$SRCH (bin, char(128)var, bin, bin, bin) [NOT RELEASED]
call px$srch(action+ref+newfil, filename, funit, type, code);
Special version of srch$$ for creating items with preset ACL.

PX$SRCH0 (char(32)var, bin) [NOT RELEASED]
call px$srch0(filename, code);
Special version of srch$$ for creating items with preset ACL.

PX$SVTIM (bin, bin(31), bin(31), bin) [NOT RELEASED]
call px$svtim(key, cpu, io, code);
ates the CPU and I/O time for the forked process.

PX$SYNC (bin, bin) [NOT RELEASED]
call px$sync(user_id, status_code);
Primix PM support for fork synchronization.

PX$UNSYNC (bin, bin) [NOT RELEASED]
call px$unsync(user_id, status_code);
Primix PM support for fork synchronization.

PX$WAITP (bin, bin, bin, ptr, bin) [NOT RELEASED]
call px$waitp(user_id, child_status, child_id, file_info_ptr, 
file_info_count, status_code);
Primix PM support for wait system call.

Q$READ (char(128)var, (8)bin(31), bin, bin, bin) II-4-68
  call q$read(path_name, quota_info, quota_info_len, dir_type, 
code)
Read quota information.

Q$READ0 (char(32)var, (8)bin(31), bin, bin, bin) [NOT RELEASED]
call q$read0(dir_name, output_structure, max_entries, dir_type, code);
Read quota information for current directory.

Q$SET (bin, char(128)var, bin(31), bin) II-4-71
  call q$set(key, path_name, max_quota, code);
  Set quota maximum.  key = K$SMAX.

QUIT$ (bit(16) aligned) III-3-62
  call quit$ (pending_quit);
  Determine if there are any pending quits. pending_quit = 0 if none.

QUOTE_ (char(*), var, char(*), var, bin, bin) [NOT RELEASED]
  call quote_ (input_string, output_string, output_size, status);
  Quote a given string.

R$ALLC(ptr, fixed bin) returns(ptr) [NOT RELEASED]
  smt_pointer = r$allc(smt, status);
  Allocate linkage for an EPF. Obsolete after 19.3; use EPF$ALLC.

R$AL01 (char(8), bin) returns(bin) [NOT RELEASED]
  alloc_count = r$al01(slave_id, code);
  This routine increment the ALOCNT by 1.

R$ALOC(fixed bin) [NOT RELEASED]
  call r$aloc(remote_node);
  Allocate an index to a slot in VCDATA for a node number. (NPX)

R$BGIN (bin, char(8), char(*), bin, (*8392)bin, bin(31), bin, variable) [NOT RELEASED]
  call r$bgin(key, slave_id, subr_name, subr_name_len, buffer,
     buffer_len, code [], arg1, arg1len, arg1key, ...,
     arg15, arg15len, arg15key]);
  The user callable interface to NPX for synchronous and asynchronous RPCL. Key = 0 (2-
  called by R$CALL).

R$CALL (bin, bin, char(*), bin, bin, variable) [NOT RELEASED]
  call r$call(key, rnode, subroutine_name, subroutine_namlen,
     rcode, arg1, arg1len, arg1key, arg2, arg2len,
     arg2key, ...);Perform remote procedure call. Key = 0, K$FUNC.

R$CKNT (char(32)var, bin) [NOT RELEASED]
  call r$cknt(node_name, code);
  Subroutine to check the validity of the supplied node name.

R$CPF (ptr, bit(3), fixed bin, bit(1), bit(4)) [NOT RELEASED]
  call r$cpf(smt, expand_wildcards, eq_position, vfy_default,
     match_type_default);
  Get command processor flags from an EPF. Obsolete after 19.3.

R$CVT(char(32), bin) returns(bin) [NOT RELEASED]
  nodenum = r$cvt(node_name, node_name_length);
  Convert node name to the corresponding node number. Obsolete after 19.3; use NPXPRC.

R$DEL (ptr) [NOT RELEASED]
  call r$del(smt);
  Delete an EPF from a user's address space. Obsolete after 19.3.
R$END (bin, char(8), bin, bin, bin) returns(bin(31)) [NOT RELEASED]
  func_rtn = R$end(key, slave_id, buffer, time, code);
  The asynchronous remote procedure call-end, check slave's task.

R$MYNM (char(32)var) [NOT RELEASED]
call R$ynm(system_name);
  Return name of local node.

R$RLS((fixed bin(15))) [NOT RELEASED]
call R$rls(xrnnode);
  Decrement slave allocation count. (NPX)

R$SLID (char(32)var, char(8), bin) [NOT RELEASED]
call r$slid(node_name, slave_id, code);
  Subroutine to convert node name to slave id if the VC is secured.

R$SLST (struct, bin, bin) [NOT RELEASED]
call R$slist(slave_list, slave_list_size, error_code);
  Return a list of a user's active slaves.

R$SYSN (char(32)var, char(8), bin) [NOT RELEASED]
call R$syns(slave_id, node_name, code);
  Subroutine to return the system name for a given slave_id.

R$WAIT ((*)bin) [NOT RELEASED]
call R$wait(buffer);
  Wait for a call request and initialize user profile.

R$WHERE (bin, char(*) var, bin, bin) [NOT RELEASED]
call R$where(key, fnsname, unit, code);
  Returns the location of a file (local or remote). Obsolete.

R$ABUF (bin, bin, bin)returns(bit(1)) [NOT RELEASED]
  success = R$abuf(number, avail, code);
  Allocates "reserved buffers" for R0AM users.

R$BL (bin, bin(31), ptr, bin, bin) [NOT RELEASED]
call R$bl(bi_unit, bi_address, buffer_ptr, bi_fileid, code);
  Writes before images for R0AM.

R$CHK (bin) returns(bin) [NOT RELEASED]
  status = R$chk(key);
  Checks if R0AM ring zero is initialized.

R$FBUF (bin, bin, bin, bin, bit(1), bit(1), bin) [NOT RELEASED]
call R$fbuf(user, num_free, num_avail, num_freed, release_flag, had_zero, code);
  De-allocates "reserved buffers" for R0AM users.

R$INI (bin, bin) [NOT RELEASED]
call R$ini(num_buffers, code);
  Initializes R0AM ring zero data structures.

R$PUR (bin, bin) [NOT RELEASED]
call R$pur(fileid, code);
  Purges the specified file from the R0AM buffer pool.
R0$RBUF (bin, bin, ptr, buffer, bin) [NOT RELEASED]
call r0$rbuf(key, priority, buffer_ptr, file_id, code);
   Releases R0AM buffer(s).

R0$RW (bit(16), bin, bin(31), bin, bin, ptr, ptr, bin) [NOT RELEASED]
call r0$rw(key, unit, address, length, access, fileid,
   user_buf_ptr, shared_buf_ptr, code);
   R0AM ring zero buffer manager.

R0$RWM (bit(16), bin, bin(31), bin, bin, ptr, ptr, bin) [NOT RELEASED]
call r0$rwm(key, unit, xpage_num, access, fileid,
   user_buf_ptr, shared_buf_ptr, code);
   R0AM ring zero buffer manager.

R0BASE (ptr) [NOT RELEASED]
call robase(r0_first_ptr);
   Get a pointer to the first frame on the ring 0 stack.

R3FALT [NOT RELEASED] Ring 3 fault table.

RBK$ (bin, bin(31), ptr, bin, bin) [NOT RELEASED]
call rbk$ (unit, logical_block, buffer_ptr, words_read, code);
   Logical Block I/O block read routine.

RCINF$ (bit(16), ptr, bin) [NOT RELEASED]
call rcinf$ (pdev, info_structure_ptr, code);
   Return information about disk controller.

RD$CED, RD$CE_DP (bin) II-6-22
   call rd$ced (program_session_depth)
   Return to the current depth of the command env. program session.

RDEN$ (bin, bin, (')1, 2 bin, 2 char(32), 2 (7)bin, bin, bin(31) or char(32), bin, bin) (svc = 1507) A-9
   call rdens$ (key, file_unit, buffer, buffer_len, rtn_buffer_len,
   file_name, name_len, code)
   Position and read from a UFD. (Obsolete; use DIR$RD and ENTRD)

RDLIN$ (bin, char('), bin, bin) (svc = 1525) II-4-74
   call rdlin$ (file_unit, buffer, buffer_len, code);
   Read a specified number of characters. buffer_len is size in words.

RDTK$ (bin, (8)bin, char( '), bin, bin) (svc = 1517) III-3-16
   call rdtk$ (key, info, token, token_len, code);
   Parse a command line (Obsolete; use CL$PIX or CL$PAR).

RDTKP (bin, (8)bin, char( ')), bin, char(') var, bin, bin) [NOT RELEASED]
call rdtkp (key, info, buffer, buflen, com_line, com_state,
   code);
   Parse a command line (use CL$PIX).

READY$ (bit(16) aligned, fixed bin) III-2-29
   call ready$ (format_sw, error_code);
   Print the ready message on the terminal.

RECYCL (svc = 0505) [NOT RELEASED]
call recyc1
    Pass control to next user.

REMEPF$ (bin, char(*) var, bin) II-5-22
    call remepf$(key, epf_tree_name, status);
    Remove an EPF from a user's environment. Key = K$FRC_DEL, K$NO_FRC_DEL.

REST$$ ((9)bin, char(32), bin, bin) (svc = 1520) III-5-13
    call rest$$ (r_vector, file_name, file_name_len, code)
    Read an R-mode runfile.

RESU$$ (char(32), bin) (svc = 1521) III-5-15
    call resu$$ (file_name, file_name_len)
    Restore and execute an R-mode runfile.

RIPC$C (bin, char(*)var, bin, bin) [NOT RELEASED]
    call rpc$c (uid, node, mbx_id, code);
    Close a IPC mailbox using the mbx_id specified (Remote; NPX).

RIPC$GU (bin, char(*)var, bin, bin, ptr, bin, bin, bin) [NOT RELEASED]
    call rpc$gu (uid, node, key, mbx_id, buf_ptr, buf_size,
                 returned_size, code);
    Get the desired mailbox user ID specified by key (Remote; NPX).

RIPC$NF (ptr, bin) [NOT RELEASED]
    call rpc$nf (receiver_ptr, code);
    Interrupt a specified IPC user by mailbox user ID (remote; NPX).

RIPC$O (bin, bin, char(*)var, bin, char(8), char(*)var, bin, bin) [NOT RELEASED]
    call rpc$o (access_key, notification_key, entryname, mbx_uid,
               uu_name, node, mbx_id, code);
    Open an IPC mailbox for specified access using entryname for access control (remote
    version for NPX).

RIPC$R (bin, char(*)var, bin, ptr, bin, bin, bin, bin) [NOT RELEASED]
    call rpc$r (uid, node, mbx_id, buf_ptr, buf_size, msg_size,
                mbx_send_uid, code);
    Receive a message from specified IPC mailbox waiting if specified (NPX only).

RIPC$SA (bin, char(*)var, bit(1), bin, ptr, bin, ptr, bin, bin, bin) [NOT RELEASED]
    call rpc$sa (uid, node, mbx_id, msg_ptr, msg_size,
                 addr (recvrr_list), max_recvr, num_recvr, code);
    Send a message to any IPC user attach to specified mailbox (NPX only).

RIPC$SB (bin, char(*)var, bit(1), bin, ptr, bin, ptr, bin, bin, bin) [NOT RELEASED]
    call rpc$sb (uid, node, mbx_id, msg_ptr, msg_size,
                 addr (recvrr_list), max_recvr, num_recvr, code);
    Send a message to all IPC users attach to specified mailbox (NPX only).

RIPC$SS (bin, char(*)var, bin, bin, ptr, bin, ptr, bin) [NOT RELEASED]
    call rpc$ss (uid, node, mbx_id, mbx_uid, msg_ptr, msg_size,
                 receiver_ptr, code);
    Send a message to a specific IPC user (NPX only).

RIPC$ST (bin, char(*)var, bin, bin, bin, bin) [NOT RELEASED]
    call rpc$st (uid, node, key, mbx_id, value, code);
    Return various IPC statuses determined by user specified key (NPX only).
RJSATT (bin, ptr, ptr, ptr, (2)bin) [NOT RELEASED]
call rj$satt(key, addr(line_info), addr(device_info),
            addr(other_info), errvec);
    Allow process to attach for line.

RJSDET (bin, bin, (2)bin) [NOT RELEASED]
call rj$det(key, line, errvec);
    Disable the line. Key = 0 if drop DTR.

RJSINF (bin, ptr, (3)bin) [NOT RELEASED]
call rj$inf(worker_id, addr(rtn_info), errvec);
    Return control information from the protocol handler.

RJSINF (bin, ptr, ptr, bin, bin, (3)bin) [NOT RELEASED]
call rj$inp(worker_id, addr(rtn_info), addr(buffer), buffer_len,
            msg_type, errvec);
    Receive a block of data from the RJL.

RJSMSG (bin, bin, char(80)var) [NOT RELEASED]
call rj$msg(type, num, string);
    Return RJE message.

RJSOUT (bin, ptr, ptr, (2)bin) [NOT RELEASED]
call rj$sout(key, addr(info), addr(buffer), errvec);
    Queue a block of data for transmission.

RJSSET (bin, bin, bin, (2)bin) [NOT RELEASED]
call rj$set(line, key, param, errvec);
    Send request to protocol handler.

RJDBG (ptr, bin, bin) [NOT RELEASED]
call rj$dbg(com_block_ptr, length, code);
    Debug gate returns pointer to RJL common blocks for worker RJL.

RJMNI T (bin, ptr, bin) [NOT RELEASED]
call rj$mni t(line, ptr_to_structure, return_code);
    Ring 0 code required to run the Monit facility.

RJPROC (bin, bin) [NOT RELEASED]
call rj$proc(chap_level, code);
    Main driver for RJE emulator process.

RLSLV$ [NOT RELEASED]
call rslsv$;
    Restore a command environment level.

RMSGD$ (char(*), bin, bin, char(*), bin, bin, char(*), bin) III-9-7
call rmsgd$(sender_uname, username_len, sender_unum, system_name,
            system_name_len, time_sent, message, msg_len);
    Receive a deferred message. Time_sent in minutes past midnight.

ROM$CN (char(32), bin, char(32), bin, bin [, bin]) [NOT RELEASED]
call rom$cn (old_name, old_namel, new_name, new_namel, code,
              open);
    Changes the name of an RBF file.

ROM$D0 (char(32)var, bin) [NOT RELEASED]
call rom$d0(obj_name, code);
   Delete a ROAM file in current directory.

ROM$DL (char (128) var, fixed bin) [NOT RELEASED]
call rom$dl (obj_path, code);
   Delete a ROAM file.

RPL$ (char(*) var, char(*) var, char(*) var, bit(1), bin) II-5-24
   call rpl$(source_path, target_path, rpl_path, no_query, code);
      Replace one EPF with another.

RPL$CN (char(*) var, char(*) var, bit(1), bin) [NOT RELEASED]
call rpl$cn(target_tree, rpl_tree, no_query, code);
   Change the name of an open EPF.

RRECL$ (struc, (3)ptr, (3)bin, bin(31), bit(16), bin) [NOT RELEASED]
call rrecl$(nch, buf_ptrs, buf_len, rec_adr, pdev, code);
   Handle READ requests for ASSIGNED disks.

RSEGAC$(bin, (2)bin) III-2-13
   have_access = rsega$(segno, access);
      Function which returns per ring access to the segment if segment is in use.

RTIME$(1, 2 bin(32), 2 bin) [NOT RELEASED]
call rtime$(rt_data);
   Return real-time as 48 bit value in PIC counts.

RTNSDTR3 (ptr, bin) [NOT RELEASED]
call rtn$st3 block_ptr, code);
   Return storage allocated from DTAR 3 segments through GET$DTR3.

RTNSG$ (bin, bin, bin, bin) [NOT RELEASED]
call rtn$sg$(segment_number, code [, user, epf_delete ok]);
   Returns segments to the system. -1 - all static mode; -2 all static & 5002; -3 all user segs;
   -4 all user and 6002.

RVON$F (int*2(*), int*2) III-7-28
   call rvon$F(condition_name, condition_name_len)
      Revert an on-unit (F77 or FTN).

RVONU$ (char(*)var) III-7-29
   call rvonu$(condition_name);
      Revert an on-unit (PL1G, SPL, PLP or PMA)

RVSON$ (entry, fixed bin) [NOT RELEASED]
call rvson$(static_on_unit, code);
   Remove a static on unit.

S$ATRB (1, 2 bin, 2 bit(1), 2 bit(1), 2 bin, 2 bit(1), 2 bin, 2 bit(1), 2 bin, 2 bit(1), 2 bin,
   bin) [NOT RELEASED]
call s$atrb(attr, status);
   Sets up default attributes (in memory copy) for the system.

S$ATRG (1, 2 bin, 2 bit(1), 2 bit(1), 2 bin, 2 bit(1), 2 bin, 2 bit(1), 2 bin, 2 bit(1), 2 bin, 2 bit(1), 2 bin,
   1, 2 bit(1), 2 bit(1), 2 bit(1), 2 bit(1), bin) [NOT RELEASED]
call s$atrg (attr, legalz, status);
   Range checks for attributes.
SAL$SYS1 (bin(31), bin) returns(ptr) [NOT RELEASED]
block_ptr = sal$sys1(block_size, ercode);
System Class Storage Allocator.

SAL_HP(bin, ptr, bin(31), bin) [NOT RELEASED]
block_ptr = sal_heap(storage_class, hcb_ptr, block_size, ercode);
Allocate heap storage.

SANAMS (char(32) var) [NOT RELEASED]
call sanams(system_administrator_id);
Returns user id of system administrator.

SATR$(bin, char(32), bin, var, bin) (svc = 1510) II-4-76
call satr$(key, object_name, object_name_len, attributes, code);
Set or modify a file's attributes. Key = K$PROT, K$D TIM, K$DM PB, K$RW LK, K$SDL.

SAVE$( (9) bin, char(32), bin) (svc = 1522) III-5-17
call save$(rmode_vector, file_name, file_name_len, code);
Save an R-mode runfile.

SC$CLR (bit(32), bin) [NOT RELEASED]
call sc$clr(key, code);
Disable the signalling of the synchronous conditions.

SC$CLR0 (bit(16), bin) [NOT RELEASED]
call sc$clr0(key, code);
Disable ring 0 synchronous conditions.

SC$PRB - see SEC$PRB.

SC$RD (bit(32), (')bin) [NOT RELEASED]
call sc$rd(key, signal_status);
Return the value of synchronous condition flags.

SC$RD0 (bit(16), bin) [NOT RELEASED]
call sc$rd0(key, status);
Read the status of ring 0 synchronous conditions.

SC$RST0 (bit(16), bin) [NOT RELEASED]
call sc$rst0(key, code);
Reset ring 0 synchronous condition status.

SC$SET (bit(32), bin) [NOT RELEASED]
call sc$set(key, code);
Enable the signalling of the synchronous conditions.

SC$SET0 (bit(16), bin) [NOT RELEASED]
call sc$set0(key, code);
Enable ring 0 synchronous conditions.

SCH$RD (fixed bin, fixed bin, fixed bin) [NOT RELEASED]
call sch$rd(key, value, code);
Scheduler variable read subroutine.

SCH$ST (fixed bin, fixed bin, fixed bin) [NOT RELEASED]
call sch$st(key, value, code);
Scheduler variable set subroutine.
SEARCH_C, SEARCH_CASELESS_HASH_TABLE$, SRCH$CHT (ptr, char (32) var, bin) returns (ptr) [NOT RELEASED]
data_address = search_caseless_hash_table$ (table_address, name, code);
Search a standard hash table regardless of case.

SEARCH_H, SEARCH_HASH_TABLE$, SRCH$HTB (ptr, char (32) var, bin) returns (ptr) [NOT RELEASED]
data_address = search_hash_table$ (table_address, name, code);
Search a standard hash table for name.

SEC$AUD [NOT RELEASED]
call sec$aud

SEC$MON (ptr, bin) [NOT RELEASED]
call sec$mon (sec_ptr, code);
Start up or change status of SECURITY_MONITOR (privileged).

SEC$PROB, SC$PRB (char(80) var, bin, bin, bin, bin, bin, bin, char(*) var) [NOT RELEASED]
call sec$prob (program, event_group, event_number, ev_type, code, obj_len, obj_type, obj_arg);
Record an event in the security audit trail.

SEC$ST (ptr, bin) [NOT RELEASED]
call sec$st (stat_ptr, code);
Ring 0 gate to implement SECURITY_STATUS command (privileged).

SEGAC$ (ptr, fixed bin, fixed bin) [NOT RELEASED]
call segac$ (segment_pointer, access, code);
Changes access of a segment.

SEM$CL (bin, bin) III-8-17
   call sem$cl (sem_num, code);
Close named semaphore.

SEM$DR (bin, bin) III-8-19
   call sem$dr (sem_num, code);
Drain semaphore.

SEM$NF (bin, bin) III-8-21
   call sem$nf (sem_num, code);
Notify semaphore.

SEM$OP (char(32), bin, bin, (*) bin, bin) III-8-23
   call sem$op (file_name, file_name_len, sem_num, ids, code);
Open a semaphore by name.

SEM$OU (bin, bin, (*) bin, bin) III-8-23
   call sem$ou (file_unit, sem_num, ids, init_val, code);
Open a semaphore by file unit.

SEM$ST (bin, bin, bin, bin, (128) bin, bin) [NOT RELEASED]
call sem$st (key, sem_nbr_in, sem_nbr_out, wait_count, proc_cnt, proc_nbr, status);
Return status of a semaphore.
SEMS$TN (bin, bin(31), bin(31), bin) III-8-27
   call semS$tn(sem_num, first_wait_msec, other_wait_msec, code);
   Set timer for numbered semaphore.

SEMS$TS (bin, bin) returns(bin) III-8-29
   sem_value = semS$ts(sem_num, code);
   Test counter for semaphore.

SEMS$TW (bin, bin, bin) III-8-31
   call semS$tw(sem_num, time_in_tenths, code);
   Timed wait for named semaphore. code = 1 -> timed out.

SEMS$WT (bin, bin) III-8-33
   call semS$wt(sem_num, code);
   Wait on a semaphore.

SET$OR (bin, bin) [NOT RELEASED]
   call setS$or(key, code);
   Set initial attach point (origin).

SETRC$ (bin) III-5-9
   call setRC$(error_code);
   Set static mode error code.

SETREG ((4)bin, bin) [NOT RELEASED]
   call setreg(tvec, parflg)
   Set vvec from tvec and parflg.

SFR$SYS1 (ptr, bin) [NOT RELEASED]
   call sfrS$ys1(block_ptr, ercode);
   Frees Space From System Class Storage.

SFR_CFSC(fixed bin, ptr, fixed bin) [NOT RELEASED]
   call sfr_cFsc(storage_class, hcb_ptr, code);
   Completely free allocated storage for a level.

SFR_HP(fixed bin, ptr, ptr, fixed bin) [NOT RELEASED]
   call sfr_heap(storage_class, hcb_ptr, block_ptr, ercode);
   Free heap storage.

SGD$DL (bin, bin) II-4-82
   call sgdS$dl(segdir_unit, code);
   Delete an entry from a segment directory.

SGD$EX (bin, bin, bin) [NOT RELEASED]
   call sgdS$ex(unit, type, code);
   Check the existence of a segment directory entry.

SGD$OP (bin, bin, bin, bin, bin) returns (bin) II-4-84
   open_unit = sgdS$op(key, segunit, unit, type, code);
   Open a segment directory entry. Key = k$read, k$writ, k$rdwr, k$vmr.

SGD$SR$ (bin, bin, bit(16), bit(16), code) (svc = 1512) II-4-86
   call sgdS$r(key, file_unit, entry1, entry2, code);
   Position, read or modify a segment directory.

SGNLS$F (int*2(*), int*2, int*4, int*2, int*4, int*2, int*2) III-7-30
call sgnl$(condition_name, c_name_len, loc(stack_frame), sf_len, loc(aux_info), ai_len, flags);
    Signal a condition from FTN or F77.

SHARE$ (char(32)var, bin, bin, bin) [NOT RELEASED]
call share$(entryname, segment_no, access, code);
    Share a segment with specified access and file (privileged).

SHLIB (bin, (16)bin, bin) [NOT RELEASED]
    rtn_package_num = shlib(package_mnumber, ecb, code);
    Install shared library. Restricted.

SH_CMD (char(256)var, bin) [NOT RELEASED]
call sh_cmd(com_args, com_status);
    Process the SHUTDN command.

SIG$GT(fixed bin(15)) III-2-30
call sid$gt(sid);
    Get the spawner's id in a phantom process.

SIGNL$(char(*)var, ptr, bin, ptr, bin, bit(16)) III-7-32
call sign$(condition, addr(stack_frame), sf_len, addr(aux_info),
    ai_len, action);
    Signal a condition (PL1G, SPL, PLP or PMA).

SINFO$ (bin, bin, bin, bin) [NOT RELEASED]
call sinfo$(action, info_st, echo_st, code);
    Set and check values of INFO$STATUS for PRIME INFORMATION.

SLAVE((4200)bin, bin) [NOT RELEASED]
call slave(buf(1), wcix);
    Slave message handler (NPX).

SLAVER [NOT RELEASED]
call slaver;
    Root slave processor (NPX).

SLEEP$ (bin(31)) III-8-39
call sleep$(milliseconds);
    Suspend process.

SLEP$! (bin(31)) III-8-40
call sleep$(interval);
    An interruptable SLEEP$.

SMGST$(bin, char(*), bin, bin, char(*), bin, char(79), bin, (4+)bin) III-9-9
call smsg$(key, user_name, uname_len, user_num, system_name,
    system_name_len, message, message_len, error_vector);
    Send a message to another user. Key = 0 - deferred; 1 - immediate.

SMT_QFR (ptr) [NOT RELEASED]
call smt_qfr(smt_ptr);
    Unthread an entry from the smt_list for active EPFs.

SNA$CF (bin) [NOT RELEASED]
call sna$cf(code);
    Get dynamic Segments for SNA Server Wired and Unwired FS Classes.
SNA$CL (bin, bin) [NOT RELEASED]
call sna$cl(segnum, code);
Get Dynamic segments for LU6.2 free storage class.

SNA$CRFP (bin, bin, bin, bin, bin) [NOT RELEASED]
free_pool_id = sna$crfp(key, count, size, fs_class, code);
Create a free pool (interlude to crfp).

SNA$CRQS (bin, bin, ptr) returns(ptr) [NOT RELEASED]
event_qcb_ptr = sna$crqs(fs_class, length, semaphore);
Create event queue routine. Length must be \(2^{k-1}\).

SNA$CX (bin, bin) [NOT RELEASED]
call sna$cx(segnum, code);
Get Dynamic segments for PRIME/SNA RJE free storage class.

SNA$DEQA (ptr, bin, ptr, bin) returns(bin) [NOT RELEASED]
status = sna$deqa(lccb_addr, command, bha_ptr, qflag);
Dequeue a command or data block from either queue.

SNA$DEQE (ptr) returns(bin) [NOT RELEASED]
result = sna$deqe(event_qcb);
Dequeue from top of event queue.

SNA$DLQ (ptr, bin, bin) [NOT RELEASED]
call sna$dlq(qcb_ptr, free_storage_class, code);
Routine to delete a queue, returning it's storage to the free list.

SNA$ENQA (ptr, bin, ptr, bin) returns(bin) [NOT RELEASED]
status = sna$enqa(lccb_ptr, command, bha_ptr, qflag);
Enqueue a command or data block on the specified queue.

SNA$FLSH (bin, bin) [NOT RELEASED]
call sna$flsh(fs_class, code);
Flush free storage.

SNA$FREE (ptr) [NOT RELEASED]
call sna$free(block_ptr);
Return a block to its free pool.

SNA$GETB (ptr) returns(ptr) [NOT RELEASED]
BHA_ptr = sna$getb(fpid);
Unconditional get block from free pool.

SNA$GETC (ptr) returns(ptr) [NOT RELEASED]
BHA_ptr = sna$getc(fpid_threshold);
Conditional get block from free pool.

SNA$IADM (bin, bin, bin, char(*), bin, bin, bin) [NOT RELEASED]
call sna$iadm(log, trace, stats, stats_file, auto_stop,
stop_time, return_code);
Administration control request.

SNA$IAIN (char(*), ptr, ptr, bin) [NOT RELEASED]
call sna$iain(config_path, config_ptr, rem_sys_ptr, return_code);
Create and send a START.compareTo LECB to the LU Manager.
SNA$ICLS (bin) [NOT RELEASED]
call sna$icls(return_code);
  Close established Mate-Manager connection

SNA$IGD (char(*), bin, bin) [NOT RELEASED]
call sna$igd(dev_name, time, return_code);
  Build and send a GET_DEVICE LECB to the LU Manager.

SNA$IGE (ptr, bin, bin, bin) [NOT RELEASED]
call sna$ige(lecb_ptr, event_type, time_limit, return_code);
  Retrieve a message for a LU Mate from the LU Manager.

SNA$IOPN (bin) [NOT RELEASED]
call sna$iopn(code);
  Open connection between mate and manager.

SNA$IRD (bin(31), bin) [NOT RELEASED]
call sna$ird(device_id, return_code);
  Build and send a RETURN_DEVICE LECB to the LU Manager.

SNA$IRSD (bin(31), bin) [NOT RELEASED]
call sna$irsd(session_id, return_code);
  Build and send a RECOVER_SESSION LECB to the LU Manager.

SNA$IS (bin(31), char(*), ptr, bin) [NOT RELEASED]
call sna$ss(session_id, suspend_text, ssib_ptr, return_code);
  Build and send a SUSPEND_SESSION LECB to the LU Manager.

SNA$IST (bin, bin) [NOT RELEASED]
call sna$ist(status_type, return_code);
  Build and send a CHECK_STATUS LECB to the LU Manager.

SNA$ISTA (bin, char(*), bin) [NOT RELEASED]
call sna$ista(type, name, return_code);
  Administration status request.

SNA$ISTP (bin, char(*), bin) [NOT RELEASED]
call sna$istp(key, name, type, return_code);
  Administration stop request.

SNA$ISWR (bin(31), bin, ptr, bin, bin, bin) [NOT RELEASED]
call sna$iswr(sess_id, writeflag, bufptr, datalen, 
  vb versus, return_code);
  Build and send a WRITE_DATA LECB to the LU Manager.

SNA$SCLDL (ptr, bin, bin) [NOT RELEASED]
call sna$scdl(lccb_addr, stat1, stat2);
  Delete a logical connection for the IPQNM routines.

SNA$SLCIN (struc) [NOT RELEASED]
call sna$slcin(lcarray);
  Initialize a logical connection for the IPQNM routines.

SNA$NTFY (bin) [NOT RELEASED]
call sna$ntfy(user);
  Interlude to x$ntfy for SNA.
SNA$PH (char(*), char(*), bin, bin) [NOT RELEASED]
  call sna$ph(service_name, cpl_args, user_no, code);
  Create an SNA Service for an SNA Administrator.

SNAP$0 (char(32)var) returns(ptr) [NOT RELEASED]
  ecb_ptr = snap$0(name);
  Snap a dynamic link into ring zero (i.e. a gate).

SNCHE$ (bin, char(32)var, bin, bin) [NOT RELEASED]
  call snche$(keys, name, position, code);
  Check a system name for validity, return specific errors information.

SNCHK$ (bin, char(32)var) returns(bit(1)) [NOT RELEASED]
  name_ok = snchk$(key, name);
  Check a system name for validity.

SOR0$ (ptr) [NOT RELEASED]
  call sor0$ (cfh_ptr);
  Invoke the list of ring 0 static on-units.

SOR3$ (ptr) [NOT RELEASED]
  call sor3$ (cfh_ptr);
  Invoke list of ring 3 static on-units.

SOUR3_ (ptr) [NOT RELEASED]
  call sour3_ (list_ptr);
  Return pointer to the ring 3 static on-units.

SP$MGR (bin, char(32)var, struc, struc, bin(31), bin, bin) [NOT RELEASED]
  call sp$mgr(key, node, queue_entry, template, rqst_no,
              data_file_unit, code);
  Spool queue manager.

SPASS$ (char(6), char(6), bin) (svc = 1513) II-2-32
  call spass(owner_pw, non_owner_pw, code);
  Set passwords of current UFD.

SPAWS$ (1, 2 bit(13), 2 bit(1), 2 bit(1), 2 bit(1), ptr, char(32) var, bin, char(256) var, bin, bin)
  [NOT RELEASED]
  call spawn$ (key_structure, addr(spawn_data_struc), filename,
              unit, cpl_args, user_num, status);
  Spawn a process. Privileged.

SR$ABSDS, SR$ABS (char(128)var, char(32)var, bin) [NOT RELEASED]
  call sr$absd$s(rule, list, code);
  Absolutely disable an optional search rule.

SR$ADD, SR$ADB (ptr, char(128)var, char(128)var, bin) [NOT RELEASED]
  call sr$addb(arg_list_ptr, old_rule, new_rule, code);
  Add a search rule to a list before an existing rule.

SR$ADDE, SR$ADE (ptr, char(128)var, char(128)var, bin) [NOT RELEASED]
  call sr$adde(arg_list_ptr, old_rule, new_rule, code);
  Add a search rule to a list after an existing rule.

SR$CREATE, SR$CRE (char(32)var, ptr, bin) [NOT RELEASED]
  call sr$create(search_list_name, list_ptr, code);
  Create a search list by name and open it.
SR$DEL (char(32)var, bin) [NOT RELEASED]
call sr$del(search_list_name, code);
Delete an existing search rule.

SR$DSABL, SR$DSA (char(128)var, char(32)var, bin) [NOT RELEASED]
call sr$dsabl(rule, list, code);
Disable an optional search rule.

SR$ENABL, SR$ENA (char(128)var, char(32)var, bin) [NOT RELEASED]
call sr$enabl(rule, list, code);
Enable an optional search rule.

SR$EXSTR, SR$EXS (char(128)var, bin, char(32)var, bit(1)) returns (bit(1)) [NOT RELEASED]
rule_exists = sr$exstr(rule, req_type, list, case_sensitive);
Check a search list for the existence of a specific rule.

SR$FR_LS, SR$FRL (ptr) [NOT RELEASED]
call sr$fr ls(obj_ptr);
Free storage used by search rule.

SR$INIT, SR$INI (bin) [NOT RELEASED]
call sr$init(code);
Set search lists for all template files in the search rules directory.

SR$LIST, SR$LIS (ptr, bin) [NOT RELEASED]
call sr$list(arg_output_ptr, code);
Return a list of all search list names in this process.

SR$NEXT, SR$NEXT (ptr, ptr, char(128)var, ptr, char(128)var) returns(ptr) [NOT RELEASED]
next_ptr = sr$next(list_ptr, prev_rule_ptr, referencing_dir,
locator_ptr, search_place);
Fetch the next search rule from a given search list.

SR$OPEN (char(32)var, ptr, bin) [NOT RELEASED]
call sr$open(search_list_name, list_ptr, code);
Find search list specified by name and "open" it. Obsolete.

SR$READ, SR$REA (ptr, ptr, bin) [NOT RELEASED]
call sr$read(list_ptr, arg_output_ptr, code);
Return a list of all search rules of a given search list, printable.

SR$REM (ptr, char(128)var, bin) [NOT RELEASED]
call sr$rem(arg_list_ptr, the_rule, code);
Remove a search rule from a list.

SR$SET, SR$SET (ptr, ptr) [NOT RELEASED]
call sr$set1(rule_ptr, locator_ptr);
Set the locator value in a given search rule.

SR$SSR (char(128)var, char(32)var, bit(1), char(128)var, bin, bin) [NOT RELEASED]
call sr$ssr(template_path, list_name, overwrite, error_path,
error_line, code);
Set search rules from a template file.

SR$TEMPL (char(128)var, ptr, char(32)var, bit(1), bit(1), char(128)var, bin, bit(1), bin) [NOT RELEASED]
call sr$templ(template_file, list_ptr, real_list_name, 
    set_up_dflt, dflt_override, error_pathname, 
    error_line_number, rec_call, code);
Process a search list template file. Obsolete.

SR$UPDT (char(32)var, ptr, bin) [NOT RELEASED]
call sr$updt(arg_old_list_name, new_list_ptr, code);
Install (update) a new copy of a possibly existing search list. Obsolete.

SRCH$$ (bin, char(32), bin, bin, bin, bin) (svc = 1511) II-4-94
call srch$$ (key, file_name, file_name_len, file_unit, file_type, 
    code)
Open, close, delete or verify existence of a file. key = (K$READ, K$WRT, K$RDWR, 
    K$CLOS, K$DELE, K$EXST) + (K$UFD, K$ISEG, K$CACC, K$GETU) + (K$NSAM, 
    K$NDAM, K$NSGS, K$NSGD)

SRCH$CHT. See SEARCH_CASELESS_HASH_TABLE$.

SRCH$HTB. See SEARCH_HASH_TABLE$.

SRSFX$ (bin, char(*)var, bin, bin, bin, char(32)var, char(32)var, bin, bin) returns(bin(31)) II-4-103
    char_pos = srsfx$(key, path_name, file_unit, file_type, 
        num_suffixes, suffix_list, base_name, suffix_used, code)
Search for a file with any set of suffixes. Key same as SRCH$$.

SRWRREC (bin, bin, bin, bin, bin(31), bin, bin) [NOT RELEASED]
call srwrrec(key, phav, nww, nch, rel_addr, device_num, alt_rtn);
SVC handler for RREC, WREC SVC.

SS$ERR III-5-11
call ss$err;
Signal SUBSUS_ERR$ if not interactive.

ST$SGS returns(bin) III-4-26
    maximum_private_static_segs = st$sgs();
Return maximum number of static segments allowed for this user.

STD$CP (char(*) var, bin, bin, 1, 2 bit(1), 2 bit(1), 2 bit(14), ptr, ptr) [NOT RELEASED]
call std$cp (command_line, status, com_status, flags, 
    local_variable_ptr, rtn_function_ptr);
Standard command processor.

STKO$ [NOT RELEASED]
    CALF STKO$ /* PMA only
Stack overflow handler.

STK_EX (ptr) [NOT RELEASED]
call stk_ex (full_stack_ptr);
Automatic stack extender.

STPNC (ptr, bin, ptr, bin) returns(bin) [NOT RELEASED]
    status = stpnc(error_buffer, err_buf_size, trace_buffer, 
        zero_flag);
Routine to gather PNC statistics data.

STR$AL(bin, bin, bin, bin) returns(ptr) III-4-5
    block_ptr = str$al(storage_type, block_size, base wd, status);
Temporary storage allocator. Check for new calling sequence.
STR$AP (bin(31)) returns(ptr) III-4-7
  block_ptr = str$ap(block_size);
  // Process class storage allocator.

STR$AS(bin(31), bin) returns(ptr) III-4-8
  block_ptr = str$as(block_size, err_code);
  // Subsystem process class storage allocator.

STR$AU (bin(31)) returns(ptr) III-4-10
  block_ptr = str$au(block_size);
  // User program class storage allocator.

STR$FP (ptr) III-4-11
  call str$fp(block_ptr);
  // Frees space from process class storage.

STR$FR(bin, ptr, bin) III-4-12
  call str$frr(key, block_ptr, status);
  // Free allocated storage by STR$AL. Check for changed calling sequence.

STR$FS (ptr, bin) III-4-13
  call str$fs(block_ptr, bin);
  // Frees space from subsystem process class storage.

STR$FU (ptr) III-4-14
  call str$fu(block_ptr);
  // Frees space from user program class storage.

STR$BL (bin, ptr, bin) returns(bin) [NOT RELEASED]
  node_status = str$bl(my_node, target_buffer, zero_flag);
  // Routine to move the ring break information to a ring 3 buffer.

STUFF (ptr, bin, char(253) var, bin) [NOT RELEASED]
  call stuff(addr(msg), type, str, str_len);
  // Put subfield data into spare data field of a message.

SUSR$ returns(bit(1)) III-2-31
  is_user_l = susr$();
  // Returns whether or not caller is user 1.

SW$INT (bin, 1, 2 bin, 2 bit(16), 2 bit(16), 1, 2 bin, 2 bit(16), 2 bit(16), bin [, bin]) returns(bin)/*
  ring 0) [NOT RELEASED]
  call sw$int(key, selection, value, ercode [, outer_ring]);
  already_deferred = sw$int(key, selection, value, ercode
  [, outer_ring]);
  // Software interrupt enable control module. Key = K$ON, K$OFF, K$RDON, K$RDOF,
  K$READ, K$ALON, K$ALOF, K$RAON, K$RAOF, K$RDAL.

SW$ON (1, 2 fixed bin, 2 bit(16), 2 bit(16)) [NOT RELEASED]
  call sw$on(selection);
  // Turns on the specified software interrupts for ring 3.

SW$ROF, SW$ROA (1, 2 fixed bin, 2 bit(16), 2 bit(16)) [NOT RELEASED]
  call sw$rof(value);
  // Reads and then turn off all present interrupts for ring 3.

SW$RST [NOT RELEASED]
call sw$rst;
Reset ring 0 software interrupt enable mechanism.

SWFBK   [NOT RELEASED]
call swfbk;
Invoke QUIT condition in ring 3 with pb backup.

SWFM  [NOT RELEASED]
call swfm;
Invoke QUIT condition in ring 3.

SYNSCHK (bin, bin, bin, bin) [NOT RELEASED]
call sy$schk(SyncNum, NumberOfNotices, NumberOfWaiters, Status);
Returns the number of outstanding notices or number of waiters on an event synchronizer.

SYNSCREA (bin, bin, bin) [NOT RELEASED]
call sy$crea(InitialNoticeCount, SyncNum, Status);
Create an event synchronizer for this server.

SYNSDEST (bin, bin) [NOT RELEASED]
call sy$destr(SyncNum, Status);
Destroy a synchronizer belonging to this server.

SYNSGCHK (bin, bin, bin, bin, bin) [NOT RELEASED]
call sy$ghck(GroupName, PriorityLevel, NumberOfNotices,
NumberOfWaiters, Status);
Returns the number of outstanding notices or number of waiters on an event group.

SYNSGCRE (bin, bin, bin) [NOT RELEASED]
call sy$gcre(PriorityLevels, GroupNum, Status);
Create an event group for this server.

SYNSGDST (bin, bin) [NOT RELEASED]
call sy$gdst(GroupName, Status);
Destroy the event group after first removing any event synchronizers from the group.

SYNSGLST (bin, (*))bin, bin, bin) [NOT RELEASED]
call sy$glst(GroupName, GroupList, GroupCount, Status);
Returns the numbers of the event groups belonging to this server (process).

SYNSGRTR (bin, bin, bin, bin, ptr, bin) [NOT RELEASED]
call sy$grtr(GroupName, PriorityLevel, WhatHappened,
SyncNum, ForClientUse, Status);
Retrieve a notice from an event group if at least one has been posted.

SYNSGWT (bin, bin, bin, ptr, bin) [NOT RELEASED]
call sy$gwt(GroupName, WaitTime, WhatHappened,
SyncNum, ForClientUse, Status);
Timed wait for a notice to be posted to an event group.

SYNSINFO (bin, ptr, bin) [NOT RELEASED]
call sy$info(SyncNum, SyncInfoPtr, Status);
Wait on an event group until a notice has been posted to it.

6-58
SYNLIST (bin, (')bin, bin, bin) [NOT RELEASED]
call syn$list(SyncListSize, SyncList, SyncCount, Status);
List event synchronizers belonging to this server (process).

SYNSLSIG (bin, bin, (')bin, bin, bin) [NOT RELEASED]
call syn$lsig(Num, SyncListSize, SyncList, SyncCount,
             Status);
Returns a list of the synchronizers currently in an event group.

SYNSMVTO (bin, bin, bin, ptr, bin) [NOT RELEASED]
call syn$mvto(Num, SyncNum, PriorityLevel, ForClientUse,
              Status);
Move an event synchronizer into an already existing event group.

SYN$POST (bin, bin) [NOT RELEASED]
call syn$post(SyncNum, Status);
Post a notice to an event synchronizer.

SYNSREMV (bin, bin) [NOT RELEASED]
call syn$remv(SyncNum, Status);
Remove a synchronizer from whatever group it is in.

SYNRTRV (bin, bin, bin) [NOT RELEASED]
call syn$trv(SyncNum, WhatHappened, Status);
Retrieve a notice on an event synchronizer if at least one has been posted.

SYNSMWT (bin, bin(31), bin, bin) [NOT RELEASED]
call syn$mwt(SyncNum, WaitTime, WhatHappened, Status);
Timed wait on an event synchronizer.

SYNSWAIT (bin, bin) [NOT RELEASED]
call syn$wait(SyncNum, Status);
Wait on an event synchronizer until a notice is returned.

T$AML C (bin, ptr, bin, bin, (2)bin [, bin, bin]) (svc = 0513) IV-8-23
   call t$amlc(line, addr(buffer), buf_char_count, key, status_vec,
              [buf_start_char, code]);
Communicate with AMLC driver. See Subroutine Ref Guide for keys.

T$CMPC (bin, ptr, bin, bin, (2)bin) (svc = 0512) IV-7-28
   call t$cmpe(unit, addr(buffer), num_words, inst, status);
Input from MPC card reader.

T$GPPI (bin, bin, bin, bin, (4096)bin, bin) [NOT RELEASED]
call t$gppei(unit, key, data1, data2, array, code);
General purpose parallel interface routine.

T$GS (bin, bin, bin, ptr, bin, bin) [NOT RELEASED]
call t$gs(unit, key, function, buffer, buf_len, non_std_code);
Driver for Vector General graphics terminals.

T$LMPC (bin, ptr, bin, bin, (2)bin) (svc = 0511) IV-7-6
   call t$lmpe(unit, addr(buffer), num_words, inst, status);
Move data to MPC line printer.

T$MG (bin, bin, bin, ptr, bin, (3)bin) [NOT RELEASED]
call t$mg(unit, key, aux_data, addr(buffer), buf_len, stat_vec);
Driver for SOC-Megaphoric 7000 interface.
T$MT (bin, ptr, bin, bin, (2)bin) (svc = 0510) IV-7-37
  call t$mt(unit, addr(buffer), num_words, inst, status);
  Raw data mover for tape drive.

T$PMP C (bin, ptr, bin, bin, (2)bin) (svc = 0515) IV-7-34
  call t$pmpc(unit, addr(buffer), num_words, inst, status);
  Raw data mover for card reader.

T$SLC1 (bin, bin, ptr, bin) [NOT RELEASED]
  call t$slc1(key, line, addr(block), block_len);
  Control block interpreter for HSSMLC, MDLC, and LYNX controllers.

T$VG (bin, ptr, bin, bin, (2) bin) (svc = 0514) IV-7-16
  call t$vg(phys_unit, addr(buffer), num_words, inst, status)
  Interface to Versatec printer.

TAS (char(*) var, bin, bin, char(32), bin, bit(16), bin) returns(bin) [NOT RELEASED]
  outc = tas(line, state, key, entry_name, entry_name_length,
            attach_switch, code);
  Attach to directory. Obsolete; use AT$.

TERM$ (bin) [NOT RELEASED]
  call term$(key)
  SET/reset terminal parameters for use with the INFORMATION product. Key = 1 (enter
  INFO), 2 (leave INFO). Obsolete.

TEXTOS (char(32), bin, bin, bin) III-10-15
  call textos(filename, file_name_len, actual_len, text_ok)
  Check a filename for valid format. Text_ok is a fortran logical.

TISMSG (bin, bin, bin, bin) III-2-32
  call tismsg (user, connect_minutes, cpu_seconds, io_seconds);
  Print accumulated time message (for logout message).

TIMDAT (1, 2 (3)char(2), 2 (9)bin, 2 char(32), bin) (svc = 0502) III-2-34
  call timdat(time_date_stuff, time_date_stuff_len)
  Return system and user information.

TL$SGS returns(bin) III-4-27
  max_segn in_dtar2 = tl$sgs();
  Return highest segment number allowed in dtar 2.

TM3270 ((3) bin, (3) bin, bin) [NOT RELEASED]
  call tm3270 (delays, polling_periods, code);
  Initiate the Traffic Manager for IBM 3270 terminals. (DPTX)

TMRSCANL (bin, bit(1), bin) [NOT RELEASED]
  PROCEDURE tmrscanl(Timer: TimerNumber;
    VAR Expired: plp_boolean; VAR Status: TimerStatusCode);
  Cancel the pending timer identified by Timer. (Timer)

TMR$CREA (bin, bin, bin) [NOT RELEASED]
  PROCEDURE tmr$crea(WhichKind: KindOfTimer;
    VAR NewTimer: TimerNumber; VAR Status: TimerStatusCode);
  Create a timer private to the calling server. (Timer)

TMR$DEST (bin, bit(1), bin) [NOT RELEASED]
PROCEDURE tmr$dest(Timer: TimerNumber;
    VAR Expired: plp_boolean; VAR Status: TimerStatusCode);
Destroy a timer. (Timer)

TMRSSINF, TMRS$NF (struc) [NOT RELEASED]
PROCEDURE tmr$sinf(VAR CurrentTimeInfo: PermTimeInfo);
Returns the PermTimeInfo. (Timer)

TMRSS$TIM, TMRSS$GT (struc) [NOT RELEASED]
PROCEDURE tmr$gtim(VAR CurrentTime: AbsoluteTime);
    SystemTime is returned in Universal Time. (Timer)

TMRSS$GMR (bin, struc, bin) [NOT RELEASED]
PROCEDURE tmr$gmr(Timer: TimerNumber; VAR Info: TimerInfo;
    VAR Status: TimerStatusCode);
Returns information on the timer specified. (Timer)

TMRSS$LIST (bin, (0:15)bin, bin, bin) [NOT RELEASED]
PROCEDURE tmr$list(TimerListSize: SHORT_CARDINAL;
    VAR TimerList: TimerListArray;
    VAR NumberOfTimes: SHORT_CARDINAL;
    VAR Status: TimerStatusCode);
Returns the timer numbers belonging to this server in TimerList. (Timer)

TMRSSLOCALCONV (struc, struc) [NOT RELEASED]
PROCEDURE TMRSSLocalConvert(LocalTime: LocTime;
    VAR UnivTime: CARDINAL_64);
    Converts the local time provided to Universal Time.

TMRSS$PROC [NOT RELEASED]
call tmr$proc;
The timer process. (TimerMDK)

TMRSS$SABS (bin, bin, struc, bit(1), bin) [NOT RELEASED]
PROCEDURE tmr$sabs(Timer: TimerNumber; Sync: EventSyncNumber;
    ExpirationTime: AbsoluteTime; VAR Expired: plp_boolean;
    VAR Status: TimerStatusCode);
Sets the timer to expire at the absolute time specified. (Timer)

TMRSS$SINT (bin, bin, bin(31), bit(1), bin) [NOT RELEASED]
PROCEDURE tmr$sint(Timer: TimerNumber;
    Sync: EventSyncNumber; ExpirationInterval: IntervalTime;
    VAR Expired: plp_boolean; VAR Status: TimerStatusCode);
Sets the timer to expire after the interval specified. (Timer)

TMRSS$SREP (bin, bin, bin(31), bin) [NOT RELEASED]
PROCEDURE tmr$srep(Timer: TimerNumber; Sync: EventSyncNumber;
    ExpirationIntervals: IntervalTime; VAR Status: TimerStatusCode);
Sets a repetitive timer to expire every ExpirationIntervals. (Timer)

TMRSS$STI (char(*)var, bin) [NOT RELEASED]
PROCEDURE TMRSS$STI (zline: ComLineString;
    VAR status: TimerStatusCode);
    Implements the SET_TIME_INFO operator command. (TimerMDK)
TMRSSTIM (struc, bin) [NOT RELEASED]
PROCEDURE tmrsstim(NewSysTime: AbsoluteTime;
                   VAR Status: TimerStatusCode);
Sets the system time. Changes will not affect interval timers. (Timer)

TMRSUNIVCONVERT (struc, struc) [NOT RELEASED]
PROCEDURE TMRSUnivConvert(UnivTime: CARDINAL_64;
                          VAR LocTime: LocTime);
Converts the Universal time value, UnivTime, to local time in LocTime format.
(TimeLibrary)

TNCHK$ (bin, char(128)var) returns(bit(1)) II-4-109
path_name_ok = tnchk$ (key, path_name)
Check pathname for valid format. Key = K$UPRC, K$WLDC, K$NULL.

TNOU (char(‘)‘), bin (svc = 0702) III-3-40
    call tnou(string, string_size);
Output characters and newline to terminal.

TNOUA (char(‘)‘), bin (svc = 0703) III-3-41
    call tnoua(string, string_size);
Output characters to terminal.

TP$CON (bin) [NOT RELEASED]
    call tp$con(code);
Reconnect user process to a terminal line.

TP$DIS (bin) [NOT RELEASED]
    call tp$dis(code);
Disconnect the terminal from this process making it assignable.

TRNRCV (bin, bin, bin, bin, bin) [NOT RELEASED]
    call trnrcv(key, vcix, mitlan, buffer, code);
Transmits and receives messages between master and slave processes.

TSRC$$ (bin, char(128), bin, bin, bin) II-A-17
    call tsrc$$ (key, path_name, file_unit, chr_pos, type, code)
Open, close, delete or find file. (Obsolete; use SRSFX$)

TTY$CNT returns(bin) [NOT RELEASED]
    num_chars = tty$cnt();
    Ring 3 interlude for Tf$cnt - returns # of characters in user's IRB.

TTY$IN returns(bit(1)) III-3-63
    characters_waiting = tty$in();
    Check if there are any characters in the tty input buffer for user.

TTY$RS (bit(16), bin) III-3-65
    call tty$rs(key, code);
    Routine to clear a process's I/O buffers. Key: bit 1 - output buffer; bit 2 - input buffer.

U$TERM (bit(1)) returns(bit(1)) [NOT RELEASED]
    previous_state = u$term(enable_terminal_output);
    Enable/disable terminal output from a child process.

UID$BT (bit(48) aligned) III-6-39
    call uid$bt (unique_bit_string);
    Return unique bit string.
UID$CH (bit (48) aligned, char (13)) III-6-40
   call uid$ch (unique_bit_string, character_string);
   Return a unique character sequence based on a unique bit string.

UNITS$ (bin, bin) II-4-112
   call units$ (num_unit, max_unit);
   Get the current unit number bounds.

UNLKF$ [NOT RELEASED]
   call unlkf$;
   Unlock all N1 locks owned by the calling process.

UNO$GT((128) bin, bin, bin) III-2-36
   call uno$gt (ids, lenids, numids);
   Return all ids for the current user.

UNWND$ (label) returns (bit(1)) [NOT RELEASED]
   unwind_ok = unwnd$ (target_of_nl_goto);
   Prepare the stack for nonlocal-goto-induced unwinding.

UPDATE (bin, bin) [NOT RELEASED]
   call update (key, 0);
   Update current UFD (Primos II). Key = 1.

USER$ (bin, bin) III-2-15
   call user$ (current_user_num, num_users);
   Return process number and total user count.

USNMT$ (bit(16), char(256) var, bin) [NOT RELEASED]
   call usnmt$ (no_margs, user_unassign_cmd_line, return_status);
   Unassigns magnetic tape drive. (DOSSUB only)

USRAS$ (char(256) var, fixed bin) [NOT RELEASED]
   call usras$ (com_args, com_status);
   Process USRASR command.

UTYPE$ (bin) III-2-38
   call utype$ (user_type);
   Return type of current process.

VALID$ (char(32) var, bin) returns (bit(1)) III-2-41
   id_found = valid$ (name, code);
   Validates name passed vs. user's composite ID (user ID plus groups).

VINIT$ (bin, bin, (*)bin, (*)bin, (*)bin, (*)bin, bin) [NOT RELEASED]
   call vinit$ (key, unit, segment_numbers, number_of_segments, window, access, segment_length, code);
   Map in a DAM file using initial VMFA.

WARM$ (ptr, bin) [NOT RELEASED]
   call warm$ (data_ptr, code);
   Handle warm start setup for INFORMATION.

WBK$ (bin, bin(31), ptr, bin, bin) [NOT RELEASED]
   call wbk$ (unit, logical_block, buffer_ptr, num_words, code);
   Logical Block I/O block write routine.
WILDS$ (char(32)var, char(32)var, bin) returns (bit(1)aligned) II-4-113
    match = wild$ (wildcard_name, entry_name, code);
    Compare entry_name against wildcard_name for containment.

WRECLS$ (1, 2 bit(1), 2 bit(1), 2 bit(1), 2 bit(1), 2 bit(1), 2 bit(1), 2 bit(1), 2 bit(1),
    2 bit(1), (3)ptr, (3)bin, bin(31), bit(16), bin) [NOT RELEASED]
    call wrec$l$ (nch, buf_ptr, buf_len, rec_adr, pdev, code);
    Write record to assigned disk.

WRL$ (ptr, fixed bin) [NOT RELEASED]
    call wrl$ (list_ptr, entries);
    Return a pointer to the caller's list of static on-units.

WTLIN$ (bin, char(*), bin, bin) (svc = 1526) II-4-115
    call wtlin$ (file_unit, buffer, buffer_len, code)
    Write a given number of ASCII chars. Buffer_len is in words.

X$ASGN (bin, bin, bin) P-14-4
    call x$asgn (subprocess, count, code);
    Assign primitive for general users.

X$CLRA P-14-23
    call x$clra;
    Routine that can be used to clear all connections a user owns.

X$FRPL (bin, ptr, bin) [NOT RELEASED]
    call x$frpl (version, buffer_ptr, status_code);
    Gathers size information for all Primenet free pools.

X$GVVC (bin, bin, bin) P-14-25
    call x$gvvc (vcid, user, code);
    Pass control of a virtual circuit to another user.

X$LHCS (bin, ptr, bin) [NOT RELEASED]
    call x$lhcs (version, buffer_ptr, status_code);
    Gathers traffic information for Ethernet Primenet.

X$LTRC (bin, ptr, bin) [NOT RELEASED]
    call x$ltrc (version, buffer_ptr, status_code);
    Gathers traffic information for Ethernet Primenet nodes.

X$PRTQ (bin, ptr, bin) [NOT RELEASED]
    call x$prtq (version, buffer_ptr, status_code);
    Gathers length information concerning the Primenet protocol queues.

X$RCV (bin, char(*), bin, bin) P-14-18
    call x$rcv (vcid, buffer, buffer_size, state);
    Provide receive buffers for X.25 packet input.

X$RSET (bin, bin, bin) [NOT RELEASED]
    call x$rset (vcid, why, status)
    Allow a user to cause a reset on one of his virtual circuits.

X$RT (bin, bin, char(32)var, char(32)var, bin, bin, bin, char(32)var, bin) [NOT RELEASED]
    call x$rt (key, option_key, src_item, dest_item, path,
        ret_rt_class, ret_path, ret_item, code);
    Ring 0 support for route through configuration information. Key = xk$nam, xk$adr (only for
    xr$name). Option_key = xr$me, xr$name, xr$path, xr$scradr.
X$RTI returns(bit(1)) [NOT RELEASED]
    made-it = x$rti();
    Set up this process to run as the route-through server.

X$STAT (bin, bin, (255)bin, bin, (255)bin, bin, bin, bin) P-14-29
    call x$stat(key, array_or_vcn, array1, ar1_len, array2, ar2_len,
        code, time);
    Routine to return status information to user space.

X$TRAN (bin, bin, char(*), bin, bin) P-14-16
    call x$tran(vcid, buffer_type, buffer, buffer_count, state);
    X.25 transmit primitive.

X$UASN (bin) P-14-23
    call x$uasn(subprocess);
    Unassign primitive for general users.

X$VCLT (bin, bin, 1, 2 bin, 2 bin, 2 bin, 2 (*)bin, bin) [NOT RELEASED]
    call x$vclt(user_id, vc_list_size, vc_list, error_code);
    Return a list of a user's active VCs.

X$WAIT (bin) returns(bin) P-14-24
    timer_expired = x$wait(tenths);
    Timed wait for network event.

X$LACP (bin, bin, char(*), bin, char(4), bin, char(*), bin, bin) P-14-14
    call x$lacpt(key, vcid, fclty, fclty_n, prid, prid_n,
        udata, udatan, state);
    Accept pending x.25 connection. Key = 1, xk$mdb, xk$ctl, xk$sve.

X$LASGN (bin, char(16)var, char(16)var, char(4)var, char(32)var, bin, bin, bin, char(16)var,
    char(16)var, bin, bin) [NOT RELEASED]
    call x$lasgn(key, adr, subadr, prid, udata, port, gfi, vcn,
        src_adr, src_sadr, count, code);
    Extended declaration of interest in incoming calls.

X$LCLR (bin, bin, char(*), bin, char(*), bin, bin, bin) [NOT RELEASED]
    call x$lclr(key, vcid, why, fclty, fclty_n, udata, udatan, xtra3, state);

X$LCLR (bin) [NOT RELEASED]
    call x$lclr(key);
    Clear either 'USER' or 'SYSTEM' VCs. Key = xk$suv, xk$sve.

X$LCONN (bin, bin, char(*), bin, char(*), bin, char(4), bin, char(*), bin, (2)bin [], char(*) bin,
    bin, bin, char(*), bin, char(*), bin) P-14-6
    call x$lconn(key, vcid, port, addr, addr_len, fclty, fclty_n, pr_id, pr_id_n,
        udata, udata_n, state[],
        rtn_utdata, rtn_utdata_len, r_u_rtn_cnt, more_key,
        src_addr, src_sadr_len, pnet, pnet_len);
    Request a virtual circuit connection. Key = (xk$sany, xk$sie, xk$syn, xk$syn, xk$sng, xk$snpn) +
        (xk$sctl, xk$smbd) + (xk$sadr, xk$nam) + [xk$sam, xk$srig] + [xk$srd]. (X.25)

X$LGAS (bin, bin, bin, bin, bin, char(15), bin, bin, char(15), bin, bin, char(4), bin, bin, char(*),
    bin, bin, bin) [NOT RELEASED]
call xlgas$ (key, vcid, port, gfi, vcn, cmnd,
    faddr, faddr, faddr, taddr, taddr, taddr, taddr, taddr,
fcty, fctyn, fctyl, prid, pridm, prid,
    udata, udatam, udatal, state);
Get all of the fields in a call accept packet. Key is ignored.

XLGC$ (bin, bin, bin, bin, bin, bin, (8)bin, bin, (8)bin, bin, bin, (32)bin, bin, bin, (2)bin, bin,
    bin, (63)bin, bin, bin, (2)bin) [NOT RELEASED]
call xlgcs$ (key, vcid, port, gfi, vcn, cmnd, faddr, faddr,
    faddr, taddr, taddr, taddr, taddr, fcty, fctyn, fctyl,
    prid, pridm, prid, udata, udatam, udatal, state);
Get all of the fields in a connect request packet. Key = 0, xk$reg.

XLGCN (bin, bin, bin, char(\t), bin, bin, char(\t), bin, bin, char(\t), bin, bin, bin, bin)
P-14-11
call xlgcn (key, vcid, port, saddr, saddr, saddr,
    fcty, fctyn, fctyl, prid, pridm, pridbc,
    udata, udatam, udatabc, state);
Get information about pending call requests. Key = xk$nam, xk$sadr.

XLGI$ (bin, bin, bin, bin, bin, (*)bin, bin, bin, (*)bin, bin, bin, (*)bin, bin, bin, (*)bin, bin, bin,
    (*)bin, bin, bin, bin) [NOT RELEASED]
call xlgis$ (key, vcid, port, GFI, VCN, command, calling_addr,
calling_addr_len, calling_addr_rtn_len, called_addr,
called_addr_len, called_addr_rtn_len, facilities,
facil_len, facil_rtn_len, proto_id, prid_len,
prid_rtn_len, user_data, user_data_len, user_data_rtn_len,
result_state)
Get all of the fields in an extended CLEAR INDICATION.

XLGVVC (bin, bin, bin, (8)bin, bin, bin, (8)bin, bin, (8)bin, bin, bin, (32)bin, bin, (2)bin, bin, (62)bin,
    bin, bin) P-14-25
call xlgvvc (key, vcid, user, addr, admn, port, faddr, faddr, taddr,
taddr, fcty, fctyl, prid, pridm, udata, udatan, code)
Pass control of a virtual circuit to another user. Key = xk$usr, xk$port, xk$sadr.

XLUASN (bin, char(16)var, char(16)var, char(4)var, char(32)var, bin, bin, bin, char(16)var,
    char(16)var, bin) [NOT RELEASED]
call xluasn (key, addr, saddr, prid, udata, port, gfi, vcn, src_addr,
    src_saddr, code);
Unassign an extended declaration.

XMTRCV (bin, char(8), bin(31), (*)bin, bin, bin) [NOT RELEASED]
call xmtrc$ (caller_key, slave_id, xmit_len, buffer, time, rcode);
Transmits and receives messages to and from slaves in one operation under quitter
protection.

6.2. Spool library
Spool routines are in the shared spool library or (at 21.0) SPOOL_LIBRARY.RUN.

SPOOL$ (bin, char(\t), bin, (29)bin, (*)bin, bin, bin)
call spool$ (key, filename, namelen, info, buffer, buflen, code);
Insert a file in spooler queue.
6.3. Application Library

Binary routines are in NVAPPLB.BIN; dynts in VAPPLB.BIN; runtime library is APPLICATIONS_LIBRARY.RUN. Mainly used in FORTRAN programs. It is recommended that the appropriate system routines be used instead of application library routines where possible. R-mode binaries are found in APPLIB.BIN.

CASE$A (int*2, char*, int*2) returns logical
   [valid_length =] case$A (key, string, length)
   Converts case from lower to upper or upper to lower. Key = A$FUPP, A$FLOW.

CLOSS$A (int*2) returns logical
   [closed_ok =] clos$A (file_unit)
   Closes the file open on file_unit.

CMDSL$A (int*2, int*2(*), int*2, char*, int*2, int*4, int*2) returns logical
   [command_ok =] cmdsl$A (key, kwlist, kwindx, optbf, bfin, opt,
   val, kwinfo)
   Parses a command line.

CNVASA (int*2, char*, int*2, int*4) returns logical
   [conversion_ok =] cnvas$a (numkey, name, namlen, val)
   Convert an ASCII digit string to its binary value for octal, decimal and hex. numkey = (A$DEC, A$BIN, A$OCT, A$HEX)

CNVB$A (int*2, char*, int*2, int*4) returns int*2
   [int_2_val =] cnvb$A (numkey, val, name, namlen)
   Convert a binary number to an ASCII string.

CSTR$A (char*, int*2, char*, int*2) returns logical
   strings_equal = cstr$A (astring, alen, bstring, blen)
   Compares two strings for equality.

CSUB$A (char*, int*2, int*2, char*, int*2, int*2, int*2) returns logical
   substrings_match = csub$A (a, alen, afc, alc, b, blen, bfc, blc)
   Compare two substrings for equality.

CTIM$A (int*4) returns real*8
   seconds = ctim$A (cputim_in_centiseconds)
   Returns elapsed CPU time.

DATE$A (char*) returns real*8
   mm_dd_yy = date$A (date)
   Returns today's date.

DELE$A (char*, int*2) returns logical
   success = dele$A (name, namlen)
   Delete a file.

DOFY$A (char*) returns real*8
   yr_dd = dofy$A (dofy)
   Returns the day of the year (DDD).

DTIM$A (int*4) returns real*8
   time_in_seconds = dtim$A (disktim)
   Returns disk time in centiseconds.

EDAT$A (char*) returns real*8
dd mm yy = edat$ (edate)

Returns date in European (military) form.

ENCD$A (char*, int*2, int*2, real*8) returns logical

success = encd$A (array, width, dec, val)

Encodes a real number into a string in Fwidth.dec format.

EXST$A (char*, int*2) returns logical

exists = exst$A (name, namlen)

Indicates whether a file exists.

FDAT$A (int*2, char*) returns real*8

mm dd yy = fdat$A (datemod, date)

Converts file date to string.

FEDT$A (int*2, char*) returns real*8

mm dd yy = edt$A (datemod, date)

Converts file date to string, European style.

FILL$A (char*, int*2, char*1)

CALL FILL$A (name, namlen, char)

Fill a buffer with char.

FSUB$A (char*, int*2, int*2, int*2, char*1) returns logical

success = fsub$A (string, len, fchar, lchar, filchar)

Fills a substring with a character.

FTIM$A (int*2, char*) returns real

realtimemod = ftim$A (timemod, time)

Converts a file time to string or real.

GCHR$A (char*, int*2) returns int

character = gchr$A (farray, fchar)

Extracts a character from a string.

GEND$A (int*2) returns logical

success = gend$A (unit)

Position a file to EOF.

JSTR$A (int*2, char*, int*2) returns logical

success = jstr$A (key, string, len)

Justify a string (left, right, or center). Key = (A$RIGHT, A$LEFT, A$CNTR).

LSTR$A (char*, int*2, char*, int*2, int*2, int*2) returns logical

found = lstr$A (a, alen, b, blen, fcp, lcp)

Locates one string within another.

LSUB$A (char*, int*2, int*2, char*, int*2, int*2, int*2, int*2, int*2) returns logical

found = lsub$A (a, alen, afc, alc, b, blen, bfc, blc, fcp, lcp)

Locates one substring within another.

LTOK$A (char*, int*2, int*2, char*, int*2, int*2, int*2, int*2, int*2) returns logical

found = ltok$A (a, alen, afc, alc, b, blen, bfc, blc, hcp, lcp, ndel)

Locates character substrings as tokens. (V-mode only)

LWSC$A (char*, int*2, int*2)
call lwcs$ (string, position, length)
Translates a substring to lowercase. (V-mode only)

MCHR$ (char*, int*2, char*, int*2) returns int
char moved = mchr$ (tarzey, tchar, farzey, fchar)
Moves a character from one array to another array.

MOVE$ (char*, int*2, char*, int*2, int*2)
call move$ (str, fpos, tstr, tpos, len)
Move a string to another. (V-mode only)

MSTR$ (char*, int*2, char*, int*2) returns int
number moved = mstr$ (astring, alen, bstring, blen)
Move a string to another.

MSUB$ (char*, int*2, int*2, char*, int*2, int*2, int*2) returns int
number moved = msub$ (a, alen, afc, alc, b, blen, bfc, blc)
Move a substring to another.

NLEN$ (char*, int*2) returns int*2
length = nlen$ (name, namlen)
Returns actual length of the string.

OPEN$ (int*2, char*, int*2, int*2) returns logical
success = open$ (opnkey+typkey+untkey, name, namlen, unit)
Opens a file.

OPVP$ (char*, int*2, int*2, char*, int*2, int*2, int*2, int*2) returns logical
success = opvp$ (msg, msglen, opnkey+typkey+untkey, name, namlen, unit, wtime, retries)
Prompts a user for a file name and opens it with retries and verification.

OPNP$ (char*, int*2, int*2, char*, int*2, int*2) returns logical
success = opnp$ (msg, msglen, opnkey+typkey+untkey, name, namlen, unit)
Prompts user for a file name and opens it.

OPNV$ (int*2, char*, int*2, int*2, int*2, int*2) returns logical
success = opnv$ (opnkey+typkey+untkey, name, namlen, unit, verkey, wtime, retries)
Opens a file with retries and verification.

POSN$ (int*2, int*2, int*4) returns logical
success = posn$ (poskey, unit, pos)
Positions in a file.

RAND$ (int*4) returns real
random number = rand$ (seed)
Generates a pseudo-random number.

RNAM$ (char*, int*2, int*2, char*, int*2) returns logical
success = rnam$ (msg, msglen, namkey, name, namlen)
Prompts user for a name.

RNDI$ (int*4) real
random number = rndi$ (seed)
Initializes the random number generator.
PRNUM$A (char*, int2, int2, int4) returns logical
success = rnum$A (msg, msglen, numkey, val)
  Prompts user for a number and returns it.

RPOS$A (int2, int4) returns logical
success = rpos$A (unit, pos)
  Returns the absolute position of a file.

RSTR$A (char*, int2, int2) returns logical
success = rstr$A (string, len, cnt)
  Rotates a string.

RSUB$A (char*, int2, int2, int2, int2) returns logical
success = rsub$A (string, len, fchar, lchar, cnt)
  Rotates a substring.

RWND$A (int2) returns logical
success = rwnd$A (unit)
  Rewinds a file.

SSTR$A (char*, int2, int2, int2) returns logical
success = sstr$A (string, len, cnt, filchr)
  Shifts a string.

SSUB$A (char*, int2, int2, int2, int2, int2) returns logical
success = sssub$A (string, len, fchar, lchar, cnt, filchar)
  Shifts a substring.

TEMP$A (int2, char*, int2, int2) returns logical
success = temp$A (typkey+untkey, name, namlen, unit)
  Creates a temporary file and opens it.

TIME$A (char*) returns real
real time = time$A (time)
  Returns the time of day.

TREES$A (char*, int2, int2, int2) returns logical
is a treename = trees$A (name, namlen, fstart, flen)
  Checks a treename for validity.

TRNC$A (int2) returns logical
success = trnc$A (unit)
  Truncates a file at its current position.

TSCN$A (int2, int2(*), int2(*), int2, int2, int2, int2) returns logical
success = tscn$A (key, units, entry, maxix, entsix, maxlev,
lev, code)
  Scans a tree.

TYPE$A (int2, char*, int2) returns logical
is valid = type$A (key, string, len)
  Checks a string for being a valid type.

UNIT$A (int2) returns logical
unit_open = unit$A (unit)
  Determines if a file unit is open.
UPC\texttt{\$A} (char*, int*2, int*2)
\hspace{1cm} call upc\texttt{\$a} (string, position, length)
\hspace{1cm} Translates a substring to uppercase. (V-mode only)

YSNO\texttt{\$A} (char*, int*2, int*2) returns logical
\hspace{1cm} answer_is_yes = ysnos\texttt{\$a} (msg, msglen, defkey)
\hspace{1cm} Prompts a user and returns true if answer is yes.

6.4. DBMS routines

PRISAM routines.

Z\$ABRT (bin, bin)
\hspace{1cm} call z\$abrt (user\_transid, scode);
\hspace{1cm} Abort an active transaction, and remove any updates from the file.

Z\$CLOS (bin, bin)
\hspace{1cm} call z\$clos uniq\_file\_id, code);
\hspace{1cm} Close an open PRISAM file.

Z\$DDL (bin, pointer, bin)
\hspace{1cm} call z\$ddl (file\_id, info\_ptr, scode);
\hspace{1cm} Return DDL information. (Data structure layout subject to change without notice - this
\hspace{1cm} routine intended for DISCOVER support).

Z\$DELE (bin, bin)
\hspace{1cm} call z\$dele (file\_id, scode);
\hspace{1cm} Delete the current record of a PRISAM file.

Z\$ENDT (bin, bin)
\hspace{1cm} call z\$endt (user\_transid, scode);
\hspace{1cm} End an active transaction, and commit any updates to the file.

Z\$FIND (bin, bin, char\texttt{/}, bin, bin, char\texttt{/}, bin, bin, bin)
\hspace{1cm} call z\$find (uniq\_file\_id, funct, key\_buff, key\_len, key\_num,
\hspace{1cm} \hspace{1cm} found\_key\_buff, found\_key\_len, reserved, code);
\hspace{1cm} Find (not read) a record in a PRISAM file and make it the current record. funct = P\$FST,
\hspace{1cm} \hspace{1cm} P\$LST, P\$EQU, P\$GRT, P\$GRE.

Z\$INSR (bin, char\texttt{/}, bin, bin(31), bin, bin)
\hspace{1cm} call z\$insr (uniq\_file\_id, rec\_buff, rec\_len, rec\_num, reserved,
\hspace{1cm} \hspace{1cm} code);
\hspace{1cm} Insert a new record into a PRISAM file.

Z\$KDEL (bin, char\texttt{/}, bin, bin, bin)
\hspace{1cm} call z\$kdel (uniq\_file\_id, key\_buff, key\_len, key\_num, scode);
\hspace{1cm} Delete a record by key match.

Z\$KUPD (bin, char\texttt{/}, bin, bin(31), bin, bin)
\hspace{1cm} call z\$kupd (uniq\_file\_id, rec\_buff, rec\_len, rec\_num, reserved,
\hspace{1cm} \hspace{1cm} code);
\hspace{1cm} Replace the record which the keys in the record presented uniquely identify.

Z\$KYST (bin, (num\_items * 2)bin, bin, bin, 1, 2 char(30), 2 bin, 2 bin, 2 bin, 2 bin, 2 bin, 2 bin, 2 bin, 2 bin, 2 bin, 2 bin, 2 bin, 2 bin)
\hspace{1cm} call z\$kyst (file\_id, key\_info, num\_items, info\_len, key\_found,
\hspace{1cm} \hspace{1cm} code);
\hspace{1cm} Return key\_num and information about a key.
Z$OPEN (bin, char(*), bin, bin, bin, bin, bin)
call z$open (open_key, pathname, pathname_size, tran_key,
            file_org, uniq_file_id, code);
    Open an existing PRISM file. open_key = (O$NWT, O$WAT) + (O$FSH, O$PRO,
            O$EXC) + (O$RDO, O$EXO, O$UPD). tran_key = O$NCK, O$NTM, O$TRM. file_org =
            O$IND, O$REL.

Z$READ (bin, bin, char(*), bin, char(*), bin, bin, char(*), bin, bin, bin, bin)
call z$read (uniq_file_id, funct, rec_buff, rec_len, key_buff,
            key_len, found_key_buff, found_key_len, recsize, reserved,
            code);
    Read a record from a PRISM file and make it the current record. funct = P$FST, P$LST,
            P$EQU, P$GRT, P$GRE, P$NXT, P$NXE, P$NXG, P$PCD, P$CUR.

Z$STRT (bin, bin, bin(31), bin)
call z$strt(key, user_tranid, roam_tranid, acode);
    Start a transaction. key = [T$RTV, T$UPD] + (T$CLR, T$NCL)

Z$UPDT (bin, char(*), bin, bin, bin)
call z$updt(uniq_file_id, rec_buff, rec_len, reserved, code);
    Replace the current record with a user-supplied record.
7. INSTRUCTION SET

7.1. Instruction formats
Further information may be found in the Instruction Sets Guide [19].

7.1.1. S, R, and V mode
S & R - memory reference:
1 2 3 6 7 8 16
I X Opcode S Disp

V - memory reference, short:
1 2 3 6 7 8 16
I X Opcode S Disp

R - memory reference, long:
1 2 3 6 7 12 13 14 15 16
I X Opcode 110000 Ext Cls

or
1 2 3 6 7 12 13 14 15 16 17 32
I X Opcode 110000 Ext Cls Disp

V - memory reference, long:
1 2 3 6 7 11 12 13 14 15 16 17 32
I X Opcode 11000 Y Ext BR DDDDDDDDDDDDDDDDDDDDDDD

or
1 2 3 6 7 11 12 13 14 15 16 17 32
I X Opcode 11000 Y Ext BR DDDDDDDDDDDDDDDDDDDDDDD

Quad opcode

Generic AP:
1 16 17 20 21 22 23 24 25 32

Generic A or B Bit I 0 BR 00000000

33 48

Disp

S, R & V - generic A:
1 6 7 16
110000 Ext
S, R & V - generic B:

<table>
<thead>
<tr>
<th></th>
<th>6</th>
<th>7</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>000000</td>
<td>Ext</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

S, R & V - shift:

<table>
<thead>
<tr>
<th></th>
<th>6</th>
<th>7</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>010000</td>
<td>Ext</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

S, R & V - skip:

<table>
<thead>
<tr>
<th></th>
<th>6</th>
<th>7</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>100000</td>
<td>Ext</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

BR - base register
Bit - bit number
Cls - class bits
Disp - displacement
Ext - extended opcode
I - indirect
Opcodes - opcode
Quad opcode - quad extended opcode
X - index
Y - use Y register for indexing

7.1.2. I mode

General memory reference:

<table>
<thead>
<tr>
<th></th>
<th>6</th>
<th>7</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opcode</td>
<td>Dest</td>
<td>Tag</td>
<td>Src</td>
<td>BR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

or

<table>
<thead>
<tr>
<th></th>
<th>6</th>
<th>7</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opcode</td>
<td>Dest</td>
<td>Tag</td>
<td>Src</td>
<td>BR</td>
<td>Disp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Special memory reference:

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Op</td>
<td>110</td>
<td>OF</td>
<td>O</td>
<td>Tag</td>
<td>Src</td>
<td>BR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

or

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Op</td>
<td>110</td>
<td>OF</td>
<td>O</td>
<td>Tag</td>
<td>Src</td>
<td>BR</td>
<td>Disp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Generic AP:

<table>
<thead>
<tr>
<th></th>
<th>16</th>
<th>17</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>23</th>
<th>25</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gen or Rgen Opcode</td>
<td>Bit</td>
<td>I</td>
<td>0</td>
<td>BR</td>
<td>00000000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

33

Disp

7-2 Prime Restricted
Register generic:

<table>
<thead>
<tr>
<th>1</th>
<th>6 7</th>
<th>9 10</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>011000</td>
<td>Reg</td>
<td>Opcode</td>
</tr>
</tbody>
</table>

Register generic branch:

<table>
<thead>
<tr>
<th>1</th>
<th>6 7</th>
<th>9 10</th>
<th>16 17</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>011000</td>
<td>Reg</td>
<td>Opcode</td>
<td>Disp</td>
</tr>
</tbody>
</table>

BR - base register  
Bit - bit  
Disp - displacement  
F - floating point register  
O, Opc, Opcode - opcode  
Dst - destination register  
Src - source register  
Reg - general register  
Tag - tag modifier

### 7.2. Machine Instructions

The 'type' column indicates the format and/or function of the operation as follows.

- **AP**  Three-word operation, the last two words of which are an AP address pointer.  
- **BR**  Two-word operation, the second word of which is a word number within the current procedure segment to which to branch.  
- **CON**  Single-word control operation.  
- **DA**  Decimal arithmetic operation.  
- **FE**  Field and edit operation.  
- **FLD**  Single-word field operation.  
- **FOP**  Single-word floating-point operation.  
- **FSK**  Single-word floating-point skip operation.  
- **IG**  Single-word integrity operation.  
- **IO**  Single-word input/output operation.  
- **LOG**  Single-word logicle operation.  
- **MGR**  Memory reference/general register to register operation.  
- **MOD**  Single-word mode operation.  
- **MR**  Memory-reference operation.  
- **OPR**  Single-word miscellaneous operation.  
- **PIO**  Programmed input/output operation.  
- **QAD**  Quad floating point.  
- **RAP**  Register A P.  
- **RGN**  Register generic.  
- **SH**  Single-word shift operation.  
- **SKP**  Single-word skip operation.  
- **VM**  Virtual memory operation.

The 'C' column indicates the effect of the operation on the C-bit and the L-bit as follows.

- C and L are unchanged by the operation.  
- 1 C is unchanged, L is carry.  
- 2 C is overflow, L is carry.  
- 3 C is overflow, L is indeterminate.  
- 4 C is shift extension, L is indeterminate.  
- 5 C is a result of op, L is indeterminate.  
- 6 C and L are indeterminate.
7 C and L are loaded by the operation.
8 C is cleared, L is indeterminate.
9 C is a result of op, L is unchanged.

The 'cc' column indicates the effect of the operation on the condition codes as follows.

- Cond. codes are unchanged.
1,4 Cond. codes result of arith op or compare.
5 Cond. codes indeterminate.
6 Cond. codes loaded by operation.
7 Cond. codes indicate result of operation.

The 'Modes' column indicates in which addressing modes the operation is available as follows.

S Available in 16S and 32S modes.
R Available in 32R and 64R modes.
V Available in 64V mode.
I Available in 32I mode (and 32IX).
* Restricted to Ring 0 execution.

Notes following instruction description in parentheses:

32IX I-mode extended instruction. Will not run on all machines.

Long: xxxxxx
   Long form of instruction.
Pxxx For Prime xxx model only.
R Register to register form available.
RI Register to register and immediate forms available.

<table>
<thead>
<tr>
<th>Mnem</th>
<th>Opcode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>004----</td>
<td>MR</td>
<td>2</td>
<td>1</td>
<td>SRV</td>
<td>Add. R + [EA]32 =&gt; R. (RI)</td>
</tr>
<tr>
<td>A1A</td>
<td>141206</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>SRV</td>
<td>Add 1 to A. A + 1 =&gt; A.</td>
</tr>
<tr>
<td>A2A</td>
<td>140304</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>SRV</td>
<td>Add 2 to A. A + 2 =&gt; A.</td>
</tr>
<tr>
<td>ABQ</td>
<td>060134</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td></td>
<td>Add to bottom of Q. CCEQ -&gt; FULL.</td>
</tr>
<tr>
<td>ABQ</td>
<td>141716</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>V</td>
<td>Add to bottom of Q. CCEQ -&gt; FULL.</td>
</tr>
<tr>
<td>ACA</td>
<td>141216</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>SRV</td>
<td>Add CBIT to A. CBIT + A =&gt; A.</td>
</tr>
<tr>
<td>ACP</td>
<td>132----</td>
<td>MGR</td>
<td>-</td>
<td></td>
<td></td>
<td>Add character pointer. (32IX, RI only, see SCC)</td>
</tr>
<tr>
<td>ADL</td>
<td>-15414</td>
<td>MR</td>
<td>2</td>
<td></td>
<td>V</td>
<td>Add long. L + [EA]32 =&gt; L.</td>
</tr>
<tr>
<td>ADL</td>
<td>141000</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>V</td>
<td>Add LINK to L.</td>
</tr>
<tr>
<td>ADR</td>
<td>060014</td>
<td>RGN</td>
<td>-</td>
<td>7</td>
<td>I</td>
<td>Add LINK to R.</td>
</tr>
<tr>
<td>AH</td>
<td>024----</td>
<td>MR</td>
<td>2</td>
<td></td>
<td>I</td>
<td>Add halfword. RH + [EA]16 =&gt; RH. (RI)</td>
</tr>
<tr>
<td>AIP</td>
<td>172----</td>
<td>MGR</td>
<td>2</td>
<td>1</td>
<td></td>
<td>Add indirect pointer. (32IX)</td>
</tr>
<tr>
<td>ALFA</td>
<td>001301</td>
<td>FLD</td>
<td>6</td>
<td></td>
<td>V</td>
<td>Add L to FAR.</td>
</tr>
<tr>
<td>ALL</td>
<td>0414--</td>
<td>SH</td>
<td>4</td>
<td></td>
<td>SRV</td>
<td>A left logical.</td>
</tr>
<tr>
<td>ALR</td>
<td>0416--</td>
<td>SH</td>
<td>4</td>
<td></td>
<td>SRV</td>
<td>A left rotate.</td>
</tr>
<tr>
<td>ALS</td>
<td>0415--</td>
<td>SH</td>
<td>3</td>
<td></td>
<td>SRV</td>
<td>A left shift (arith).</td>
</tr>
<tr>
<td>ANL</td>
<td>-07414</td>
<td>MR</td>
<td>-</td>
<td></td>
<td>V</td>
<td>And long. AND(L, [EA]32) =&gt; L.</td>
</tr>
<tr>
<td>AOA</td>
<td>141206</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>SRV</td>
<td>OBSOLETE. Add 1 to A. A + 1 =&gt; A. (Use A1A)</td>
</tr>
<tr>
<td>ARFA</td>
<td>060161</td>
<td>FLD</td>
<td>6</td>
<td></td>
<td>I</td>
<td>Add R to FAR. FAR + R =&gt; FAR.</td>
</tr>
<tr>
<td>ARG</td>
<td>000605</td>
<td>CON</td>
<td>6</td>
<td>5</td>
<td>VI</td>
<td>Argument transfer (used with PCL).</td>
</tr>
<tr>
<td>AR1</td>
<td>0404--</td>
<td>SH</td>
<td>4</td>
<td></td>
<td></td>
<td>A right logical.</td>
</tr>
<tr>
<td>ARR</td>
<td>0406--</td>
<td>SH</td>
<td>4</td>
<td></td>
<td>SRV</td>
<td>A right rotate.</td>
</tr>
<tr>
<td>ARS</td>
<td>0405--</td>
<td>SH</td>
<td>4</td>
<td></td>
<td>SRV</td>
<td>A right shift (arith).</td>
</tr>
<tr>
<td>ATQ</td>
<td>060135</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>I</td>
<td>Add to top of queue. RH =&gt; Q. CCEQ -&gt; FULL.</td>
</tr>
<tr>
<td>Mnem</td>
<td>OpCode</td>
<td>Typ</td>
<td>C</td>
<td>cc</td>
<td>Modes</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>-----</td>
<td>---</td>
<td>----</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>ATO</td>
<td>141717</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>V</td>
<td>Add to top of queue. A -&gt; Q. CCEQ -&gt; FULL.</td>
</tr>
<tr>
<td>BCEQ</td>
<td>141602</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on Condition Code .EQ.</td>
</tr>
<tr>
<td>BCEG</td>
<td>141605</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on Condition Code .GE.</td>
</tr>
<tr>
<td>BCEH</td>
<td>141601</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on Condition Code .GT.</td>
</tr>
<tr>
<td>BCEI</td>
<td>141500</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on Condition Code .LE.</td>
</tr>
<tr>
<td>BCLT</td>
<td>141604</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on Condition Code .LT.</td>
</tr>
<tr>
<td>BCR</td>
<td>141705</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on CBIT reset.</td>
</tr>
<tr>
<td>BCS</td>
<td>141704</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on CBIT set.</td>
</tr>
<tr>
<td>BDX</td>
<td>140734</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Branch on decremented X.</td>
</tr>
<tr>
<td>BDY</td>
<td>140724</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Branch on decremented Y.</td>
</tr>
<tr>
<td>BEO</td>
<td>140612</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on A . EQ . 0.</td>
</tr>
<tr>
<td>BFEQ</td>
<td>020122</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on FAC . EQ . 0.</td>
</tr>
<tr>
<td>BFEG</td>
<td>020112</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on FAC . GE . 0.</td>
</tr>
<tr>
<td>BFGE</td>
<td>141615</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on FAC . GT . 0.</td>
</tr>
<tr>
<td>BFJG</td>
<td>020121</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on FAC . LT . 0.</td>
</tr>
<tr>
<td>BFLE</td>
<td>141611</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on FAC . NE . 0.</td>
</tr>
<tr>
<td>BFLE</td>
<td>141610</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on FAC . NE . 0.</td>
</tr>
<tr>
<td>BGJ</td>
<td>020120</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on A . EQ . 0.</td>
</tr>
<tr>
<td>BGJG</td>
<td>141613</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on A . GT . 0.</td>
</tr>
<tr>
<td>BHD1</td>
<td>020144</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on RH dec by 1 . RH - 1 =&gt; RH.</td>
</tr>
<tr>
<td>BHD2</td>
<td>020145</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on RH dec by 2 . RH - 2 =&gt; RH.</td>
</tr>
<tr>
<td>BHD4</td>
<td>020146</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on RH dec by 4 . RH - 4 =&gt; RH.</td>
</tr>
<tr>
<td>BHEQ</td>
<td>020112</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on RH . EQ . 0.</td>
</tr>
<tr>
<td>BHEG</td>
<td>020115</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on RH . GE . 0.</td>
</tr>
<tr>
<td>BHTG</td>
<td>020111</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on RH . GT . 0.</td>
</tr>
<tr>
<td>BHI1</td>
<td>020140</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on RH incr by 1 . RH + 1 =&gt; RH.</td>
</tr>
<tr>
<td>BHI2</td>
<td>020141</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on RH incr by 2 . RH + 2 =&gt; RH.</td>
</tr>
<tr>
<td>BHI4</td>
<td>020142</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on RH incr by 4 . RH + 4 =&gt; RH.</td>
</tr>
<tr>
<td>BHLG</td>
<td>020110</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on RH . LE . 0.</td>
</tr>
<tr>
<td>BHLT</td>
<td>020114</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on RH . LT . 0.</td>
</tr>
<tr>
<td>BHNK</td>
<td>020113</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on RH . NE . 0.</td>
</tr>
<tr>
<td>BIY</td>
<td>141334</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Branch on incremented X ^= 0.</td>
</tr>
<tr>
<td>BJY</td>
<td>141324</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Branch on incremented Y ^= 0.</td>
</tr>
<tr>
<td>BLE</td>
<td>140610</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on A &lt; 0.</td>
</tr>
<tr>
<td>BLEQ</td>
<td>140702</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on L = 0.</td>
</tr>
<tr>
<td>BLGE</td>
<td>140615</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on L &gt; 0.</td>
</tr>
<tr>
<td>BLGT</td>
<td>140704</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on L = 0.</td>
</tr>
<tr>
<td>BLLE</td>
<td>140700</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on L &lt; 0.</td>
</tr>
<tr>
<td>BLTT</td>
<td>140614</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on L &gt;= 0.</td>
</tr>
<tr>
<td>BLNE</td>
<td>140703</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on L &lt;= 0.</td>
</tr>
<tr>
<td>BLR</td>
<td>141707</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Branch on LINK reset.</td>
</tr>
<tr>
<td>BLS</td>
<td>141706</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Branch on LINK set.</td>
</tr>
<tr>
<td>BLT</td>
<td>140614</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Branch on A . LT . 0.</td>
</tr>
<tr>
<td>BMEQ</td>
<td>141602</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on mag-cond L . CC . EQ. (BCEO)</td>
</tr>
<tr>
<td>BMGE</td>
<td>141706</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on mag-cond L . CC . GE. (BLS)</td>
</tr>
<tr>
<td>BMGT</td>
<td>141710</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on mag-cond L . CC . GT.</td>
</tr>
<tr>
<td>BMLE</td>
<td>141711</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on mag-cond L . CC . LE.</td>
</tr>
<tr>
<td>BMLT</td>
<td>141707</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on mag-cond L . CC . LT. (BLR)</td>
</tr>
<tr>
<td>BMNE</td>
<td>141603</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on mag-cond L . CC . NE. (BCNE)</td>
</tr>
<tr>
<td>BNE</td>
<td>140613</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on A . NE. 0.</td>
</tr>
<tr>
<td>BRBR</td>
<td>02004-</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on R bit reset.</td>
</tr>
<tr>
<td>BRBS</td>
<td>02000-</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on R bit set.</td>
</tr>
<tr>
<td>BRD1</td>
<td>020134</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on R dec by 1 . R - 1 =&gt; R.</td>
</tr>
<tr>
<td>BRD2</td>
<td>020135</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on R dec by 2 . R - 2 =&gt; R.</td>
</tr>
<tr>
<td>BRD4</td>
<td>020136</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on R dec by 4 . R - 4 =&gt; R.</td>
</tr>
<tr>
<td>BREQ</td>
<td>020102</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on R . EQ. 0.</td>
</tr>
<tr>
<td>BRGE</td>
<td>020105</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>4</td>
<td>Branch on R . NE. 0.</td>
</tr>
<tr>
<td>Mnem</td>
<td>Opcode</td>
<td>Typ</td>
<td>C</td>
<td>cc</td>
<td>Modes</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>-----</td>
<td>---</td>
<td>----</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>BRGT</td>
<td>020101</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Branch on R.LE. 0.</td>
</tr>
<tr>
<td>BRI1</td>
<td>020130</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on R inc by 1. R + 1 =&gt; R.</td>
</tr>
<tr>
<td>BRI2</td>
<td>020131</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on R inc by 2. R + 2 =&gt; R.</td>
</tr>
<tr>
<td>BRI4</td>
<td>020132</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on R inc by 4. R + 4 =&gt; R.</td>
</tr>
<tr>
<td>BRI6</td>
<td>020133</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Branch on R.LT. 0.</td>
</tr>
<tr>
<td>BRLT</td>
<td>020104</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Branch on R.GT. 0.</td>
</tr>
<tr>
<td>BRNE</td>
<td>020103</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Branch on R.GE. 0.</td>
</tr>
<tr>
<td>C</td>
<td>142-----</td>
<td>MR</td>
<td>1</td>
<td>1</td>
<td>I</td>
<td>Compare R with [EA]32. (RI)</td>
</tr>
<tr>
<td>CAI</td>
<td>000411</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRV*</td>
<td>OBSOLETE. Clear active interrupt.</td>
</tr>
<tr>
<td>CAL</td>
<td>141050</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Clear left byte of A.</td>
</tr>
<tr>
<td>CALF</td>
<td>000705</td>
<td>AP</td>
<td>6</td>
<td>5</td>
<td>SRVI</td>
<td>Proc call from faulting proc.</td>
</tr>
<tr>
<td>CAR</td>
<td>141044</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Clear right byte of A.</td>
</tr>
<tr>
<td>CAS</td>
<td>-22-----</td>
<td>MR</td>
<td>1</td>
<td>1</td>
<td>SRV</td>
<td>Skip 0,1,2 if A =&gt;,=,&lt; [EA]16. (Long: -23400)</td>
</tr>
<tr>
<td>CAZ</td>
<td>140214</td>
<td>GEN</td>
<td>1</td>
<td>1</td>
<td>SRV</td>
<td>Skip 0,1,2 if A =&gt;,=,&lt;.</td>
</tr>
<tr>
<td>CEA</td>
<td>000111</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>Compute effective address. EA =&gt; A.</td>
</tr>
<tr>
<td>CGT</td>
<td>060026</td>
<td>RGN</td>
<td>6</td>
<td>5</td>
<td>V</td>
<td>Computed go to.</td>
</tr>
<tr>
<td>CGT</td>
<td>001314</td>
<td>GEN</td>
<td>6</td>
<td>5</td>
<td>V</td>
<td>Computed go to.</td>
</tr>
<tr>
<td>CH</td>
<td>-162----</td>
<td>MR</td>
<td>1</td>
<td>1</td>
<td>I</td>
<td>Compare RH with [EA]16. (RI)</td>
</tr>
<tr>
<td>CHS</td>
<td>060040</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Change sign of R. A^R(R1) =&gt; R(1).</td>
</tr>
<tr>
<td>CLS</td>
<td>-23414</td>
<td>MR</td>
<td>1</td>
<td>1</td>
<td>V</td>
<td>Skip 0,1,2 if L =&gt;,=,&lt; [EA]32.</td>
</tr>
<tr>
<td>CLS</td>
<td>140401</td>
<td>GEN</td>
<td>1</td>
<td>1</td>
<td>V</td>
<td>One's complement A. A^R =&gt; A.</td>
</tr>
<tr>
<td>CMH</td>
<td>060045</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Complement RH. A^RH =&gt; RH.</td>
</tr>
<tr>
<td>CMR</td>
<td>060044</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Complement R. R^R =&gt; R.</td>
</tr>
<tr>
<td>CRI</td>
<td>060056</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Clear R. 0 =&gt; R.</td>
</tr>
<tr>
<td>CRI</td>
<td>140400</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Clear A. 0 =&gt; A.</td>
</tr>
<tr>
<td>CRLE</td>
<td>060055</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Clear B. 0 =&gt; B.</td>
</tr>
<tr>
<td>CRLE</td>
<td>-21410</td>
<td>MR</td>
<td>-</td>
<td>R</td>
<td>I</td>
<td>OBS. Call re-ent. proc. P+1 =&gt; [S+1]16,</td>
</tr>
<tr>
<td>CRH</td>
<td>060054</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Clear RH. 0 =&gt; RH.</td>
</tr>
<tr>
<td>CRHR</td>
<td>060055</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Clear R right halfword. 0 =&gt; R(17-32).</td>
</tr>
<tr>
<td>CRL</td>
<td>-162----</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Clear L. 0 =&gt; L.</td>
</tr>
<tr>
<td>CRL</td>
<td>141410</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Clear L and E. 0 =&gt; L, 0 =&gt; E.</td>
</tr>
<tr>
<td>CSA</td>
<td>140320</td>
<td>GEN</td>
<td>5</td>
<td>-</td>
<td>SRV</td>
<td>Copy sign of A. A(1) =&gt; CBIT,0 =&gt; A(1).</td>
</tr>
<tr>
<td>CSR</td>
<td>060041</td>
<td>RGN</td>
<td>5</td>
<td>-</td>
<td>SRV</td>
<td>Copy &amp; save sign. R(1) =&gt; C, 0 =&gt; R(1).</td>
</tr>
<tr>
<td>CXCS</td>
<td>001714</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>OBSOLETE. Control extended control store.</td>
</tr>
<tr>
<td>D</td>
<td>144-----</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Divide. (R,R+1)[EA]32 =&gt; R; REM =&gt; R</td>
</tr>
<tr>
<td>DAD</td>
<td>-14-----</td>
<td>MR</td>
<td>2</td>
<td>1</td>
<td>SR</td>
<td>Dbl. add. (A,B)+[EA]32 =&gt; A,B whole. (DP,Long:-15400)</td>
</tr>
<tr>
<td>DBL</td>
<td>000007</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Enter double-prec mode.</td>
</tr>
<tr>
<td>DBLE</td>
<td>060106</td>
<td>FOP</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Convert single to double fltg pt.</td>
</tr>
<tr>
<td>DCP</td>
<td>060160</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Decrement character pointer. (321X)</td>
</tr>
<tr>
<td>DFA</td>
<td>0352-----</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Dbl fltg add. DFR + [EA]64 =&gt; DFR. (RI)</td>
</tr>
<tr>
<td>DFD</td>
<td>-15410</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Dbl fltg add. DFAC + [EA]64 =&gt; DFAC.</td>
</tr>
<tr>
<td>DFDC</td>
<td>0152-----</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Dbl fltg compare DFR to [EA]64. (RI)</td>
</tr>
<tr>
<td>DFDM</td>
<td>060144</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Dbl fltg complement. -DFAC =&gt; DFAC.</td>
</tr>
<tr>
<td>DFDM</td>
<td>145074</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Dbl fltg complement. -DFAC =&gt; DFAC.</td>
</tr>
<tr>
<td>DFCS</td>
<td>-23410</td>
<td>MR</td>
<td>6</td>
<td>5</td>
<td>RV</td>
<td>Skip 0,1,2 if DFAC =&gt;,=,&lt; [EA]64.</td>
</tr>
<tr>
<td>DFD</td>
<td>0742-----</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Dbl fltg divide. DFAC/[EA]64 =&gt; DFAC. (RI)</td>
</tr>
<tr>
<td>DFDV</td>
<td>-37410</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Dbl fltg divide. DFAC/[EA]64 =&gt; DFAC.</td>
</tr>
<tr>
<td>DFL</td>
<td>0142-----</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Dbl fltg load. [EA]64 =&gt; DFAC. (RI)</td>
</tr>
<tr>
<td>DFLD</td>
<td>-05410</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>RV</td>
<td>Dbl fltg load. [EA]64 =&gt; DFAC.</td>
</tr>
<tr>
<td>DFLX</td>
<td>-33410</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Load dbl fltg index. 4*[EA]16 =&gt; X. (No X)</td>
</tr>
<tr>
<td>Mnem</td>
<td>OpCode</td>
<td>Typ</td>
<td>C</td>
<td>cc</td>
<td>Modes</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>-----</td>
<td>---</td>
<td>----</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>DFM</td>
<td>0552--</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Dbl fltg multiply. DFAC * [EA]64 =&gt; DFAC. (RI)</td>
</tr>
<tr>
<td>DFMP</td>
<td>-35410</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Dbl fltg multiply. DFAC * [EA]64 =&gt; DFAC.</td>
</tr>
<tr>
<td>DFS</td>
<td>0542--</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Dbl fltg subtract. DFAC - [EA]64 =&gt; DFAC. (RI)</td>
</tr>
<tr>
<td>DFSB</td>
<td>-17410</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Dbl fltg subtract. DFAC - [EA]64 =&gt; DFAC.</td>
</tr>
<tr>
<td>DFSF</td>
<td>0342--</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Dbl fltg store. DFAC =&gt; [EA]64.</td>
</tr>
<tr>
<td>DFSF</td>
<td>-11410</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>RV</td>
<td>Dbl fltg store. DFAC =&gt; [EA]64.</td>
</tr>
<tr>
<td>DH</td>
<td>164---</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Divide halfword. R/[EA]16 =&gt; RH; RM =&gt; RL. (RI)</td>
</tr>
<tr>
<td>DH1</td>
<td>060130</td>
<td>RGN</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Decr RH by 1. RH - 1 =&gt; RH.</td>
</tr>
<tr>
<td>DH2</td>
<td>060131</td>
<td>RGN</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Decr RH by 2. RH - 2 =&gt; RH.</td>
</tr>
<tr>
<td>DIV</td>
<td>-36---</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>SR</td>
<td>Divide. (A,B)/31/[EA]16 =&gt; A; REM =&gt; B. (Long: -37400)</td>
</tr>
<tr>
<td>DIV</td>
<td>-36---</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Divide. L/[EA]16 =&gt; A; REM =&gt; B. (Long: -37400)</td>
</tr>
<tr>
<td>DR1</td>
<td>060124</td>
<td>RGN</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Decr R by 1. R - 1 =&gt; R.</td>
</tr>
<tr>
<td>DR2</td>
<td>060125</td>
<td>RGN</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Decr R by 2. R - 2 =&gt; R.</td>
</tr>
<tr>
<td>DRN</td>
<td>040300</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Double round from quad</td>
</tr>
<tr>
<td>DRNM</td>
<td>140571</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Double round from quad to -infinity.</td>
</tr>
<tr>
<td>DRNP</td>
<td>040301</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Double round from quad to -infinity.</td>
</tr>
<tr>
<td>DRNZ</td>
<td>040302</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Double round from quad to 0.</td>
</tr>
<tr>
<td>DRX</td>
<td>140210</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Decrement X and skip if 0.</td>
</tr>
<tr>
<td>DST</td>
<td>-10---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>Double store. (A,B) =&gt; [EA]32. (DP, Long: -11400)</td>
</tr>
<tr>
<td>DVL</td>
<td>-37414</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Divide long. (L,E)/[EA]32 =&gt; L; REM =&gt; E.</td>
</tr>
<tr>
<td>DXA</td>
<td>000011</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>SRVI</td>
<td>OBSOLETE. Enter 16K secedted mode. (Use E16S)</td>
</tr>
<tr>
<td>E16S</td>
<td>000011</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>SRVI</td>
<td>Enter P300 16K sectored mode.</td>
</tr>
<tr>
<td>E32I</td>
<td>001010</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>SRVI</td>
<td>Enter P500 32I mode.</td>
</tr>
<tr>
<td>E32R</td>
<td>001013</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>SRVI</td>
<td>Enter P300 32K relative mode.</td>
</tr>
<tr>
<td>E32S</td>
<td>000013</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>SRVI</td>
<td>Enter P300 32K sectored mode.</td>
</tr>
<tr>
<td>E64R</td>
<td>001011</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>SRVI</td>
<td>Enter P300 64K relative mode.</td>
</tr>
<tr>
<td>E64V</td>
<td>000010</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>SRVI</td>
<td>Enter P400 64K virtual mode.</td>
</tr>
<tr>
<td>EAA</td>
<td>-03404</td>
<td>MR</td>
<td>-</td>
<td>R</td>
<td>V</td>
<td>Eff. addr to A, EA =&gt; A.</td>
</tr>
<tr>
<td>EAF</td>
<td>001300</td>
<td>AP</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Eff. addr to A, EA =&gt; A.</td>
</tr>
<tr>
<td>EAL</td>
<td>-03404</td>
<td>MR</td>
<td>-</td>
<td>R</td>
<td>V</td>
<td>Eff. addr to L, EA =&gt; L.</td>
</tr>
<tr>
<td>EALB</td>
<td>1144--</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Eff. addr to LB, EA =&gt; LB.</td>
</tr>
<tr>
<td>EALB</td>
<td>-27410</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Eff. addr to LB, EA =&gt; LB.</td>
</tr>
<tr>
<td>EAR</td>
<td>146---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Eff. addr to R, EA =&gt; R.</td>
</tr>
<tr>
<td>EAXB</td>
<td>1344--</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Eff. addr to XB, EA =&gt; XB.</td>
</tr>
<tr>
<td>EAXB</td>
<td>-25410</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Eff. addr to XB, EA =&gt; XB.</td>
</tr>
<tr>
<td>EIO</td>
<td>070---</td>
<td>MR</td>
<td>-</td>
<td>7</td>
<td>I*</td>
<td>Execute EA as I/O inst. CCEQ =&gt; success.</td>
</tr>
<tr>
<td>EIO</td>
<td>-31404</td>
<td>MR</td>
<td>-</td>
<td>7</td>
<td>V*</td>
<td>Execute EA as I/O inst. CCEQ =&gt; success.</td>
</tr>
<tr>
<td>EMCM</td>
<td>000503</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>OBSOLETE. Enter machine check mode.</td>
</tr>
<tr>
<td>ENB</td>
<td>00401</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Enable interrupts.</td>
</tr>
<tr>
<td>ENBL</td>
<td>00401</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Enable interrupts (local). (P850)</td>
</tr>
<tr>
<td>ENBM</td>
<td>00400</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Enable interrupts (mutual). (P850)</td>
</tr>
<tr>
<td>ENBP</td>
<td>00402</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Enable interrupts (process). (P850)</td>
</tr>
</tbody>
</table>

The document is from the Prime Engineering Handbook and contains instructions for various operations in the Prime computer system.
<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Opcode</th>
<th>Type</th>
<th>C</th>
<th>cc</th>
<th>Mode(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTR</td>
<td>-03414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>OBSOLETE. Enter recursive proc stack.</td>
</tr>
<tr>
<td>EPMJ</td>
<td>000217</td>
<td>VM</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>OBSOLETE. Enter page mode &amp; jump (P300).</td>
</tr>
<tr>
<td>EPMX</td>
<td>000237</td>
<td>VM</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>OBS. Enter page mode &amp; jump to microcode (P300).</td>
</tr>
<tr>
<td>ERA</td>
<td>-12</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Exclusive or. XOR(A, [EA]16) =&gt; A.</td>
</tr>
<tr>
<td>ERL</td>
<td>-13414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Exclusive or long. XOR(L, [EA]32) =&gt; L.</td>
</tr>
<tr>
<td>ERMJ</td>
<td>000701</td>
<td>VM</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>OBSOLETE. Enter restricted mode &amp; jump (P300).</td>
</tr>
<tr>
<td>ERMX</td>
<td>000721</td>
<td>VM</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>OBS. Enter restr'd mode &amp; jump to ucode (P300).</td>
</tr>
<tr>
<td>ESIM</td>
<td>000415</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>OBSOLETE. Enter standard interrupt mode.</td>
</tr>
<tr>
<td>EVIM</td>
<td>000417</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>OBSOLETE. Enter vectored interrupt mode.</td>
</tr>
<tr>
<td>EVMJ</td>
<td>000703</td>
<td>VM</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>OBSOLETE. Enter virtual mode &amp; jump (P300).</td>
</tr>
<tr>
<td>EVMX</td>
<td>000723</td>
<td>VM</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>OBS. Enter virtual mode &amp; jump to ucode (P300).</td>
</tr>
<tr>
<td>EXA</td>
<td>000013</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>SRVI</td>
<td>OBSOLETE. Enter 32K sectored mode. (Use E325)</td>
</tr>
<tr>
<td>FA</td>
<td>0350</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Flt add. FAC + [EA]32 =&gt; FAC. (RI)</td>
</tr>
<tr>
<td>FAD</td>
<td>-15404</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Flt add. FAC + [EA]32 =&gt; FAC.</td>
</tr>
<tr>
<td>FC</td>
<td>0150</td>
<td>MR</td>
<td>-</td>
<td>1</td>
<td>I</td>
<td>Flt compare FAC with [EA]32. (RI)</td>
</tr>
<tr>
<td>FDCQ</td>
<td>140571</td>
<td>FOP</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Flt convert dbl to quad. (P9950)</td>
</tr>
<tr>
<td>FCM</td>
<td>060100</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Flt complement. -FAC =&gt; FAC.</td>
</tr>
<tr>
<td>FCM</td>
<td>140530</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Flt complement. -FAC =&gt; FAC.</td>
</tr>
<tr>
<td>FCS</td>
<td>-23404</td>
<td>MR</td>
<td>6</td>
<td>5</td>
<td>I</td>
<td>Skip 0, 1, 2 if FAC &gt;= &lt; [EA]32. (RI)</td>
</tr>
<tr>
<td>FD</td>
<td>0740</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Flt divide. FAC / [EA]32 =&gt; FAC.</td>
</tr>
<tr>
<td>FDBL</td>
<td>140016</td>
<td>FOP</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Flt convert single to dbl. FAC =&gt; DFAC.</td>
</tr>
<tr>
<td>FDP</td>
<td>-37404</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Flt divide. FAC / [EA]32 =&gt; FAC.</td>
</tr>
<tr>
<td>FL</td>
<td>0140</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Flt load. [EA]32 =&gt; FAC. (RI)</td>
</tr>
<tr>
<td>FLD</td>
<td>-05404</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>RV</td>
<td>Flt load. [EA]32 =&gt; FAC.</td>
</tr>
<tr>
<td>FLOT</td>
<td>140550</td>
<td>FOP</td>
<td>6</td>
<td>5</td>
<td>R</td>
<td>Convert int to flt. Flot(A,B) =&gt; FAC w/ hole.</td>
</tr>
<tr>
<td>FLT</td>
<td>060105</td>
<td>FOP</td>
<td>6</td>
<td>5</td>
<td>I</td>
<td>Convert int to flt. Flot(R) =&gt; FAC.</td>
</tr>
<tr>
<td>FLT</td>
<td>140532</td>
<td>FOP</td>
<td>6</td>
<td>5</td>
<td>V</td>
<td>Convert int to flt. Flot(A) =&gt; FAC.</td>
</tr>
<tr>
<td>FLTH</td>
<td>060102</td>
<td>FOP</td>
<td>6</td>
<td>5</td>
<td>I</td>
<td>Convert half word int to flt pt.</td>
</tr>
<tr>
<td>FLTL</td>
<td>140535</td>
<td>FOP</td>
<td>6</td>
<td>5</td>
<td>V</td>
<td>Convert long to flt. Flot(L) =&gt; FAC.</td>
</tr>
<tr>
<td>FLX</td>
<td>-33404</td>
<td>MR</td>
<td>5</td>
<td>5</td>
<td>I</td>
<td>Load flt index. 2*[EA]16 =&gt; X. (No X)</td>
</tr>
<tr>
<td>FM</td>
<td>0550</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Flt multiply. FAC * [EA]32 =&gt; FAC. (RI)</td>
</tr>
<tr>
<td>FMP</td>
<td>-35404</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Flt multiply. FAC * [EA]32 =&gt; FAC.</td>
</tr>
<tr>
<td>FRN</td>
<td>060107</td>
<td>FOP</td>
<td>6</td>
<td>5</td>
<td>RV</td>
<td>Flt round.</td>
</tr>
<tr>
<td>FRN</td>
<td>140534</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Flt round up.</td>
</tr>
<tr>
<td>FRNM</td>
<td>060146</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Flt round towards - infinity.</td>
</tr>
<tr>
<td>FRNM</td>
<td>040320</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Flt round towards - infinity.</td>
</tr>
<tr>
<td>FRNP</td>
<td>060145</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Flt round towards + infinity.</td>
</tr>
<tr>
<td>FRNP</td>
<td>040303</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Flt round towards + infinity.</td>
</tr>
<tr>
<td>FRNZ</td>
<td>060147</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Flt round towards zero.</td>
</tr>
<tr>
<td>FRNZ</td>
<td>040321</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Flt round towards zero.</td>
</tr>
<tr>
<td>FS</td>
<td>0540</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Flt subtract. FAC - [EA]32 =&gt; FAC. (RI)</td>
</tr>
<tr>
<td>FSB</td>
<td>-17404</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Flt subtract. FAC - [EA]32 =&gt; FAC.</td>
</tr>
<tr>
<td>FSGL</td>
<td>140515</td>
<td>FSK</td>
<td>1</td>
<td>-</td>
<td>RV</td>
<td>Flt skip if .GT. 0.</td>
</tr>
<tr>
<td>FSL</td>
<td>140514</td>
<td>FSK</td>
<td>1</td>
<td>-</td>
<td>RV</td>
<td>Flt skip if .LE. 0.</td>
</tr>
<tr>
<td>FSMD</td>
<td>140512</td>
<td>FSK</td>
<td>1</td>
<td>-</td>
<td>RV</td>
<td>Flt skip if .LT. 0.</td>
</tr>
<tr>
<td>FSNZ</td>
<td>140511</td>
<td>FSK</td>
<td>1</td>
<td>-</td>
<td>RV</td>
<td>Flt skip if .NE. 0.</td>
</tr>
<tr>
<td>FSPL</td>
<td>140513</td>
<td>FSK</td>
<td>1</td>
<td>-</td>
<td>RV</td>
<td>Flt skip if .GE. 0.</td>
</tr>
<tr>
<td>FST</td>
<td>0340</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Flt store. FAC =&gt; [EA]32.</td>
</tr>
<tr>
<td>FST</td>
<td>-11404</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Flt store. FAC =&gt; [EA]32.</td>
</tr>
<tr>
<td>FSZE</td>
<td>140510</td>
<td>FSK</td>
<td>1</td>
<td>-</td>
<td>RV</td>
<td>Flt skip if .EQ. 0.</td>
</tr>
<tr>
<td>HLT</td>
<td>000000</td>
<td>CON</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Halt computer operation.</td>
</tr>
<tr>
<td>I</td>
<td>102</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Interchange R with [EA]32. (RI)</td>
</tr>
<tr>
<td>Mnem</td>
<td>OpCode</td>
<td>Typ</td>
<td>C</td>
<td>cc</td>
<td>Modes</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>-----</td>
<td>----</td>
<td>-----</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>IAB</td>
<td>000201</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Exchange A and B. A =&gt; B &amp; B =&gt; A.</td>
</tr>
<tr>
<td>ICA</td>
<td>141340</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Interchange bytes of A.</td>
</tr>
<tr>
<td>ICBL</td>
<td>060065</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Exchange bytes. 0 =&gt; RH(1-8) =&gt; RH(9-16).</td>
</tr>
<tr>
<td>ICBR</td>
<td>060066</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Exchange bytes. 0 =&gt; RH(9-16) =&gt; RH(1-8).</td>
</tr>
<tr>
<td>ICHL</td>
<td>060060</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Interchange halfwords. RH =&gt; RL, 0 =&gt; RH.</td>
</tr>
<tr>
<td>ICHR</td>
<td>060061</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Interchange halfwords. RL =&gt; RH, 0 =&gt; RL.</td>
</tr>
<tr>
<td>ICL</td>
<td>141140</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Exchange bytes of A &amp; clrt left.</td>
</tr>
<tr>
<td>ICP</td>
<td>060167</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Increment character pointer. (32IX)</td>
</tr>
<tr>
<td>ICR</td>
<td>141240</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Exchange bytes of A &amp; clrt right.</td>
</tr>
<tr>
<td>IH</td>
<td>122</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Interchange RH with [EA]16. (R)</td>
</tr>
<tr>
<td>IH1</td>
<td>080126</td>
<td>RGN</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Incr halfword by 1. RH + 1 =&gt; RH.</td>
</tr>
<tr>
<td>IH2</td>
<td>060127</td>
<td>RGN</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Incr halfword by 2. RH + 2 =&gt; RH.</td>
</tr>
<tr>
<td>ILE</td>
<td>141414</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Exchange L and E. L =&gt; E &amp; E =&gt; L.</td>
</tr>
<tr>
<td>IM</td>
<td>1140</td>
<td>MR</td>
<td>-</td>
<td>1</td>
<td>I</td>
<td>Incr memory. [EA]32 + 1 =&gt; [EA]32.</td>
</tr>
<tr>
<td>INA</td>
<td>130</td>
<td>PIO</td>
<td>-</td>
<td>-</td>
<td>SR*</td>
<td>Input to A.</td>
</tr>
<tr>
<td>INBC</td>
<td>001217</td>
<td>AP</td>
<td>6</td>
<td>5</td>
<td>VI*</td>
<td>Interrupt ntfy LIFO, clear active interrupt.</td>
</tr>
<tr>
<td>INBN</td>
<td>001215</td>
<td>AP</td>
<td>6</td>
<td>5</td>
<td>VI*</td>
<td>Interrupt ntfy FIFO.</td>
</tr>
<tr>
<td>INEC</td>
<td>001216</td>
<td>AP</td>
<td>6</td>
<td>5</td>
<td>VI*</td>
<td>Interrupt ntfy FIFO, clear active interrupt.</td>
</tr>
<tr>
<td>INEN</td>
<td>001214</td>
<td>AP</td>
<td>6</td>
<td>5</td>
<td>VI*</td>
<td>Interrupt ntfy FIFO.</td>
</tr>
<tr>
<td>INH</td>
<td>001001</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Inhibit interrupts.</td>
</tr>
<tr>
<td>INHL</td>
<td>001001</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Inhibit interrupts (local), (P850).</td>
</tr>
<tr>
<td>INHM</td>
<td>001000</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Inhibit interrupts (mutual), (P850).</td>
</tr>
<tr>
<td>INHP</td>
<td>001002</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Inhibit interrupts (process), (P850).</td>
</tr>
<tr>
<td>INK</td>
<td>060070</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Input keys to RH.</td>
</tr>
<tr>
<td>INK</td>
<td>000043</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>Input P300 keys into A.</td>
</tr>
<tr>
<td>INT</td>
<td>060103</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Convert fltg to int. INT(FAC) =&gt; R.</td>
</tr>
<tr>
<td>INTA</td>
<td>140554</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>R</td>
<td>Convert fltg to int. INT(FAC) =&gt; A, B w/ hole.</td>
</tr>
<tr>
<td>INTB</td>
<td>060101</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Convert fltg to int. INT(FAC) =&gt; A.</td>
</tr>
<tr>
<td>INTH</td>
<td>060101</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Convert fltg to whalford. INT(FAC) =&gt; RH.</td>
</tr>
<tr>
<td>INTL</td>
<td>140533</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Convert fltg to int long. INT(FAC) =&gt; L.</td>
</tr>
<tr>
<td>IR1</td>
<td>060122</td>
<td>RGN</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Incr R by 1. R + 1 =&gt; R.</td>
</tr>
<tr>
<td>IR2</td>
<td>060123</td>
<td>RGN</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Incr R by 2. R + 2 =&gt; R.</td>
</tr>
<tr>
<td>IRB</td>
<td>060064</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Interchange halfwords. RH(1-8) =&gt; RH(9-16).</td>
</tr>
<tr>
<td>IRH</td>
<td>060057</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Interchange halves. RH =&gt; RL.</td>
</tr>
<tr>
<td>IRS</td>
<td>-24</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Inc. replace, and skip if zero. (LONG: -25400).</td>
</tr>
<tr>
<td>IRTC</td>
<td>000603</td>
<td>CON</td>
<td>7</td>
<td>6</td>
<td>VI*</td>
<td>Interrupt return, clear active intrpt.</td>
</tr>
<tr>
<td>IRTN</td>
<td>000601</td>
<td>CON</td>
<td>7</td>
<td>6</td>
<td>VI*</td>
<td>Interrupt return.</td>
</tr>
<tr>
<td>IRX</td>
<td>140114</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Increment X and skip if 0.</td>
</tr>
<tr>
<td>ITLB</td>
<td>000615</td>
<td>CON</td>
<td>6</td>
<td>5</td>
<td>VI*</td>
<td>Invalidate STLB entry, L, R2 = VADDR.</td>
</tr>
<tr>
<td>JDX</td>
<td>-33410</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>Decrement X &amp; jump if not zero. (No X)</td>
</tr>
<tr>
<td>JEO</td>
<td>-05414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>OBsolete. Jump if A .EQ. 0, EA =&gt; P.</td>
</tr>
<tr>
<td>JEG</td>
<td>-17414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>OBsolete. Jump if A .GE. 0, EA =&gt; P.</td>
</tr>
<tr>
<td>JGT</td>
<td>-13414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>OBsolete. Jump if A .GT. 0, EA =&gt; P.</td>
</tr>
<tr>
<td>JIX</td>
<td>-33414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>Increment X &amp; jump if not zero. (No X)</td>
</tr>
<tr>
<td>JLE</td>
<td>-11414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>OBsolete. Jump if A .LE. 0, EA =&gt; P.</td>
</tr>
<tr>
<td>JLT</td>
<td>-15414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>OBsolete. Jump if A .LT. 0, EA =&gt; P.</td>
</tr>
<tr>
<td>JMP</td>
<td>1342</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Jump. EA =&gt; P.</td>
</tr>
<tr>
<td>JNE</td>
<td>-07414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>OBsolete. Jump if A .NE. 0, EA =&gt; P.</td>
</tr>
<tr>
<td>JSR</td>
<td>166</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Jump to subr. P =&gt; RH, EA32 =&gt; P.</td>
</tr>
<tr>
<td>Mnem</td>
<td>OpCode</td>
<td>Typ</td>
<td>C</td>
<td>cc</td>
<td>Modes</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>-----</td>
<td>---</td>
<td>----</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>JSX</td>
<td>-73414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>RV</td>
<td>Jump &amp; save in X. P =&gt; X, EA =&gt; P. (No X)</td>
</tr>
<tr>
<td>JSXB</td>
<td>1542--</td>
<td>MR</td>
<td>-</td>
<td>I</td>
<td></td>
<td>Jump &amp; set XB. P =&gt; XB, EA =&gt; P.</td>
</tr>
<tr>
<td>JSY</td>
<td>-31410</td>
<td>MR</td>
<td>-</td>
<td>V</td>
<td></td>
<td>Jump &amp; set XB, PB =&gt; XB, EA =&gt; PB.</td>
</tr>
<tr>
<td>L</td>
<td>002---</td>
<td>MR</td>
<td>-</td>
<td>I</td>
<td></td>
<td>Load R. [EA]32 =&gt; R. (RI)</td>
</tr>
<tr>
<td>LCC</td>
<td>112---</td>
<td>MGR</td>
<td>-</td>
<td>I</td>
<td></td>
<td>Load character via char pointer. (32IX, RI)</td>
</tr>
<tr>
<td>LCEQ</td>
<td>060153</td>
<td>LOG</td>
<td>-</td>
<td>I</td>
<td></td>
<td>Load RH if EQ. CCEQ =&gt; RH.</td>
</tr>
<tr>
<td>LCEQ</td>
<td>141503</td>
<td>LOG</td>
<td>-</td>
<td>V</td>
<td></td>
<td>Load A if EQ. CCEQ =&gt; A.</td>
</tr>
<tr>
<td>LCEQ</td>
<td>060154</td>
<td>LOG</td>
<td>-</td>
<td>I</td>
<td></td>
<td>Load RH if GE. CCGE =&gt; RH.</td>
</tr>
<tr>
<td>LCEQ</td>
<td>141504</td>
<td>LOG</td>
<td>-</td>
<td>V</td>
<td></td>
<td>Load A if GE. CCGE =&gt; A.</td>
</tr>
<tr>
<td>LCT</td>
<td>060155</td>
<td>LOG</td>
<td>-</td>
<td>I</td>
<td></td>
<td>Load RH if GT. CCGT =&gt; RH.</td>
</tr>
<tr>
<td>LCT</td>
<td>141505</td>
<td>LOG</td>
<td>-</td>
<td>V</td>
<td></td>
<td>Load A if GT. CCGT =&gt; A.</td>
</tr>
<tr>
<td>LCE</td>
<td>060151</td>
<td>LOG</td>
<td>-</td>
<td>I</td>
<td></td>
<td>Load RH if LE. CCLE =&gt; RH.</td>
</tr>
<tr>
<td>LCE</td>
<td>141501</td>
<td>LOG</td>
<td>-</td>
<td>V</td>
<td></td>
<td>Load A if LE. CCLE =&gt; A.</td>
</tr>
<tr>
<td>LCL</td>
<td>060150</td>
<td>LOG</td>
<td>-</td>
<td>I</td>
<td></td>
<td>Load RH if LT. CCLT =&gt; RH.</td>
</tr>
<tr>
<td>LCL</td>
<td>141500</td>
<td>LOG</td>
<td>-</td>
<td>V</td>
<td></td>
<td>Load A if LT. CCLT =&gt; A.</td>
</tr>
<tr>
<td>LCN</td>
<td>060152</td>
<td>LOG</td>
<td>-</td>
<td>I</td>
<td></td>
<td>Load RH if NE. CCNE =&gt; RH.</td>
</tr>
<tr>
<td>LCN</td>
<td>141502</td>
<td>LOG</td>
<td>-</td>
<td>V</td>
<td></td>
<td>Load A if NE. CCNE =&gt; A.</td>
</tr>
<tr>
<td>LDA</td>
<td>110---</td>
<td>MR</td>
<td>-</td>
<td>I(*)</td>
<td></td>
<td>Load addressed register.</td>
</tr>
<tr>
<td>LDC</td>
<td>060162</td>
<td>FLD</td>
<td>-</td>
<td>7</td>
<td></td>
<td>Load char to RH.</td>
</tr>
<tr>
<td>LDC</td>
<td>001302</td>
<td>FLD</td>
<td>-</td>
<td>7</td>
<td></td>
<td>Load char to A via FAR.</td>
</tr>
<tr>
<td>LDL</td>
<td>-05414</td>
<td>MR</td>
<td>-</td>
<td>7</td>
<td></td>
<td>Load long. [EA]32 =&gt; L.</td>
</tr>
<tr>
<td>LDLR</td>
<td>-13404</td>
<td>MR</td>
<td>-</td>
<td>V(*)</td>
<td></td>
<td>Load long from addressed reg.</td>
</tr>
<tr>
<td>LDY</td>
<td>-73404</td>
<td>MR</td>
<td>-</td>
<td>V</td>
<td></td>
<td>Load Y. [EA]16 =&gt; Y. (No X)</td>
</tr>
<tr>
<td>LEQ</td>
<td>060003</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if R = 0. (R = 0) =&gt; RH.</td>
</tr>
<tr>
<td>LEQ</td>
<td>140413</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>SRV</td>
<td>If A .EQ. 0, 1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LF</td>
<td>060016</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Logicize false. 0 =&gt; RH.</td>
</tr>
<tr>
<td>LF</td>
<td>140416</td>
<td>LOG</td>
<td>-</td>
<td>5</td>
<td>SRV</td>
<td>Logicize false. 0 =&gt; A.</td>
</tr>
<tr>
<td>LFEQ</td>
<td>060015</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if FAC = 0. (FAC = 0) =&gt; RH.</td>
</tr>
<tr>
<td>LFEQ</td>
<td>141113</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If FAC .EQ. 0, 1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LFGE</td>
<td>060024</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if FAC &gt;= 0. (FAC &gt;= 0) =&gt; RH.</td>
</tr>
<tr>
<td>LFGE</td>
<td>141114</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If FAC .GE. 0, 1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LFGT</td>
<td>060025</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if FAC &gt; 0. (FAC &gt; 0) =&gt; RH.</td>
</tr>
<tr>
<td>LFGT</td>
<td>141115</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If FAC .GT. 0, 1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LFLE</td>
<td>060021</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if FAC &lt;= 0. (FAC &lt;= 0) =&gt; RH.</td>
</tr>
<tr>
<td>LFLE</td>
<td>141111</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If FAC .LE. 0, 1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LFLL</td>
<td>001303</td>
<td>FLD</td>
<td>-</td>
<td></td>
<td>SRVI</td>
<td>Load FLR immediate.</td>
</tr>
<tr>
<td>LFLL</td>
<td>060020</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if FAC &lt; 0. (FAC &lt; 0) =&gt; RH.</td>
</tr>
<tr>
<td>LFLL</td>
<td>141110</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If FAC .LT. 0, 1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LFNE</td>
<td>060022</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if FAC &lt;&gt; 0. (FAC &lt;&gt; 0) =&gt; RH.</td>
</tr>
<tr>
<td>LFNE</td>
<td>141112</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If FAC .NE. 0, 1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LGE</td>
<td>060004</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if R &gt;= 0. (R &gt;= 0) =&gt; RH.</td>
</tr>
<tr>
<td>LGE</td>
<td>140414</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>SRV</td>
<td>If A .GE. 0, 1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LGL</td>
<td>0414--</td>
<td>SH</td>
<td>4</td>
<td>5</td>
<td>SRV</td>
<td>OBSOLETE. A left logical. (Use ALL)</td>
</tr>
<tr>
<td>LGR</td>
<td>0404--</td>
<td>SH</td>
<td>4</td>
<td>5</td>
<td>SRV</td>
<td>OBSOLETE. A right logical. (Use ARL)</td>
</tr>
<tr>
<td>LGT</td>
<td>060005</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if R &gt; 0. (R &gt; 0) =&gt; RH.</td>
</tr>
<tr>
<td>LGT</td>
<td>140415</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>SRV</td>
<td>If A .GT. 0, 1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LH</td>
<td>022---</td>
<td>MR</td>
<td>-</td>
<td>I</td>
<td></td>
<td>Load halfword. [EA]16 =&gt; RH. (RI)</td>
</tr>
<tr>
<td>LEHO</td>
<td>060013</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if RH &gt;= 0. (RH = 0) =&gt; RH.</td>
</tr>
<tr>
<td>LHGE</td>
<td>060004</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if RH &gt;= 0. (RH &gt; 0) =&gt; RH.</td>
</tr>
<tr>
<td>LHGT</td>
<td>060015</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if RH &gt; 0. (RH &gt; 0) =&gt; RH.</td>
</tr>
<tr>
<td>LHL1</td>
<td>010---</td>
<td>MR</td>
<td>-</td>
<td></td>
<td></td>
<td>Load halfwd shifted by 1. LS( [EA]16,1 ) =&gt; RH. (R)</td>
</tr>
<tr>
<td>Mnem</td>
<td>OpCode</td>
<td>Typ</td>
<td>C</td>
<td>cc</td>
<td>Modes</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>-----</td>
<td>---</td>
<td>----</td>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>LHL2</td>
<td>030---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load halfword shifted by 2. LS((EA)16,2) =&gt; RH. (R)</td>
</tr>
<tr>
<td>LHL3</td>
<td>072---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load halfword shifted by 3. LS((EA)16,3) =&gt; RH. (R)</td>
</tr>
<tr>
<td>LHLE</td>
<td>060011</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if RH &lt;= 0. (RH &lt;= 0) =&gt; RH.</td>
</tr>
<tr>
<td>LHLT</td>
<td>060000</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if RH &lt; 0. (RH &lt; 0) =&gt; RH.</td>
</tr>
<tr>
<td>LHNE</td>
<td>060012</td>
<td>LOG</td>
<td>-</td>
<td>2</td>
<td>I</td>
<td>Load RH if RH &lt;= 0. (RH &lt;= 0) =&gt; RH.</td>
</tr>
<tr>
<td>LIOT</td>
<td>000444</td>
<td>AP</td>
<td>6</td>
<td>5</td>
<td>VI*</td>
<td>Load IOTLB. L, R2 =&gt; target virt addr.</td>
</tr>
<tr>
<td>LIP</td>
<td>152---</td>
<td>MGR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load indirect pointer. (32IX)</td>
</tr>
<tr>
<td>LLE</td>
<td>060001</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if R &lt; 0. (R &lt; 0) =&gt; RH.</td>
</tr>
<tr>
<td>LLEQ</td>
<td>140411</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>SRV</td>
<td>If A.LEQ. 0, 1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LLGE</td>
<td>140414</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If L.GE. 0, 1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LLGT</td>
<td>141515</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If L.GT. 0, 1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LLL</td>
<td>0410--</td>
<td>SH</td>
<td>4</td>
<td>-</td>
<td>V</td>
<td>Long left logical.</td>
</tr>
<tr>
<td>LLLE</td>
<td>141511</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If L.LE. 0, 1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LLLT</td>
<td>140410</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If L.LT. 0, 1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LLNE</td>
<td>141512</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If L.NE. 0, 1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LLR</td>
<td>0412--</td>
<td>SH</td>
<td>4</td>
<td>-</td>
<td>V</td>
<td>Long left rotate.</td>
</tr>
<tr>
<td>LLS</td>
<td>0411--</td>
<td>SH</td>
<td>3</td>
<td>5</td>
<td>SRV</td>
<td>Long left shift. (SR -&gt; B(1) ignored)</td>
</tr>
<tr>
<td>LLLT</td>
<td>060000</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load R if R &lt; 0. (R &lt; 0) =&gt; R.</td>
</tr>
<tr>
<td>LLMC</td>
<td>000501</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>SRVI</td>
<td>Leave machine check mode.</td>
</tr>
<tr>
<td>LNE</td>
<td>060002</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Leave machine check mode.</td>
</tr>
<tr>
<td>LNCR</td>
<td>140412</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>SRV</td>
<td>If A.NE. 0, 1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LPID</td>
<td>000611</td>
<td>CON</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>Load process ID from A(1-12), R2(1-12).</td>
</tr>
<tr>
<td>LPMJ</td>
<td>000215</td>
<td>VM</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>OBSOLETE. Leave page mode &amp; jump to microcode (P300).</td>
</tr>
<tr>
<td>LPMX</td>
<td>000235</td>
<td>VM</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>OBS. Leave page mode &amp; jump to microcode (P300).</td>
</tr>
<tr>
<td>LPSW</td>
<td>000711</td>
<td>AP</td>
<td>7</td>
<td>6</td>
<td>VI*</td>
<td>Load PSW (SN,WN,KEY,MODAL).</td>
</tr>
<tr>
<td>LRL</td>
<td>0400--</td>
<td>SH</td>
<td>4</td>
<td>-</td>
<td>SRV</td>
<td>Long right logical.</td>
</tr>
<tr>
<td>LRR</td>
<td>0402--</td>
<td>SH</td>
<td>4</td>
<td>-</td>
<td>SRV</td>
<td>Long right rotate.</td>
</tr>
<tr>
<td>LRS</td>
<td>0401--</td>
<td>SH</td>
<td>4</td>
<td>-</td>
<td>SRV</td>
<td>Long right shift. (SR -&gt; B(1) ignored)</td>
</tr>
<tr>
<td>LT</td>
<td>060017</td>
<td>LOG</td>
<td>-</td>
<td>5</td>
<td>I</td>
<td>Logic set true. 1 =&gt; R.</td>
</tr>
<tr>
<td>LTT</td>
<td>141515</td>
<td>LOG</td>
<td>-</td>
<td>5</td>
<td>SRV</td>
<td>Logic set true. 1 =&gt; A.</td>
</tr>
<tr>
<td>LWCS</td>
<td>001710</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>OBSOLETE. Load writable control store.</td>
</tr>
<tr>
<td>M</td>
<td>104---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>OBSOLETE. Mem diag inhibit interleave.</td>
</tr>
<tr>
<td>MDEI</td>
<td>001304</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>OBSOLETE. Mem diag inhibit interleave.</td>
</tr>
<tr>
<td>MDID</td>
<td>001305</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>OBSOLETE. Mem diag inhibit interleave.</td>
</tr>
<tr>
<td>MDIW</td>
<td>001324</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>OBSOLETE. Mem diag inhibit interleave.</td>
</tr>
<tr>
<td>MDRS</td>
<td>001306</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>OBSOLETE. Mem diag inhibit interleave.</td>
</tr>
<tr>
<td>MDCW</td>
<td>001307</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>OBSOLETE. Mem diag inhibit interleave.</td>
</tr>
<tr>
<td>MH</td>
<td>124---</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Multiply halfword. RH * [EA]16 =&gt; R. (R)</td>
</tr>
<tr>
<td>MIA</td>
<td>150---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>OBSOLETE. Microcode execute A.</td>
</tr>
<tr>
<td>MIA</td>
<td>-25404</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>OBSOLETE. Microcode execute A.</td>
</tr>
<tr>
<td>MIA</td>
<td>170---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>OBSOLETE. Microcode execute B.</td>
</tr>
<tr>
<td>MIB</td>
<td>-27404</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>OBSOLETE. Microcode execute B.</td>
</tr>
<tr>
<td>MPL</td>
<td>-35414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Multiply long. L * [EA]32 =&gt; L,E.</td>
</tr>
<tr>
<td>N</td>
<td>006---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>And. AND(R, [EA]32) =&gt; R. (R)</td>
</tr>
<tr>
<td>NFYB</td>
<td>001211</td>
<td>AP</td>
<td>6</td>
<td>5</td>
<td>VI*</td>
<td>Notify on sem at AP. LIFO Q.</td>
</tr>
<tr>
<td>NFYE</td>
<td>001210</td>
<td>AP</td>
<td>6</td>
<td>5</td>
<td>VI*</td>
<td>Notify on sem at AP. FIFO Q.</td>
</tr>
<tr>
<td>Mnem</td>
<td>OpCode</td>
<td>Typ</td>
<td>C</td>
<td>cc</td>
<td>Modes</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>-----</td>
<td>---</td>
<td>----</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>NH</td>
<td>026---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>And halfword. AND(RH, [EA]16) =&gt; RH. (RI)</td>
</tr>
<tr>
<td>NOP</td>
<td>000001</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRVI</td>
<td>No operation.</td>
</tr>
<tr>
<td>NOP</td>
<td>101000</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>No operation (faster on certain machines);</td>
</tr>
<tr>
<td>NRM</td>
<td>000101</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>OBSOLETE. Normalize A,B as on P300.</td>
</tr>
<tr>
<td>O</td>
<td>046---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Or. OR(R, [EA]32) =&gt; R. (RI)</td>
</tr>
<tr>
<td>OCP</td>
<td>030---</td>
<td>PIO</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Output control pulse.</td>
</tr>
<tr>
<td>OH</td>
<td>006---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Or halfword. OR(RH, [EA]16) =&gt; RH. (RI)</td>
</tr>
<tr>
<td>ORA</td>
<td>-07410</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Or. OR(A, [EA]16) =&gt; A.</td>
</tr>
<tr>
<td>OTA</td>
<td>170---</td>
<td>PIO</td>
<td>-</td>
<td>-</td>
<td>SR*</td>
<td>Output from A.</td>
</tr>
<tr>
<td>OTK</td>
<td>050071</td>
<td>RGN</td>
<td>7</td>
<td>6</td>
<td>I</td>
<td>Output keys from RH. [RH] =&gt; KEYS.</td>
</tr>
<tr>
<td>OTP</td>
<td>000405</td>
<td>GEN</td>
<td>7</td>
<td>6</td>
<td>SR</td>
<td>Output A to P300 KEYS &amp; S. (TAK in V-mode)</td>
</tr>
<tr>
<td>PCL</td>
<td>1142--</td>
<td>MR</td>
<td>6</td>
<td>5</td>
<td>I</td>
<td>Procedure call.</td>
</tr>
<tr>
<td>PCL</td>
<td>-21410</td>
<td>MR</td>
<td>6</td>
<td>5</td>
<td>V</td>
<td>Procedure call.</td>
</tr>
<tr>
<td>PID</td>
<td>060052</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Pos for int divide. R =&gt; R+1; w/ sign extend.</td>
</tr>
<tr>
<td>PID</td>
<td>000211</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>Pos for divide. A =&gt; L w/ sign ext. &amp; hole.</td>
</tr>
<tr>
<td>PIDA</td>
<td>000115</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Pos for int divide. A =&gt; L w/ sign extend.</td>
</tr>
<tr>
<td>PIDH</td>
<td>060053</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Pos RH for div. RH =&gt; RL; RH(1) =&gt; RH(2-16).</td>
</tr>
<tr>
<td>PIDL</td>
<td>000305</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Pos for long divide. L =&gt; E w/ sign extend.</td>
</tr>
<tr>
<td>PIM</td>
<td>060050</td>
<td>RGN</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Pos after int multiply. (R+1) =&gt; R.</td>
</tr>
<tr>
<td>PIM</td>
<td>000205</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>Pos after mult. B(2-16) =&gt; A(2-16)</td>
</tr>
<tr>
<td>PIMA</td>
<td>000015</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Pos after mult. L =&gt; A.</td>
</tr>
<tr>
<td>PIMH</td>
<td>060051</td>
<td>RGN</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Pos RH after int multiply. RL =&gt; RH.</td>
</tr>
<tr>
<td>PIML</td>
<td>000301</td>
<td>GEN</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Pos after mult long. (L,E) =&gt; L.</td>
</tr>
<tr>
<td>PRTN</td>
<td>000611</td>
<td>CON</td>
<td>7</td>
<td>-</td>
<td>VI</td>
<td>Procedure return.</td>
</tr>
<tr>
<td>PTLB</td>
<td>000064</td>
<td>MOD</td>
<td>6</td>
<td>5</td>
<td>V*</td>
<td>Purge TLB (non-IO). L, R2, R3. (CRE first)</td>
</tr>
<tr>
<td>QFAD</td>
<td>0754--</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Quad flt add. QAC + [EA]12 =&gt; QAC.</td>
</tr>
<tr>
<td>QFAD</td>
<td>-13410</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Quad flt add. QAC + [EA]12 =&gt; QAC. (Ext: 2)</td>
</tr>
<tr>
<td>QFC</td>
<td>1156--</td>
<td>QAD</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Quad floating compare QAF to [EA]12. (RI)</td>
</tr>
<tr>
<td>QFCM</td>
<td>140570</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Quad flt complement. -QAC =&gt; QAC.</td>
</tr>
<tr>
<td>QFCS</td>
<td>-13410</td>
<td>QAD</td>
<td>6</td>
<td>5</td>
<td>V</td>
<td>Skip 0,1,2 if QAC =&gt; &lt; [EA]28. (Ext: 6)</td>
</tr>
<tr>
<td>QFDV</td>
<td>1154--</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Quad flt divide. QAC / [EA]12 =&gt; QAC.</td>
</tr>
<tr>
<td>QFDV</td>
<td>-13410</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Quad flt divide. QAC / [EA]12 =&gt; QAC. (Ext: 5)</td>
</tr>
<tr>
<td>QFLD</td>
<td>0750--</td>
<td>QAD</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Quad flt load. [EA]12/128 =&gt; QAC.</td>
</tr>
<tr>
<td>QFLD</td>
<td>-13410</td>
<td>QAD</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Quad flt load. [EA]12/128 =&gt; QAC. (Ext: 0)</td>
</tr>
<tr>
<td>QFLX</td>
<td>-33414</td>
<td>QAD</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Quad flt load index. [EA]*8 =&gt; X,Y (No X)</td>
</tr>
<tr>
<td>QFMP</td>
<td>1152--</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Quad flt multiply. QAC * [EA]12 =&gt; QAC.</td>
</tr>
<tr>
<td>QFMP</td>
<td>-13410</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Quad flt mpy. QAC * [EA]12 =&gt; QAC. (Ext: 4)</td>
</tr>
<tr>
<td>QFSB</td>
<td>0756--</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Quad flt subtract. QAC - [EA]12 =&gt; QAC.</td>
</tr>
<tr>
<td>QFSB</td>
<td>-13410</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Quad flt sub. QAC - [EA]12 =&gt; QAC. (Ext: 3)</td>
</tr>
<tr>
<td>QFST</td>
<td>0752--</td>
<td>QAD</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Quad flt store. QAC =&gt; [EA]128.</td>
</tr>
<tr>
<td>QFST</td>
<td>-13410</td>
<td>QAD</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Quad flt store. QAC =&gt; [EA]128. (Ext: 1)</td>
</tr>
</tbody>
</table>

7-12 Prime Restricted
<table>
<thead>
<tr>
<th>Mnem</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QINQ</td>
<td>140572</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Convert quad to integer.</td>
</tr>
<tr>
<td>QIQR</td>
<td>140573</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Convert quad to integer rounded.</td>
</tr>
<tr>
<td>RBQ</td>
<td>060133</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>I</td>
<td>Remove from bottom of Q. emp -&gt; 0 =&gt; RH, CCEO</td>
</tr>
<tr>
<td>RBQ</td>
<td>141715</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>V</td>
<td>Remove from bottom of Q. emp -&gt; 0 =&gt; A, CCEO</td>
</tr>
<tr>
<td>RCB</td>
<td>140200</td>
<td>GEN</td>
<td>9</td>
<td>-</td>
<td>SRVI</td>
<td>Reset CBIT. 0 =&gt; CBIT.</td>
</tr>
<tr>
<td>RMC</td>
<td>000021</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Reset machine check flag.</td>
</tr>
<tr>
<td>RMP</td>
<td>000021</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>OBSOLETE. Reset machine check flag. (Use RMC)</td>
</tr>
<tr>
<td>RSR</td>
<td>000717</td>
<td>AP</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Restore registers (GEN, FLT, XB).</td>
</tr>
<tr>
<td>RSAV</td>
<td>000715</td>
<td>AP</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Save registers (GEN, FLT, XB).</td>
</tr>
<tr>
<td>RTN</td>
<td>000105</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>OBSOLETE. Return from P300 recur proc.</td>
</tr>
<tr>
<td>RTQ</td>
<td>060132</td>
<td>AP</td>
<td>7</td>
<td>-</td>
<td>I</td>
<td>Remove from top of Q. empty -&gt; 0 =&gt; RH, CCEO</td>
</tr>
<tr>
<td>RTQ</td>
<td>141714</td>
<td>AP</td>
<td>7</td>
<td>-</td>
<td>V</td>
<td>Remove from top of Q. empty -&gt; 0 =&gt; A, CCEO</td>
</tr>
<tr>
<td>RTS</td>
<td>005511</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>Reset time slice with A, R2.</td>
</tr>
<tr>
<td>S1A</td>
<td>140110</td>
<td>GEN</td>
<td>2</td>
<td>-</td>
<td>1 SRV</td>
<td>Subtract 1 from A. A - 1 =&gt; A.</td>
</tr>
<tr>
<td>S2A</td>
<td>140310</td>
<td>GEN</td>
<td>2</td>
<td>-</td>
<td>1 SRV</td>
<td>Subtract 2 from A. A - 2 =&gt; A.</td>
</tr>
<tr>
<td>SAR</td>
<td>10026-</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A(n) reset.</td>
</tr>
<tr>
<td>SAS</td>
<td>10126-</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A(n) set.</td>
</tr>
<tr>
<td>SBL</td>
<td>-17414</td>
<td>MR</td>
<td>2</td>
<td>1</td>
<td>V</td>
<td>Subtract long. L - [EA32] =&gt; L.</td>
</tr>
<tr>
<td>SCA</td>
<td>000041</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>OBSOLETE. Load P300 shift count into A.</td>
</tr>
<tr>
<td>SCB</td>
<td>140600</td>
<td>GEN</td>
<td>5</td>
<td>-</td>
<td>SRVI</td>
<td>Set CBIT. 1 =&gt; CBIT.</td>
</tr>
<tr>
<td>SCC</td>
<td>132---</td>
<td>MGR</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Store character via char pointer. (32IX)</td>
</tr>
<tr>
<td>SEQ</td>
<td>100040</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Skip if A.EQ. 0. (Use SZE)</td>
</tr>
<tr>
<td>SGE</td>
<td>100400</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Skip if A.LE. GE. 0. (Use SLP)</td>
</tr>
<tr>
<td>SGT</td>
<td>000005</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Enter single-precision mode.</td>
</tr>
<tr>
<td>SHL</td>
<td>100220</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A.GT. 0.</td>
</tr>
<tr>
<td>SHL1</td>
<td>060076</td>
<td>RGN</td>
<td>4</td>
<td>-</td>
<td>I</td>
<td>Logical shift. Shift(R,[EA16] =&gt; R).</td>
</tr>
<tr>
<td>SHL2</td>
<td>060077</td>
<td>RGN</td>
<td>4</td>
<td>-</td>
<td>I</td>
<td>Shift halfword left 1. LS(RH, 1) =&gt; RH.</td>
</tr>
<tr>
<td>SHR1</td>
<td>060120</td>
<td>RGN</td>
<td>4</td>
<td>-</td>
<td>I</td>
<td>Shift halfword left 2. LS(RH, 2) =&gt; RH.</td>
</tr>
<tr>
<td>SHR2</td>
<td>060121</td>
<td>RGN</td>
<td>4</td>
<td>-</td>
<td>I</td>
<td>Shift halfword right 1. RS(RH, 1) =&gt; RH.</td>
</tr>
<tr>
<td>SKP</td>
<td>100000</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip one word.</td>
</tr>
<tr>
<td>SKS</td>
<td>070---</td>
<td>PIO</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>Skip if condition set.</td>
</tr>
<tr>
<td>SL1</td>
<td>060072</td>
<td>RGN</td>
<td>4</td>
<td>1</td>
<td>I</td>
<td>Shift halfword left 1. LS(RH, 1) =&gt; R.</td>
</tr>
<tr>
<td>SL2</td>
<td>060073</td>
<td>RGN</td>
<td>4</td>
<td>1</td>
<td>I</td>
<td>Shift halfword left 2. LS(RH, 2) =&gt; R.</td>
</tr>
<tr>
<td>SLE</td>
<td>101220</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A.LE. 0.</td>
</tr>
<tr>
<td>SLN</td>
<td>101100</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A.bit 16 set.</td>
</tr>
<tr>
<td>SLT</td>
<td>101400</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Skip if A.LT. 0. (Use SM)</td>
</tr>
<tr>
<td>SLZ</td>
<td>101000</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A.bit 16.EQ. 0.</td>
</tr>
<tr>
<td>SMCR</td>
<td>100200</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if machine check reset.</td>
</tr>
<tr>
<td>SMCS</td>
<td>101200</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if machine check set.</td>
</tr>
<tr>
<td>SMK</td>
<td>170002</td>
<td>PIO</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>OBSOLETE. Set interrupt masks. (P100-P300)</td>
</tr>
<tr>
<td>SNE</td>
<td>101040</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Skip if A.NE. 0. (Use SNZ)</td>
</tr>
<tr>
<td>SNR</td>
<td>10024-</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV*</td>
<td>OBSOLETE. Skip if sense switch N set.</td>
</tr>
<tr>
<td>SNS</td>
<td>10124-</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV*</td>
<td>OBSOLETE. Skip if sense switch N set.</td>
</tr>
<tr>
<td>SNZ</td>
<td>101040</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Subtract 1 from A. A - 1 =&gt; A (Use S1A)</td>
</tr>
<tr>
<td>SOA</td>
<td>140110</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>SRV</td>
<td>OBSOLETE. Subtract 1 from A. A - 1 =&gt; A (Use S1A)</td>
</tr>
<tr>
<td>SPL</td>
<td>100400</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A.GE. 0.</td>
</tr>
<tr>
<td>Mnem</td>
<td>OpCode</td>
<td>Typ</td>
<td>C</td>
<td>cc</td>
<td>Modes</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>-----</td>
<td>---</td>
<td>---</td>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SPN</td>
<td>100200</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV*</td>
<td>OBSOLETE. Skip if machine check reset. (Use SMCR)</td>
</tr>
<tr>
<td>SPS</td>
<td>101200</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Skip if machine check set. (Use SMCS)</td>
</tr>
<tr>
<td>SR1</td>
<td>060074</td>
<td>RGN</td>
<td>4</td>
<td>-</td>
<td>SRV*</td>
<td>Shift halfword right 1. RS(RH, 1) =&gt; R.</td>
</tr>
<tr>
<td>SR1</td>
<td>100020</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Skip if sense switch 1 reset.</td>
</tr>
<tr>
<td>SR2</td>
<td>060075</td>
<td>RGN</td>
<td>4</td>
<td>-</td>
<td>SRV*</td>
<td>Shift halfword right 2. RS(RH, 2) =&gt; R.</td>
</tr>
<tr>
<td>SR2</td>
<td>100010</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Skip if sense switch 2 reset.</td>
</tr>
<tr>
<td>SR3</td>
<td>100004</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV*</td>
<td>OBSOLETE. Skip if sense switch 3 reset.</td>
</tr>
<tr>
<td>SR4</td>
<td>100002</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Skip if sense switch 4 reset.</td>
</tr>
<tr>
<td>SRC</td>
<td>100001</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if CBIT reset.</td>
</tr>
<tr>
<td>SS1</td>
<td>101020</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV*</td>
<td>OBSOLETE. Skip if sense switch 1 set.</td>
</tr>
<tr>
<td>SS2</td>
<td>101010</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Skip if sense switch 2 set.</td>
</tr>
<tr>
<td>SS3</td>
<td>101004</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV*</td>
<td>OBSOLETE. Skip if sense switch 3 set.</td>
</tr>
<tr>
<td>SS4</td>
<td>101002</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Skip if sense switch 4 set.</td>
</tr>
<tr>
<td>SCC</td>
<td>101001</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if CBIT set.</td>
</tr>
<tr>
<td>SSM</td>
<td>060042</td>
<td>RGN</td>
<td>-</td>
<td>I</td>
<td>SRV</td>
<td>Set sign minus. 1 =&gt; R(1).</td>
</tr>
<tr>
<td>SSM</td>
<td>140500</td>
<td>GEN</td>
<td>1</td>
<td>-</td>
<td>V</td>
<td>Set sign of A minus. 0 =&gt; A(1).</td>
</tr>
<tr>
<td>SSM</td>
<td>060043</td>
<td>RGN</td>
<td>-</td>
<td>I</td>
<td>SRV</td>
<td>Set sign of A plus. 0 =&gt; R(1).</td>
</tr>
<tr>
<td>SSP</td>
<td>140100</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Set sign of A plus. 0 =&gt; A(1).</td>
</tr>
<tr>
<td>SSR</td>
<td>100036</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV*</td>
<td>OBSOLETE. Skip if all sense switches set.</td>
</tr>
<tr>
<td>SSS</td>
<td>101036</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Skip if all sense switches set.</td>
</tr>
<tr>
<td>SSSN</td>
<td>040310</td>
<td>GEN</td>
<td>6</td>
<td>5</td>
<td>VI</td>
<td>Store system serial number =&gt; [XB]16 halfwords.</td>
</tr>
<tr>
<td>STAC</td>
<td>001200</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>V</td>
<td>Store A if B = [EA]16 (=&gt; CCEQ).</td>
</tr>
<tr>
<td>STAR</td>
<td>130----</td>
<td>MR</td>
<td>-</td>
<td>5</td>
<td>I(*)</td>
<td>Store address register.</td>
</tr>
<tr>
<td>STC</td>
<td>060166</td>
<td>FLD</td>
<td>-</td>
<td>7</td>
<td>V</td>
<td>Store character from RH.</td>
</tr>
<tr>
<td>STC</td>
<td>001322</td>
<td>FLD</td>
<td>-</td>
<td>7</td>
<td>V</td>
<td>Store char from A via FAR.</td>
</tr>
<tr>
<td>STEX</td>
<td>060027</td>
<td>RAP</td>
<td>6</td>
<td>5</td>
<td>V</td>
<td>Stack extend by R.</td>
</tr>
<tr>
<td>STEX</td>
<td>001315</td>
<td>GEN</td>
<td>6</td>
<td>5</td>
<td>V</td>
<td>Stack extend. Extent in L.</td>
</tr>
<tr>
<td>STG</td>
<td>006----</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Store long. L =&gt; [EA]32.</td>
</tr>
<tr>
<td>STLC</td>
<td>001204</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>V</td>
<td>Store L if E = [EA]32 (=&gt; CCEQ).</td>
</tr>
<tr>
<td>STLR</td>
<td>-07404</td>
<td>MR</td>
<td>-</td>
<td>5</td>
<td>V(*)</td>
<td>Store long into register(EA).</td>
</tr>
<tr>
<td>STPM</td>
<td>000024</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Store processor model via XB.</td>
</tr>
<tr>
<td>STM</td>
<td>000510</td>
<td>MOD</td>
<td>6</td>
<td>5</td>
<td>VI</td>
<td>Store process timer at XB. (48 bit)</td>
</tr>
<tr>
<td>STY</td>
<td>-73410</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Store Y, Y =&gt; [EA]16. (No X)</td>
</tr>
<tr>
<td>SVC</td>
<td>000505</td>
<td>CON</td>
<td>-</td>
<td>-</td>
<td>SRVI</td>
<td>Supervisor call.</td>
</tr>
<tr>
<td>SZE</td>
<td>100400</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A. Eq. 0.</td>
</tr>
<tr>
<td>TAC</td>
<td>001015</td>
<td>GEN</td>
<td>7</td>
<td>6</td>
<td>V</td>
<td>Transfer A to B. A =&gt; B.</td>
</tr>
<tr>
<td>TAK</td>
<td>140314</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Transfer A to KEYS.</td>
</tr>
<tr>
<td>TAX</td>
<td>140504</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Transfer A to X. A =&gt; X.</td>
</tr>
<tr>
<td>TAY</td>
<td>140505</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Transfer A to Y. A =&gt; Y.</td>
</tr>
<tr>
<td>TBA</td>
<td>140604</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Transfer B to A. B =&gt; A.</td>
</tr>
<tr>
<td>TC</td>
<td>060046</td>
<td>RGN</td>
<td>3</td>
<td>1</td>
<td>I</td>
<td>Two's complement R. -R =&gt; R.</td>
</tr>
<tr>
<td>TCA</td>
<td>140407</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>SRV</td>
<td>Two's complement A. -A =&gt; A.</td>
</tr>
</tbody>
</table>
### Mnen OpCode Typ C cc Modes Description

<table>
<thead>
<tr>
<th>Mnen</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCH</td>
<td>060047</td>
<td>RGN</td>
<td>3</td>
<td>1</td>
<td>I</td>
<td>Two’s complement RH. -RH =&gt; RH.</td>
</tr>
<tr>
<td>TCL</td>
<td>141210</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>V</td>
<td>Two’s complement L. -L =&gt; L.</td>
</tr>
<tr>
<td>TCNP</td>
<td>1754--</td>
<td>MGR</td>
<td>-</td>
<td>1</td>
<td>V</td>
<td>Test for C null pointer. (32IX, R)</td>
</tr>
<tr>
<td>TFLL</td>
<td>001323</td>
<td>FLD</td>
<td>-</td>
<td></td>
<td>I</td>
<td>Transfer FLR to L.</td>
</tr>
<tr>
<td>TFLLR</td>
<td>080163</td>
<td>FLEN</td>
<td>-</td>
<td></td>
<td>V</td>
<td>Transfer FLR to R.</td>
</tr>
<tr>
<td>TKA</td>
<td>001005</td>
<td>GEN</td>
<td>-</td>
<td></td>
<td>V</td>
<td>Transfer KEYS to A.</td>
</tr>
<tr>
<td>TLFL</td>
<td>001321</td>
<td>FLD</td>
<td>-</td>
<td></td>
<td>V</td>
<td>Transfer L to FLR.</td>
</tr>
<tr>
<td>TM</td>
<td>1150--</td>
<td>MR</td>
<td>-</td>
<td>1</td>
<td>I</td>
<td>Test memory. ((EA)32:0) =&gt; CC.</td>
</tr>
<tr>
<td>TMH</td>
<td>1350--</td>
<td>MR</td>
<td>-</td>
<td>1</td>
<td>I</td>
<td>Test memory halfword. ((EA)16:0) =&gt; CC.</td>
</tr>
<tr>
<td>TRFL</td>
<td>060165</td>
<td>FLD</td>
<td>-</td>
<td></td>
<td>I</td>
<td>Transfer R to FLR.</td>
</tr>
<tr>
<td>TSTQ</td>
<td>060104</td>
<td>AP</td>
<td>7</td>
<td></td>
<td>V</td>
<td>Test queue. # items =&gt; RH. empty =&gt; CCEQ.</td>
</tr>
<tr>
<td>TSTQ</td>
<td>141757</td>
<td>AP</td>
<td>7</td>
<td></td>
<td>V</td>
<td>Test queue. # items =&gt; A. empty =&gt; CCEQ.</td>
</tr>
<tr>
<td>TXA</td>
<td>141034</td>
<td>GEN</td>
<td>-</td>
<td></td>
<td>V</td>
<td>Transfer X to A. X =&gt; A.</td>
</tr>
<tr>
<td>TYA</td>
<td>141124</td>
<td>GEN</td>
<td>-</td>
<td></td>
<td>V</td>
<td>Transfer Y to A. Y =&gt; A.</td>
</tr>
<tr>
<td>VIRY</td>
<td>000311</td>
<td>IG</td>
<td>6</td>
<td>6</td>
<td>SRVI*</td>
<td>OBSCOLETE. Execute verification routine.</td>
</tr>
<tr>
<td>WAIT</td>
<td>000315</td>
<td>AP</td>
<td>-</td>
<td></td>
<td>VI*</td>
<td>Wait on semaphore at AP.</td>
</tr>
<tr>
<td>WCS</td>
<td>001500</td>
<td>IG</td>
<td>-</td>
<td></td>
<td>RVI*</td>
<td>OBSCOLETE. WCS entrances. Ull on no WCS.</td>
</tr>
<tr>
<td>X</td>
<td>1146--</td>
<td>MR</td>
<td>-</td>
<td>1</td>
<td>I</td>
<td>Exclusive OR. XOR(R, (EA)32) =&gt; R. (R1)</td>
</tr>
<tr>
<td>XAD</td>
<td>001100</td>
<td>DA</td>
<td>3</td>
<td>1</td>
<td>VI</td>
<td>Decimal add. FAR1 + FAR0 =&gt; FAR1.</td>
</tr>
<tr>
<td>XBTD</td>
<td>001145</td>
<td>DA</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Convert binary to decimal.</td>
</tr>
<tr>
<td>XCA</td>
<td>140104</td>
<td>GEN</td>
<td>-</td>
<td></td>
<td>SRV</td>
<td>Exchange &amp; clear A. A =&gt; B. 0 =&gt; A.</td>
</tr>
<tr>
<td>XCB</td>
<td>140204</td>
<td>GEN</td>
<td>-</td>
<td></td>
<td>SRV</td>
<td>Exchange &amp; clear B. B =&gt; A. 0 =&gt; B.</td>
</tr>
<tr>
<td>XCM</td>
<td>001102</td>
<td>DA</td>
<td>-</td>
<td>1</td>
<td>VI</td>
<td>Decimal compare.</td>
</tr>
<tr>
<td>XDTB</td>
<td>001146</td>
<td>DA</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Convert decimal to binary.</td>
</tr>
<tr>
<td>XDV</td>
<td>001107</td>
<td>DA</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Decimal divide. FAR1 / FAR0 =&gt; FAR1.</td>
</tr>
<tr>
<td>XEC</td>
<td>-03410</td>
<td>MR</td>
<td>-</td>
<td></td>
<td>RV</td>
<td>Execute instruction at EA.</td>
</tr>
<tr>
<td>XED</td>
<td>001112</td>
<td>DA</td>
<td>-</td>
<td></td>
<td>VI</td>
<td>Edit numeric field.</td>
</tr>
<tr>
<td>XH</td>
<td>1346--</td>
<td>MR</td>
<td>-</td>
<td></td>
<td>I</td>
<td>Excl. OR halfword. XOR(RH, (EA)16) =&gt; RH. (R1)</td>
</tr>
<tr>
<td>XMP</td>
<td>001104</td>
<td>DA</td>
<td>3</td>
<td>1</td>
<td>VI</td>
<td>Decimal multiply. FAR1 * FAR0 =&gt; FAR1.</td>
</tr>
<tr>
<td>XMV</td>
<td>001101</td>
<td>DA</td>
<td>3</td>
<td>1</td>
<td>VI</td>
<td>Decimal move.</td>
</tr>
<tr>
<td>XVRY</td>
<td>001113</td>
<td>IG</td>
<td>6</td>
<td>5</td>
<td>VI*</td>
<td>OBSCOLETE. Verify XIS board. (P500)</td>
</tr>
<tr>
<td>ZCM</td>
<td>001117</td>
<td>CS</td>
<td>6</td>
<td>7</td>
<td>VI</td>
<td>Compare char fields.</td>
</tr>
<tr>
<td>ZED</td>
<td>001111</td>
<td>CS</td>
<td>-</td>
<td></td>
<td>VII</td>
<td>Edit char field.</td>
</tr>
<tr>
<td>ZFIL</td>
<td>001116</td>
<td>CS</td>
<td>6</td>
<td>5</td>
<td>VI</td>
<td>Fill string with char. (A(9-16), R2(9-16))</td>
</tr>
<tr>
<td>ZM</td>
<td>106---</td>
<td>MR</td>
<td>-</td>
<td>1</td>
<td>I</td>
<td>Zero memory. 0 =&gt; (EA)32.</td>
</tr>
<tr>
<td>ZMH</td>
<td>126---</td>
<td>MR</td>
<td>-</td>
<td>1</td>
<td>I</td>
<td>Zero memory halfword. 0 =&gt; (EA)16.</td>
</tr>
<tr>
<td>ZMV</td>
<td>001114</td>
<td>CS</td>
<td>6</td>
<td>5</td>
<td>V</td>
<td>Copy char field, space fills.</td>
</tr>
<tr>
<td>ZMVD</td>
<td>001115</td>
<td>CS</td>
<td>6</td>
<td>5</td>
<td>V</td>
<td>Copy equal length char fields.</td>
</tr>
<tr>
<td>ZTRN</td>
<td>001110</td>
<td>CS</td>
<td>-</td>
<td></td>
<td>V</td>
<td>Copy and translate char string.</td>
</tr>
</tbody>
</table>

### 7.3. Instruction Set Grouped by Function

#### 7.3.1. Address Pointer Operations

<table>
<thead>
<tr>
<th>Mnen</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABQ</td>
<td>060134</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>I</td>
<td>Add to bottom of Q. CCEQ =&gt; FULL.</td>
</tr>
<tr>
<td>ABQ</td>
<td>141716</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>V</td>
<td>Add to bottom of Q. CCEQ =&gt; FULL.</td>
</tr>
<tr>
<td>ATQ</td>
<td>060135</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>I</td>
<td>Add to top of queue. RH =&gt; Q. CCEQ =&gt; FULL.</td>
</tr>
<tr>
<td>Mnem</td>
<td>OpCode</td>
<td>Typ</td>
<td>C</td>
<td>cc</td>
<td>Modes</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>-----</td>
<td>---</td>
<td>----</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>ATQ</td>
<td>141717</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>V</td>
<td>Add to top of queue. A =&gt; Q. CCEQ =&gt; FULL.</td>
</tr>
<tr>
<td>CALF</td>
<td>000705</td>
<td>AP</td>
<td>6</td>
<td>5</td>
<td>SRVI</td>
<td>Proc call from faulting proc.</td>
</tr>
<tr>
<td>EFA</td>
<td>001300</td>
<td>AP</td>
<td>-</td>
<td>V</td>
<td></td>
<td>Eff. addr to FAR.</td>
</tr>
<tr>
<td>INBC</td>
<td>001217</td>
<td>AP</td>
<td>6</td>
<td>5</td>
<td>VI</td>
<td>Interrupt nty LIFO, clear active interrupt.</td>
</tr>
<tr>
<td>INBN</td>
<td>001215</td>
<td>AP</td>
<td>6</td>
<td>5</td>
<td>VI</td>
<td>Interrupt nty LIFO.</td>
</tr>
<tr>
<td>INEC</td>
<td>001216</td>
<td>AP</td>
<td>6</td>
<td>5</td>
<td>VI</td>
<td>Interrupt nty FIFO, clear active interrupt.</td>
</tr>
<tr>
<td>INEN</td>
<td>001214</td>
<td>AP</td>
<td>6</td>
<td>5</td>
<td>VI</td>
<td>Interrupt nty FIFO.</td>
</tr>
<tr>
<td>LIOT</td>
<td>000044</td>
<td>AP</td>
<td>6</td>
<td>5</td>
<td>V</td>
<td>Load IOTLB. L, R2 =&gt; target virt addr.</td>
</tr>
<tr>
<td>LPSW</td>
<td>000711</td>
<td>AP</td>
<td>7</td>
<td>5</td>
<td>VI</td>
<td>Load PSW (SN,WN,KEYS,MODALS).</td>
</tr>
<tr>
<td>NFBY</td>
<td>001211</td>
<td>AP</td>
<td>6</td>
<td>5</td>
<td>VI</td>
<td>Notify on sem at AP. LIFO Q.</td>
</tr>
<tr>
<td>NFYE</td>
<td>001210</td>
<td>AP</td>
<td>6</td>
<td>5</td>
<td>VI</td>
<td>Notify on sem at AP. FIFO Q.</td>
</tr>
<tr>
<td>RBQ</td>
<td>060133</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>I</td>
<td>Remove from bottom of Q. emp -&gt; 0 =&gt; RH, CCEQ</td>
</tr>
<tr>
<td>RBQ</td>
<td>141715</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>V</td>
<td>Remove from bottom of Q. emp -&gt; 0 =&gt; A, CCEQ</td>
</tr>
<tr>
<td>RST</td>
<td>000717</td>
<td>AP</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Restore registers (GEN, FLT, XB).</td>
</tr>
<tr>
<td>RSAV</td>
<td>000715</td>
<td>AP</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Save registers (GEN, FLT, XB).</td>
</tr>
<tr>
<td>RTQ</td>
<td>060132</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>I</td>
<td>Remove from top of Q. empty =&gt; 0 =&gt; RH, CCEQ</td>
</tr>
<tr>
<td>RTQ</td>
<td>141714</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>V</td>
<td>Remove from top of Q. empty =&gt; 0 =&gt; A, CCEQ</td>
</tr>
<tr>
<td>STAC</td>
<td>001200</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>V</td>
<td>Store A if B = [EA]16 (-&gt; CCEQ).</td>
</tr>
<tr>
<td>STFA</td>
<td>001320</td>
<td>AP</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Store FAR.</td>
</tr>
<tr>
<td>STLC</td>
<td>001204</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>V</td>
<td>Store L if E = [EA]32 (-&gt; CCEQ).</td>
</tr>
<tr>
<td>TSTQ</td>
<td>060104</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>I</td>
<td>Test queue. # items =&gt; RH. empty =&gt; CCEQ.</td>
</tr>
<tr>
<td>TSTQ</td>
<td>141757</td>
<td>AP</td>
<td>-</td>
<td>7</td>
<td>V</td>
<td>Test queue. # items =&gt; A. empty =&gt; CCEQ.</td>
</tr>
<tr>
<td>WAIT</td>
<td>000315</td>
<td>AP</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Wait on semaphore at AP.</td>
</tr>
</tbody>
</table>

### 7.3.2. Branch Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BCEQ</td>
<td>141602</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on Condition Code .EQ.</td>
</tr>
<tr>
<td>BCGE</td>
<td>141605</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on Condition Code .GE.</td>
</tr>
<tr>
<td>BCGT</td>
<td>141601</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on Condition Code .GT.</td>
</tr>
<tr>
<td>BCLE</td>
<td>141600</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on Condition Code .LE.</td>
</tr>
<tr>
<td>BCLT</td>
<td>141604</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on Condition Code .LT.</td>
</tr>
<tr>
<td>BCNE</td>
<td>141603</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on Condition Code .NE.</td>
</tr>
<tr>
<td>BCR</td>
<td>141705</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on CBIT reset.</td>
</tr>
<tr>
<td>BCS</td>
<td>141704</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Branch on CBIT set.</td>
</tr>
<tr>
<td>BDX</td>
<td>140734</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Branch on decremented X.</td>
</tr>
<tr>
<td>BHY</td>
<td>140724</td>
<td>BR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Branch on decremented Y.</td>
</tr>
<tr>
<td>BEQ</td>
<td>140612</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>Branch on A .EQ. 0.</td>
</tr>
<tr>
<td>BFEQ</td>
<td>020122</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>Branch on FAC .EQ. 0.</td>
</tr>
<tr>
<td>BFEQ</td>
<td>141612</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>Branch on FAC .EQ. 0.</td>
</tr>
<tr>
<td>BFGE</td>
<td>020125</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>Branch on FAC .GE. 0.</td>
</tr>
<tr>
<td>BFGE</td>
<td>141615</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>Branch on FAC .GE. 0.</td>
</tr>
<tr>
<td>BFGT</td>
<td>020121</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>Branch on FAC .GT. 0.</td>
</tr>
<tr>
<td>BFGT</td>
<td>141611</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>Branch on FAC .GT. 0.</td>
</tr>
<tr>
<td>BFLE</td>
<td>020120</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>Branch on FAC .LE. 0.</td>
</tr>
<tr>
<td>BFLE</td>
<td>141610</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>Branch on FAC .LE. 0.</td>
</tr>
<tr>
<td>BFLE</td>
<td>020124</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>Branch on FAC .LT. 0.</td>
</tr>
<tr>
<td>BFLE</td>
<td>141614</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>Branch on FAC .LT. 0.</td>
</tr>
<tr>
<td>BFNE</td>
<td>020123</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>Branch on FAC .NE. 0.</td>
</tr>
<tr>
<td>BFNE</td>
<td>141613</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>Branch on FAC .NE. 0.</td>
</tr>
<tr>
<td>BGE</td>
<td>140615</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>Branch on A .GE. 0.</td>
</tr>
<tr>
<td>BGT</td>
<td>140611</td>
<td>BR</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>Branch on A .GT. 0.</td>
</tr>
</tbody>
</table>
### 7.3.3. Control Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARG1T</td>
<td>000605</td>
<td>CON</td>
<td>6</td>
<td>5</td>
<td>VI</td>
<td>Argument transfer (used with PCL).</td>
</tr>
<tr>
<td>HLT</td>
<td>000000</td>
<td>CON</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Halt computer operation.</td>
</tr>
<tr>
<td>RTC</td>
<td>000603</td>
<td>CON</td>
<td>7</td>
<td>6</td>
<td>VI*</td>
<td>Interrupt return, clear active intrpt.</td>
</tr>
<tr>
<td>IRIN</td>
<td>000601</td>
<td>CON</td>
<td>7</td>
<td>6</td>
<td>VI*</td>
<td>Interrupt return.</td>
</tr>
<tr>
<td>ITLB</td>
<td>000615</td>
<td>CON</td>
<td>6</td>
<td>5</td>
<td>VI*</td>
<td>Invalidate STLB entry, L, R2 = VADDR.</td>
</tr>
<tr>
<td>LPID</td>
<td>000617</td>
<td>CON</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>Procedure return.</td>
</tr>
<tr>
<td>PRTN</td>
<td>000611</td>
<td>CON</td>
<td>7</td>
<td>6</td>
<td>VI</td>
<td>Supervisor call.</td>
</tr>
</tbody>
</table>
### 7.3.4. Character String Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>Opcode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZCM</td>
<td>001117</td>
<td>CS</td>
<td>6</td>
<td>7</td>
<td>VI</td>
<td>Compare char fields.</td>
</tr>
<tr>
<td>ZED</td>
<td>001111</td>
<td>CS</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Edit char field.</td>
</tr>
<tr>
<td>ZFIL</td>
<td>001116</td>
<td>CS</td>
<td>6</td>
<td>5</td>
<td>VI</td>
<td>Fill string with char. (A(9-16), R2(9-16))</td>
</tr>
<tr>
<td>ZMV</td>
<td>001114</td>
<td>CS</td>
<td>6</td>
<td>5</td>
<td>VI</td>
<td>Copy char field, space fills.</td>
</tr>
<tr>
<td>ZMVD</td>
<td>001115</td>
<td>CS</td>
<td>6</td>
<td>5</td>
<td>VI</td>
<td>Copy equal length char fields.</td>
</tr>
<tr>
<td>ZTIN</td>
<td>001110</td>
<td>CS</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Copy and translate char string.</td>
</tr>
</tbody>
</table>

### 7.3.5. Decimal Arithmetic

<table>
<thead>
<tr>
<th>Mnem</th>
<th>Opcode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XAD</td>
<td>001100</td>
<td>DA</td>
<td>3</td>
<td>1</td>
<td>VI</td>
<td>Decimal add. FAR1 + FAR0 =&gt; FAR1.</td>
</tr>
<tr>
<td>XBTD</td>
<td>001145</td>
<td>DA</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Convert binary to decimal.</td>
</tr>
<tr>
<td>XCM</td>
<td>001102</td>
<td>DA</td>
<td>-</td>
<td>1</td>
<td>VI</td>
<td>Decimal compare.</td>
</tr>
<tr>
<td>XDTB</td>
<td>001146</td>
<td>DA</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Convert decimal to binary.</td>
</tr>
<tr>
<td>XDV</td>
<td>001107</td>
<td>DA</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Decimal divide. FAR1 / FAR0 =&gt; FAR1.</td>
</tr>
<tr>
<td>XED</td>
<td>001112</td>
<td>DA</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>Edit numeric field.</td>
</tr>
<tr>
<td>XMP</td>
<td>001104</td>
<td>DA</td>
<td>3</td>
<td>1</td>
<td>VI</td>
<td>Decimal multiply. FAR1 * FAR0 =&gt; FAR1.</td>
</tr>
<tr>
<td>XMV</td>
<td>001101</td>
<td>DA</td>
<td>3</td>
<td>1</td>
<td>VI</td>
<td>Decimal move.</td>
</tr>
</tbody>
</table>

### 7.3.6. Field Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>Opcode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALFA</td>
<td>001301</td>
<td>FLD</td>
<td>6</td>
<td>-</td>
<td>V</td>
<td>Add L to FAR.</td>
</tr>
<tr>
<td>ARFA</td>
<td>060161</td>
<td>FLD</td>
<td>6</td>
<td>-</td>
<td>I</td>
<td>Add R to FAR. FAR + R =&gt; FAR.</td>
</tr>
<tr>
<td>LDFL</td>
<td>060162</td>
<td>FLD</td>
<td>-</td>
<td>7</td>
<td>-</td>
<td>Load char to RH.</td>
</tr>
<tr>
<td>LDC</td>
<td>060163</td>
<td>FLD</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Load char to A via FAR.</td>
</tr>
<tr>
<td>LFLI</td>
<td>001303</td>
<td>FLD</td>
<td>-</td>
<td>-</td>
<td>SRVI</td>
<td>Load FLR immediate.</td>
</tr>
<tr>
<td>STC</td>
<td>060166</td>
<td>FLD</td>
<td>-</td>
<td>7</td>
<td>I</td>
<td>Store character from RH.</td>
</tr>
<tr>
<td>STC</td>
<td>001322</td>
<td>FLD</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Store char from A via FAR.</td>
</tr>
<tr>
<td>TFLF</td>
<td>001323</td>
<td>FLD</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Transfer FLR to L.</td>
</tr>
<tr>
<td>TFLR</td>
<td>060163</td>
<td>FLD</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Transfer L to FLR.</td>
</tr>
<tr>
<td>TRFL</td>
<td>001321</td>
<td>FLD</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Transfer R to FLR.</td>
</tr>
<tr>
<td>TRFL</td>
<td>060165</td>
<td>FLD</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Transfer R to FLR.</td>
</tr>
</tbody>
</table>

### 7.3.7. Floating-point Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>Opcode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBLE</td>
<td>060106</td>
<td>FOP</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Convert single to double fltg pt.</td>
</tr>
<tr>
<td>DFCM</td>
<td>060144</td>
<td>FOP</td>
<td>3</td>
<td>1</td>
<td>SRV</td>
<td>Dbl fltg complement. -DFAC =&gt; DFAC.</td>
</tr>
<tr>
<td>DFCM</td>
<td>140574</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>SRV</td>
<td>Dbl fltg complement. -DFAC =&gt; DFAC.</td>
</tr>
<tr>
<td>DRN</td>
<td>040300</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Double round from quad.</td>
</tr>
<tr>
<td>DRNM</td>
<td>140571</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Double round from quad to -infinity.</td>
</tr>
<tr>
<td>DRNP</td>
<td>040301</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Double round from quad to +infinity.</td>
</tr>
<tr>
<td>DRNZ</td>
<td>040302</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Double round from quad to 0.</td>
</tr>
<tr>
<td>FCDQ</td>
<td>140571</td>
<td>FOP</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Fltg convert dbl to quad. (P9950)</td>
</tr>
<tr>
<td>FCM</td>
<td>060100</td>
<td>FOP</td>
<td>3</td>
<td>1</td>
<td>I</td>
<td>Fltg complement. -FAC =&gt; FAC.</td>
</tr>
<tr>
<td>FRM</td>
<td>140530</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Fltg complement. -FAC =&gt; FAC.</td>
</tr>
<tr>
<td>FDBL</td>
<td>140016</td>
<td>FOP</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Fltg convert single to dbl. FAC =&gt; DFAC.</td>
</tr>
<tr>
<td>FLOT</td>
<td>140550</td>
<td>FOP</td>
<td>6</td>
<td>5</td>
<td>R</td>
<td>Convert int to fltg. Flot(A,B) =&gt; FAC w/whole.</td>
</tr>
</tbody>
</table>

---

7-18 Prime Restricted
<table>
<thead>
<tr>
<th>Mnem</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLT</td>
<td>060105</td>
<td>FOP</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Convert int to fltg. Fltg(R) =&gt; FAC.</td>
</tr>
<tr>
<td>FLT A</td>
<td>140532</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Convert int to fltg. Fltg(A) =&gt; FAC.</td>
</tr>
<tr>
<td>FLT H</td>
<td>060102</td>
<td>FOP</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Convert half word int to fltg pt.</td>
</tr>
<tr>
<td>FLT L</td>
<td>140535</td>
<td>FOP</td>
<td>5</td>
<td>-</td>
<td>R</td>
<td>Convert long to fltg. Fltg(L) =&gt; FAC.</td>
</tr>
<tr>
<td>FRN</td>
<td>060107</td>
<td>FOP</td>
<td>3</td>
<td>1</td>
<td>I</td>
<td>OBSOLETE. Fltg round. (FRN)</td>
</tr>
<tr>
<td>FRN</td>
<td>060146</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>RVI</td>
<td>Fltg round up.</td>
</tr>
<tr>
<td>FRNM</td>
<td>040320</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Fltg round towards - infinity.</td>
</tr>
<tr>
<td>FRNM</td>
<td>060145</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Fltg round towards + infinity.</td>
</tr>
<tr>
<td>FRNP</td>
<td>040303</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Fltg round towards + infinity.</td>
</tr>
<tr>
<td>FRNZ</td>
<td>060147</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Fltg round towards zero.</td>
</tr>
<tr>
<td>INT</td>
<td>060103</td>
<td>FOP</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Convert fltg to int. INT(FAC) =&gt; R.</td>
</tr>
<tr>
<td>INT A</td>
<td>140554</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>SR</td>
<td>Convert fltg to int. INT(FAC) =&gt; A,B w/ hole.</td>
</tr>
<tr>
<td>INT H</td>
<td>060101</td>
<td>FOP</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Convert fltg to halfword. INT(FAC) =&gt; RH.</td>
</tr>
<tr>
<td>INTL</td>
<td>140533</td>
<td>FOP</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Convert fltg to int long. INT(FAC) =&gt; L.</td>
</tr>
</tbody>
</table>

### 7.3.8. Floating-point Skip Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSST</td>
<td>140515</td>
<td>FSK</td>
<td>-</td>
<td>1</td>
<td>RV</td>
<td>Fltg skip if .GT. 0.</td>
</tr>
<tr>
<td>FSLE</td>
<td>140514</td>
<td>FSK</td>
<td>-</td>
<td>1</td>
<td>RV</td>
<td>Fltg skip if .LE. 0.</td>
</tr>
<tr>
<td>FSMI</td>
<td>140512</td>
<td>FSK</td>
<td>-</td>
<td>1</td>
<td>RV</td>
<td>Fltg skip if .LT. 0.</td>
</tr>
<tr>
<td>FSNZ</td>
<td>140511</td>
<td>FSK</td>
<td>-</td>
<td>1</td>
<td>RV</td>
<td>Fltg skip if .NE. 0.</td>
</tr>
<tr>
<td>FSPL</td>
<td>140513</td>
<td>FSK</td>
<td>-</td>
<td>1</td>
<td>RV</td>
<td>Fltg skip if .GE. 0.</td>
</tr>
<tr>
<td>FSZE</td>
<td>140510</td>
<td>FSK</td>
<td>-</td>
<td>1</td>
<td>RV</td>
<td>Fltg skip if .EQ. 0.</td>
</tr>
</tbody>
</table>

### 7.3.9. Generic Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1A</td>
<td>141206</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>SRV</td>
<td>Add 1 to A. A + 1 =&gt; A.</td>
</tr>
<tr>
<td>A2A</td>
<td>140304</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>SRV</td>
<td>Add 2 to A. A + 2 =&gt; A.</td>
</tr>
<tr>
<td>ACA</td>
<td>141216</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>SRV</td>
<td>Add CBIT to A. CBIT + A =&gt; A.</td>
</tr>
<tr>
<td>ADLL</td>
<td>141000</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>V</td>
<td>Add LINK to L.</td>
</tr>
<tr>
<td>AOA</td>
<td>141206</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>SRV</td>
<td>OBSOLETE. Add 1 to A. A + 1 =&gt; A. (Use A1A)</td>
</tr>
<tr>
<td>CAL</td>
<td>141050</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Clear left byte of A.</td>
</tr>
<tr>
<td>CAR</td>
<td>141044</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Clear right byte of A.</td>
</tr>
<tr>
<td>CAZ</td>
<td>140214</td>
<td>GEN</td>
<td>1</td>
<td>1</td>
<td>SRV</td>
<td>Skip 0,1,2 if A &gt;,&lt;,&lt; 0.</td>
</tr>
<tr>
<td>CEA</td>
<td>000111</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>Compute effective address. EA =&gt; A.</td>
</tr>
<tr>
<td>CGT</td>
<td>001314</td>
<td>GEN</td>
<td>6</td>
<td>5</td>
<td>V</td>
<td>Computed go to.</td>
</tr>
<tr>
<td>CHS</td>
<td>140024</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Change sign of A. ^A(1) =&gt; A(1).</td>
</tr>
<tr>
<td>GMA</td>
<td>140401</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>One's complement A. ^A =&gt; A.</td>
</tr>
<tr>
<td>CRA</td>
<td>140040</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Clear A. 0 =&gt; A.</td>
</tr>
<tr>
<td>CRB</td>
<td>140015</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Clear B. 0 =&gt; B.</td>
</tr>
<tr>
<td>CRB</td>
<td>140014</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Clears B &amp; LSW of DFACE(6). (Use CRB)</td>
</tr>
<tr>
<td>CRE</td>
<td>141404</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Clear E. 0 =&gt; E.</td>
</tr>
<tr>
<td>CRL</td>
<td>140010</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Clear L. 0 =&gt; L.</td>
</tr>
<tr>
<td>CRE</td>
<td>141410</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Clear L and E. 0 =&gt; L.</td>
</tr>
<tr>
<td>CSA</td>
<td>140320</td>
<td>GEN</td>
<td>5</td>
<td>-</td>
<td>SRV</td>
<td>Copy sign of A. A(1) =&gt; CBIT,0 =&gt; A(1).</td>
</tr>
<tr>
<td>DRX</td>
<td>140210</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Decrement X and skip if 0.</td>
</tr>
<tr>
<td>IAB</td>
<td>000201</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Exchange A and B. A =&gt; B &amp; B =&gt; A.</td>
</tr>
<tr>
<td>ICA</td>
<td>141340</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Interchange bytes of A.</td>
</tr>
<tr>
<td>Mnem</td>
<td>OpCode</td>
<td>Typ</td>
<td>C</td>
<td>cc</td>
<td>Modes</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>-----</td>
<td>---</td>
<td>----</td>
<td>-------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>ICL</td>
<td>141140</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Exchange bytes of A &amp; clh left.</td>
</tr>
<tr>
<td>ICR</td>
<td>141240</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Exchange bytes of A &amp; clh right.</td>
</tr>
<tr>
<td>ILE</td>
<td>141414</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Exchange L and E. L =&gt; E &amp; E =&gt; L.</td>
</tr>
<tr>
<td>INK</td>
<td>000043</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>Input P300 keys into A</td>
</tr>
<tr>
<td>IRX</td>
<td>140114</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Increment X and skip if 0.</td>
</tr>
<tr>
<td>NOP</td>
<td>000001</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td></td>
<td>No operation</td>
</tr>
<tr>
<td>NRM</td>
<td>000101</td>
<td>GEN</td>
<td>7</td>
<td>6</td>
<td>SR</td>
<td>Output A to P300 KEYS &amp; S. (TAK in V-mode)</td>
</tr>
<tr>
<td>PID</td>
<td>000211</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>Pos for divide. A =&gt; L w/ sign ext. &amp; hole.</td>
</tr>
<tr>
<td>PIDA</td>
<td>000115</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Pos for int divide. A =&gt; L w/ sign extend.</td>
</tr>
<tr>
<td>PIDL</td>
<td>000305</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Pos for long divide. L =&gt; E w/ sign extend.</td>
</tr>
<tr>
<td>PIM</td>
<td>000205</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>Pos after mult. B(2-16) =&gt; A(2-16)</td>
</tr>
<tr>
<td>PIMA</td>
<td>000015</td>
<td>GEN</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Pos after mult. L =&gt; A</td>
</tr>
<tr>
<td>PIML</td>
<td>000301</td>
<td>GEN</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Pos after mult long, (L,E) =&gt; L</td>
</tr>
<tr>
<td>RCB</td>
<td>140200</td>
<td>GEN</td>
<td>9</td>
<td>-</td>
<td>SRVI</td>
<td>Reset CBIT. 0 =&gt; CBIT.</td>
</tr>
<tr>
<td>RTN</td>
<td>000105</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>OBSOLETE. Return from P300 recur proc.</td>
</tr>
<tr>
<td>S1A</td>
<td>140110</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>SRV</td>
<td>Subtract 1 from A. A - 1 =&gt; A.</td>
</tr>
<tr>
<td>S2A</td>
<td>140310</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>SRV</td>
<td>Subtract 2 from A. A - 2 =&gt; A.</td>
</tr>
<tr>
<td>SCA</td>
<td>000041</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td></td>
<td>OBSOLETE. Subtract 1 from: A. A - 1 =&gt; A (Use S1A)</td>
</tr>
<tr>
<td>SCB</td>
<td>140600</td>
<td>GEN</td>
<td>5</td>
<td>-</td>
<td>SRVI</td>
<td>Set CBIT. 1 =&gt; CBIT.</td>
</tr>
<tr>
<td>SOA</td>
<td>140110</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>SRV</td>
<td>OBSOLETE. Subtract 1 from: A. A - 1 =&gt; A (Use S1A)</td>
</tr>
<tr>
<td>SSM</td>
<td>140500</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Set sign of A minus. 1 =&gt; A(1).</td>
</tr>
<tr>
<td>SSP</td>
<td>140100</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Set sign of A plus. 0 =&gt; A(1).</td>
</tr>
<tr>
<td>SSSN</td>
<td>004310</td>
<td>GEN</td>
<td>6</td>
<td>5</td>
<td>VI</td>
<td>Store system serial number =&gt; [XB]16 halfwords.</td>
</tr>
<tr>
<td>STEX</td>
<td>001315</td>
<td>GEN</td>
<td>6</td>
<td>5</td>
<td>V</td>
<td>Stack extend. Extent in L.</td>
</tr>
<tr>
<td>TAB</td>
<td>140314</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Transfer A to B. B =&gt; B.</td>
</tr>
<tr>
<td>TAK</td>
<td>001015</td>
<td>GEN</td>
<td>7</td>
<td>6</td>
<td>V</td>
<td>Transfer A to KEYS.</td>
</tr>
<tr>
<td>TAX</td>
<td>140504</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Transfer A to X. X =&gt; X.</td>
</tr>
<tr>
<td>TAY</td>
<td>140505</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Transfer A to Y. Y =&gt; Y.</td>
</tr>
<tr>
<td>TBA</td>
<td>140604</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Transfer B to A. B =&gt; A.</td>
</tr>
<tr>
<td>TCA</td>
<td>140407</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>SRV</td>
<td>Two's complement A. A =&gt; A.</td>
</tr>
<tr>
<td>TCL</td>
<td>141210</td>
<td>GEN</td>
<td>2</td>
<td>1</td>
<td>V</td>
<td>Two's complement L. L =&gt; L.</td>
</tr>
<tr>
<td>TKA</td>
<td>001005</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Transfer KEYS to A.</td>
</tr>
<tr>
<td>TXA</td>
<td>141034</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Transfer X to A. X =&gt; A.</td>
</tr>
<tr>
<td>TYA</td>
<td>141124</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Transfer Y to A. Y =&gt; A.</td>
</tr>
<tr>
<td>XCA</td>
<td>140104</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Exchange &amp; clear A. A =&gt; B, 0 =&gt; A.</td>
</tr>
<tr>
<td>XCB</td>
<td>140204</td>
<td>GEN</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Exchange &amp; clear B. B =&gt; A, 0 =&gt; B.</td>
</tr>
</tbody>
</table>

7.3.10. Integrity Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CXCS</td>
<td>001714</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>OBSOLETE. Control extended control store</td>
</tr>
<tr>
<td>EMCN</td>
<td>000503</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>OBSOLETE. Enter machine check mode.</td>
</tr>
<tr>
<td>LMCN</td>
<td>000501</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Leave machine check mode.</td>
</tr>
<tr>
<td>LWCS</td>
<td>001710</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>VI</td>
<td>OBSOLETE. Load writable control store.</td>
</tr>
<tr>
<td>MDEI</td>
<td>001304</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>OBSOLETE. Mem diag enable interleave.</td>
</tr>
<tr>
<td>MDII</td>
<td>001305</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>OBSOLETE. Mem diag inhibit interleave.</td>
</tr>
</tbody>
</table>

7-20 Prime Restricted
<table>
<thead>
<tr>
<th>Mnem</th>
<th>Opcode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDIW</td>
<td>001324</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>OBSOLETE. Mem diag write interleave.</td>
</tr>
<tr>
<td>MDRS</td>
<td>001306</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>OBSOLETE. Mem diag read syndrome bits.</td>
</tr>
<tr>
<td>MDWC</td>
<td>001307</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>OBSOLETE. Mem diag load write control reg.</td>
</tr>
<tr>
<td>RMC</td>
<td>000021</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Reset machine check flag.</td>
</tr>
<tr>
<td>RMP</td>
<td>000021</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>OBSOLETE. Reset machine check flag. (Use RMC)</td>
</tr>
<tr>
<td>VRYI</td>
<td>000311</td>
<td>IG</td>
<td>5</td>
<td>6</td>
<td>SRVI*</td>
<td>OBSOLETE. Execute verification routine.</td>
</tr>
<tr>
<td>WCS</td>
<td>001600</td>
<td>IG</td>
<td>-</td>
<td>-</td>
<td>RVI*</td>
<td>OBSOLETE. WCS entrances. Ull on no WCS.</td>
</tr>
<tr>
<td>XVRY</td>
<td>001113</td>
<td>IG</td>
<td>6</td>
<td>5</td>
<td>VI*</td>
<td>OBSOLETE. Verify XIS board. (P500)</td>
</tr>
</tbody>
</table>

### 7.3.11. Input/Output Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>Opcode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAI</td>
<td>000411</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>OBSOLETE. Clear active interrupt.</td>
</tr>
<tr>
<td>ENB</td>
<td>000401</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Enable interrupts.</td>
</tr>
<tr>
<td>ENBL</td>
<td>000401</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Enable interrupts (local). (P850)</td>
</tr>
<tr>
<td>ENBM</td>
<td>000400</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Enable interrupts (mutual). (P850)</td>
</tr>
<tr>
<td>ENBP</td>
<td>000402</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Enable interrupts (process). (P850)</td>
</tr>
<tr>
<td>INH</td>
<td>001001</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Inhibit interrupts.</td>
</tr>
<tr>
<td>INHL</td>
<td>001001</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Inhibit interrupts (local). (P850)</td>
</tr>
<tr>
<td>INHM</td>
<td>001000</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Inhibit interrupts (mutual). (P850)</td>
</tr>
<tr>
<td>INHP</td>
<td>001002</td>
<td>IO</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>Inhibit interrupts (process). (P850)</td>
</tr>
</tbody>
</table>

### 7.3.12. Logicize Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>Opcode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCEO</td>
<td>060153</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if EQ, CCEQ =&gt; RH.</td>
</tr>
<tr>
<td>LCQ</td>
<td>141503</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load A if EQ. CCEQ =&gt; A.</td>
</tr>
<tr>
<td>LGEO</td>
<td>060154</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if GE. CCGE =&gt; RH.</td>
</tr>
<tr>
<td>LGEO</td>
<td>141504</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load A if GE. CCGE =&gt; A.</td>
</tr>
<tr>
<td>LGEO</td>
<td>060155</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if GT. CCGT =&gt; RH.</td>
</tr>
<tr>
<td>LGEO</td>
<td>141505</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load A if GT. CCGT =&gt; A.</td>
</tr>
<tr>
<td>LGEO</td>
<td>060156</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if LE. CCLE =&gt; RH.</td>
</tr>
<tr>
<td>LGEO</td>
<td>141506</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load A if LE. CCLE =&gt; A.</td>
</tr>
<tr>
<td>LGEO</td>
<td>060150</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if LT. CCLT =&gt; RH.</td>
</tr>
<tr>
<td>LGEO</td>
<td>141500</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load A if LT. CCLT =&gt; A.</td>
</tr>
<tr>
<td>LCNE</td>
<td>060152</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if NE. CCNE =&gt; RH.</td>
</tr>
<tr>
<td>LCNE</td>
<td>141502</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load A if NE. CCNE =&gt; A.</td>
</tr>
<tr>
<td>LEQ</td>
<td>060003</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if R = 0. (R = 0) =&gt; RH.</td>
</tr>
<tr>
<td>LEQ</td>
<td>140413</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if FAC = 0. (FAC = 0) =&gt; RH.</td>
</tr>
<tr>
<td>LF</td>
<td>060016</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if FAC = 0. (FAC = 0) =&gt; RH.</td>
</tr>
<tr>
<td>LF</td>
<td>140416</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if FAC = 0. (FAC = 0) =&gt; RH.</td>
</tr>
<tr>
<td>LF</td>
<td>060023</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if FAC = 0. (FAC = 0) =&gt; RH.</td>
</tr>
<tr>
<td>LF</td>
<td>141113</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if FAC = 0. (FAC = 0) =&gt; RH.</td>
</tr>
<tr>
<td>LGEO</td>
<td>060024</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if FAC = 0. (FAC = 0) =&gt; RH.</td>
</tr>
<tr>
<td>LGEO</td>
<td>141114</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if FAC = 0. (FAC = 0) =&gt; RH.</td>
</tr>
<tr>
<td>LGEO</td>
<td>060025</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if FAC = 0. (FAC = 0) =&gt; RH.</td>
</tr>
<tr>
<td>LGEO</td>
<td>141115</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if FAC = 0. (FAC = 0) =&gt; RH.</td>
</tr>
<tr>
<td>LGEO</td>
<td>060021</td>
<td>LOG</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load RH if FAC = 0. (FAC = 0) =&gt; RH.</td>
</tr>
<tr>
<td>Mnem</td>
<td>Opcode</td>
<td>Typ</td>
<td>C</td>
<td>cc</td>
<td>Modes</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>------</td>
<td>---</td>
<td>----</td>
<td>-----------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>LFLE</td>
<td>141111</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If FAC.LE. 0.1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LFLT</td>
<td>060020</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if FAC &lt; 0. (FAC &lt; 0) =&gt; RH.</td>
</tr>
<tr>
<td>LFLT</td>
<td>141110</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If FAC.LT. 0.1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LFNE</td>
<td>060022</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if FAC &lt;= 0. (FAC &lt;= 0) =&gt; RH.</td>
</tr>
<tr>
<td>LFNE</td>
<td>141112</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If FAC.NE. 0.1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LGE</td>
<td>060004</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if R &gt; 0. (R &gt; 0) =&gt; RH.</td>
</tr>
<tr>
<td>LGE</td>
<td>140414</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>SRV</td>
<td>If A.GE. 0.1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LGE</td>
<td>060005</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if R &lt;= 0. (R &lt;= 0) =&gt; RH.</td>
</tr>
<tr>
<td>LGE</td>
<td>140415</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>SRV</td>
<td>If A.GT. 0.1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LGE</td>
<td>060013</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if R = 0. (R = 0) =&gt; RH.</td>
</tr>
<tr>
<td>LHEQ</td>
<td>060004</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if R &lt;= 0. (R &lt;= 0) =&gt; RH.</td>
</tr>
<tr>
<td>LHGT</td>
<td>060015</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if R &lt; 0. (R &lt; 0) =&gt; RH.</td>
</tr>
<tr>
<td>LHLE</td>
<td>060011</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if R &lt;= 0. (R &lt;= 0) =&gt; RH.</td>
</tr>
<tr>
<td>LHLT</td>
<td>060000</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if R &lt; 0. (R &lt; 0) =&gt; RH.</td>
</tr>
<tr>
<td>LNE</td>
<td>060012</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if R &lt; 0. (R &lt; 0) =&gt; RH.</td>
</tr>
<tr>
<td>LLE</td>
<td>060001</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load RH if R &lt;= 0. (R &lt;= 0) =&gt; RH.</td>
</tr>
<tr>
<td>LLE</td>
<td>140411</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>SRV</td>
<td>If A.LE. 0.1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LLE</td>
<td>141513</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If L.EQ. 0.1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LLGE</td>
<td>140414</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If L.GE. 0.1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LLGT</td>
<td>141515</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If L.GT. 0.1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LLLE</td>
<td>141511</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If L.LE. 0.1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LLLT</td>
<td>140410</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If L.LT. 0.1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LLNE</td>
<td>141512</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>V</td>
<td>If L.NE. 0.1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LLIT</td>
<td>060002</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load R if R &lt; 0. (R &lt; 0) =&gt; R.</td>
</tr>
<tr>
<td>LLLT</td>
<td>140410</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>SRV</td>
<td>If A.LT. 0.1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LLLT</td>
<td>060002</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>I</td>
<td>Load R if R &lt;= 0. (R &lt;= 0) =&gt; R.</td>
</tr>
<tr>
<td>LLLT</td>
<td>140412</td>
<td>LOG</td>
<td>-</td>
<td>4</td>
<td>SRV</td>
<td>If A.NE. 0.1 =&gt; A, else 0 =&gt; A.</td>
</tr>
<tr>
<td>LLLT</td>
<td>060017</td>
<td>LOG</td>
<td>-</td>
<td>5</td>
<td>I</td>
<td>Logic set true. 1 =&gt; R.</td>
</tr>
<tr>
<td>LLLT</td>
<td>140417</td>
<td>LOG</td>
<td>-</td>
<td>5</td>
<td>SRV</td>
<td>Logicize true. 1 =&gt; A.</td>
</tr>
</tbody>
</table>

7.3.13. Memory reference/General register to register

<table>
<thead>
<tr>
<th>Mnem</th>
<th>Opcode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACP</td>
<td>132-----</td>
<td>MGR</td>
<td>-</td>
<td>I</td>
<td>Add character pointer. (32IX, RI only, see SCC)</td>
<td></td>
</tr>
<tr>
<td>AIP</td>
<td>172-----</td>
<td>MGR</td>
<td>2</td>
<td>I</td>
<td>Add indirect pointer. (32IX)</td>
<td></td>
</tr>
<tr>
<td>LCC</td>
<td>112-----</td>
<td>MGR</td>
<td>-</td>
<td>I</td>
<td>Load character via char pointer. (32IX, RI)</td>
<td></td>
</tr>
<tr>
<td>LIP</td>
<td>152-----</td>
<td>MGR</td>
<td>-</td>
<td>I</td>
<td>Load indirect pointer. (32IX)</td>
<td></td>
</tr>
<tr>
<td>SCC</td>
<td>132-----</td>
<td>MGR</td>
<td>-</td>
<td>I</td>
<td>Store character via char pointer. (32IX)</td>
<td></td>
</tr>
<tr>
<td>TCNP</td>
<td>1754----</td>
<td>MGR</td>
<td>1</td>
<td>I</td>
<td>Test for C null pointer. (32IX, RI)</td>
<td></td>
</tr>
</tbody>
</table>

7.3.14. Mode Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>Opcode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBL</td>
<td>000007</td>
<td>MOD</td>
<td>-</td>
<td>SR</td>
<td>Enter double-prec mode.</td>
<td></td>
</tr>
<tr>
<td>DXA</td>
<td>000011</td>
<td>MOD</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Enter 16K sectored mode. (Use E16S)</td>
<td></td>
</tr>
<tr>
<td>E16S</td>
<td>000011</td>
<td>MOD</td>
<td>-</td>
<td>SRV</td>
<td>Enter P300 16K sectored mode.</td>
<td></td>
</tr>
<tr>
<td>E32I</td>
<td>000100</td>
<td>MOD</td>
<td>-</td>
<td>SRV</td>
<td>Enter P500 32I mode.</td>
<td></td>
</tr>
<tr>
<td>E32R</td>
<td>000103</td>
<td>MOD</td>
<td>-</td>
<td>SRV</td>
<td>Enter P300 32R relative mode.</td>
<td></td>
</tr>
<tr>
<td>E32S</td>
<td>000013</td>
<td>MOD</td>
<td>-</td>
<td>SRV</td>
<td>Enter P300 32S sectored mode.</td>
<td></td>
</tr>
<tr>
<td>ESIM</td>
<td>000101</td>
<td>MOD</td>
<td>-</td>
<td>SRV</td>
<td>Enter P300 64K relative mode.</td>
<td></td>
</tr>
<tr>
<td>E314V</td>
<td>000101</td>
<td>MOD</td>
<td>-</td>
<td>SRV</td>
<td>Enter P300 64K virtual mode.</td>
<td></td>
</tr>
<tr>
<td>F400</td>
<td>000011</td>
<td>MOD</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Enter standard interrupt mode.</td>
<td></td>
</tr>
</tbody>
</table>

7-22 Prime Restricted
### 7.3.15. Memory-reference Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVIM</td>
<td>000417</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>SRVI*</td>
<td>OBSOLETE. Enter vectored interrupt mode.</td>
</tr>
<tr>
<td>EXA</td>
<td>000013</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>SRVI</td>
<td>OBSOLETE. Enter 32K sectored mode. (Use E32S)</td>
</tr>
<tr>
<td>PTLB</td>
<td>000064</td>
<td>MOD</td>
<td>6</td>
<td>5</td>
<td>VI*</td>
<td>Purge TLB (non-IO). L, R2, R3. (CRE first)</td>
</tr>
<tr>
<td>RTS</td>
<td>000511</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>Reset time slice with A, R2.</td>
</tr>
<tr>
<td>SGL</td>
<td>000005</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>SR</td>
<td>Enter single-precision mode.</td>
</tr>
<tr>
<td>STPM</td>
<td>000024</td>
<td>MOD</td>
<td>-</td>
<td>-</td>
<td>VI*</td>
<td>Store processor model via XB.</td>
</tr>
<tr>
<td>STTM</td>
<td>000510</td>
<td>MOD</td>
<td>6</td>
<td>5</td>
<td>VI</td>
<td>Store process timer at XB. (48 bit)</td>
</tr>
</tbody>
</table>

### Description

- **SRVI**: Select vector interruption (SRVI) mode.
- **VI**: Vector interrupt (VI) mode.

**Note:** Further details and specific instructions for each mnemonic are provided in the Prime Engineering Handbook.
<table>
<thead>
<tr>
<th>Mnem</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVL</td>
<td>-37414</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Divide long. (L,E)[E]A32 =&gt; L; REM =&gt; E.</td>
</tr>
<tr>
<td>EAA</td>
<td>-03404</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>Eff. addr to L. EA =&gt; A.</td>
</tr>
<tr>
<td>EAL</td>
<td>-03404</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Eff. addr to L. EA =&gt; L.</td>
</tr>
<tr>
<td>EALB</td>
<td>1144--</td>
<td>MR</td>
<td>1</td>
<td>-</td>
<td>I</td>
<td>Eff. addr to LB. EA =&gt; LB.</td>
</tr>
<tr>
<td>EAR</td>
<td>27410</td>
<td>MR</td>
<td>1</td>
<td>-</td>
<td>V</td>
<td>Eff. addr to LB. EA =&gt; LB.</td>
</tr>
<tr>
<td>EAXB</td>
<td>146--</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Eff. addr to RB. EA =&gt; RB.</td>
</tr>
<tr>
<td>EAXB</td>
<td>1344--</td>
<td>MR</td>
<td>1</td>
<td>-</td>
<td>I</td>
<td>Eff. addr to X. EA =&gt; XB.</td>
</tr>
<tr>
<td>EAX</td>
<td>-25410</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Eff. addr to X. EA =&gt; XB.</td>
</tr>
<tr>
<td>EIO</td>
<td>070---</td>
<td>MR</td>
<td>7</td>
<td>1</td>
<td>I*</td>
<td>Execute EA as I/O inst. CCEQ =&gt; success.</td>
</tr>
<tr>
<td>ENTR</td>
<td>-31404</td>
<td>MR</td>
<td>-</td>
<td>7</td>
<td>V*</td>
<td>Execute EA as I/O inst. CCEQ =&gt; success.</td>
</tr>
<tr>
<td>EROA</td>
<td>-03414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>OBSOLETE. Enter recursive proc stack.</td>
</tr>
<tr>
<td>ERL</td>
<td>-13414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Exclusive or XOR(L[EA]E) =&gt; A.</td>
</tr>
<tr>
<td>FA</td>
<td>0350--</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Flt add. FAC + [E]A32 =&gt; FAC. (RI)</td>
</tr>
<tr>
<td>FAD</td>
<td>15404</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Flt add. FAC + [E]A32 =&gt; FAC. (RI)</td>
</tr>
<tr>
<td>FC</td>
<td>0150--</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Flt compare FAC with [E]A32. (RI)</td>
</tr>
<tr>
<td>FCS</td>
<td>-23404</td>
<td>MR</td>
<td>6</td>
<td>5</td>
<td>RV</td>
<td>Skip 0,1 if FAC =&gt; [E]A32. (RI)</td>
</tr>
<tr>
<td>FD</td>
<td>07400</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Flt divide. FAC / [E]A32 =&gt; FAC.</td>
</tr>
<tr>
<td>FDV</td>
<td>-37404</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Flt divide. FAC / [E]A32 =&gt; FAC.</td>
</tr>
<tr>
<td>FL</td>
<td>01400</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Flt load. [E]A32 =&gt; FAC. (RI)</td>
</tr>
<tr>
<td>FLG</td>
<td>-05404</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>RV</td>
<td>Flt load. [E]A32 =&gt; FAC. (RI)</td>
</tr>
<tr>
<td>FLX</td>
<td>-33404</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>RV</td>
<td>Load flt index. 2*[E]A16 =&gt; X. (No X)</td>
</tr>
<tr>
<td>FM</td>
<td>0550--</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Flt multiply. FAC * [E]A32 =&gt; FAC. (RI)</td>
</tr>
<tr>
<td>FMP</td>
<td>-35404</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Flt multiply. FAC * [E]A32 =&gt; FAC. (RI)</td>
</tr>
<tr>
<td>FS</td>
<td>05400</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Flt subtract. FAC - [E]A32 =&gt; FAC. (RI)</td>
</tr>
<tr>
<td>FSB</td>
<td>-17404</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>RV</td>
<td>Flt subtract. FAC - [E]A32 =&gt; FAC. (RI)</td>
</tr>
<tr>
<td>FST</td>
<td>13400</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Flt store. FAC =&gt; [E]A32.</td>
</tr>
<tr>
<td>IMA</td>
<td>-26---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Exchange memory and A. (Long: -27400)</td>
</tr>
<tr>
<td>IRS</td>
<td>-24---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Inc, replace, and skip if zero. (Long: -25400)</td>
</tr>
<tr>
<td>JDX</td>
<td>-33410</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>Decrement X &amp; jump if not zero. (No X)</td>
</tr>
<tr>
<td>JEQ</td>
<td>-05414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>OBSOLETE. Jump if A.EQ. 0, EA =&gt; P.</td>
</tr>
<tr>
<td>JGE</td>
<td>-17414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>OBSOLETE. Jump if A.GE. 0, EA =&gt; P.</td>
</tr>
<tr>
<td>JGT</td>
<td>-13414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>OBSOLETE. Jump if A.GT. 0, EA =&gt; P.</td>
</tr>
<tr>
<td>JX</td>
<td>-34144</td>
<td>MR</td>
<td>1</td>
<td>-</td>
<td>R</td>
<td>Increment X &amp; jump if not zero. (No X)</td>
</tr>
<tr>
<td>JLE</td>
<td>-11414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>OBSOLETE. Jump if A.LE. 0, EA =&gt; P.</td>
</tr>
<tr>
<td>JLT</td>
<td>-15414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>OBSOLETE. Jump if A.LT. 0, EA =&gt; P.</td>
</tr>
<tr>
<td>JMP</td>
<td>13422</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Jump. EA =&gt; P.</td>
</tr>
<tr>
<td>JNE</td>
<td>-07414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>R</td>
<td>OBSOLETE. Jump if A.NE. 0, EA =&gt; P.</td>
</tr>
<tr>
<td>JSR</td>
<td>166---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Jump to subr. P =&gt; RH, EA32 =&gt; P.</td>
</tr>
<tr>
<td>Mnem</td>
<td>Opcode</td>
<td>Typ</td>
<td>C</td>
<td>cc</td>
<td>Modes</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>-----</td>
<td>---</td>
<td>----</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>JSX</td>
<td>-73414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>RV</td>
<td>Jump &amp; save in X. P =&gt; X, EA =&gt; P. (No X)</td>
</tr>
<tr>
<td>JSXB</td>
<td>1542--</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Jump &amp; set XB. P =&gt; XB, EA =&gt; P.</td>
</tr>
<tr>
<td>JSXB</td>
<td>-31410</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Jump &amp; set XB. PB =&gt; XB, EA =&gt; PB.</td>
</tr>
<tr>
<td>L</td>
<td>002---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load R. [EA]32 =&gt; R. (RI)</td>
</tr>
<tr>
<td>LDA</td>
<td>110---</td>
<td>MR</td>
<td>5</td>
<td>5</td>
<td>I(*)</td>
<td>Load addressed register.</td>
</tr>
<tr>
<td>LDAR</td>
<td>-05414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Load long. [EA]32 =&gt; L.</td>
</tr>
<tr>
<td>LDL</td>
<td>-13404</td>
<td>MR</td>
<td>5</td>
<td>5</td>
<td>V(*)</td>
<td>Load long from addressed reg.</td>
</tr>
<tr>
<td>LDY</td>
<td>-37304</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Load Y. [EA]16 =&gt; Y. (No X)</td>
</tr>
<tr>
<td>LH</td>
<td>022---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load halfword. [EA]16 =&gt; RH. (RI)</td>
</tr>
<tr>
<td>LHL1</td>
<td>010---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load halfword shifted by 1. LS([EA]16,1) =&gt; RH. (RI)</td>
</tr>
<tr>
<td>LHL2</td>
<td>030---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load halfword shifted by 2. LS([EA]16,2) =&gt; RH. (RI)</td>
</tr>
<tr>
<td>LHL3</td>
<td>072---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Load halfword shifted by 3. LS([EA]16,3) =&gt; RH. (RI)</td>
</tr>
<tr>
<td>M</td>
<td>104---</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Multiply: R * [EA]32 =&gt; (R,R+1). (RI)</td>
</tr>
<tr>
<td>MH</td>
<td>124---</td>
<td>MR</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Multiply halfword. RH * [EA]16 =&gt; R. (RI)</td>
</tr>
<tr>
<td>MIA</td>
<td>150---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>OBSOLETE. Microcode execute A.</td>
</tr>
<tr>
<td>MIA</td>
<td>-25404</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>OBSOLETE. Microcode execute A.</td>
</tr>
<tr>
<td>MIB</td>
<td>170---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>OBSOLETE. Microcode execute B.</td>
</tr>
<tr>
<td>MIB</td>
<td>-27404</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>OBSOLETE. Microcode execute B.</td>
</tr>
<tr>
<td>MPL</td>
<td>-35414</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Multiply long. L * [EA]32 =&gt; L.</td>
</tr>
<tr>
<td>N</td>
<td>006---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>And. AND(R, [EA]32) =&gt; R. (RI)</td>
</tr>
<tr>
<td>NH</td>
<td>026---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>And halfword. AND(RH, [EA]16) =&gt; RH. (RI)</td>
</tr>
<tr>
<td>O</td>
<td>046---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Or. OR(R, [EA]32) =&gt; R. (RI)</td>
</tr>
<tr>
<td>OH</td>
<td>066---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Or halfword. OR(RH, [EA]16) =&gt; RH. (RI)</td>
</tr>
<tr>
<td>ORA</td>
<td>07410</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Or. OR(A, [EA]16) =&gt; A.</td>
</tr>
<tr>
<td>PCL</td>
<td>1142--</td>
<td>MR</td>
<td>6</td>
<td>5</td>
<td>I</td>
<td>Procedure call:</td>
</tr>
<tr>
<td>PCL</td>
<td>-21410</td>
<td>MR</td>
<td>6</td>
<td>5</td>
<td>V</td>
<td>Procedure call:</td>
</tr>
<tr>
<td>ROT</td>
<td>050---</td>
<td>MR</td>
<td>4</td>
<td>-</td>
<td>I</td>
<td>Rotate. Shift(R,[EA]16) =&gt; R.</td>
</tr>
<tr>
<td>S</td>
<td>044---</td>
<td>MR</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Subtract. R - [EA]32 =&gt; R. (RI)</td>
</tr>
<tr>
<td>SBL</td>
<td>-17414</td>
<td>MR</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Subtract long. L - [EA]32 =&gt; L.</td>
</tr>
<tr>
<td>SH</td>
<td>064---</td>
<td>MR</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Subtract halfword. RH - [EA]16 =&gt; RH. (RI)</td>
</tr>
<tr>
<td>SHA</td>
<td>032---</td>
<td>MR</td>
<td>4</td>
<td>-</td>
<td>I</td>
<td>Arithmetic shift. Shift(R,[EA]16) =&gt; R.</td>
</tr>
<tr>
<td>SHL</td>
<td>012---</td>
<td>MR</td>
<td>4</td>
<td>-</td>
<td>I</td>
<td>Logical shift. Shift(R,[EA]16) =&gt; R.</td>
</tr>
<tr>
<td>STH</td>
<td>062---</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>I(*)</td>
<td>Store addressed register.</td>
</tr>
<tr>
<td>STLR</td>
<td>-07404</td>
<td>MR</td>
<td>5</td>
<td>-</td>
<td>V(*)</td>
<td>Store long into register(EA).</td>
</tr>
<tr>
<td>STY</td>
<td>-73410</td>
<td>MR</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Store Y. Y =&gt; [EA]16. (No X)</td>
</tr>
</tbody>
</table>
### 7.3.16. Programmed I/O Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INA</td>
<td>130---</td>
<td>PIO</td>
<td>-</td>
<td>-</td>
<td>SR*</td>
<td>Input to A.</td>
</tr>
<tr>
<td>OCP</td>
<td>030---</td>
<td>PIO</td>
<td>-</td>
<td>-</td>
<td>SR*</td>
<td>Output control pulse.</td>
</tr>
<tr>
<td>OTA</td>
<td>170---</td>
<td>PIO</td>
<td>-</td>
<td>-</td>
<td>SR*</td>
<td>Output from A.</td>
</tr>
<tr>
<td>SKS</td>
<td>070---</td>
<td>PIO</td>
<td>-</td>
<td>-</td>
<td>SR*</td>
<td>Skip if condition set.</td>
</tr>
<tr>
<td>SMK</td>
<td>170020</td>
<td>PIO</td>
<td>-</td>
<td>-</td>
<td>SR*</td>
<td>OBSOLETE. Set interrupt masks. (P100-P300)</td>
</tr>
</tbody>
</table>

### 7.3.17. Quad Floating Point Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QFAD</td>
<td>0754--</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Quad fltg add. QAC + [EA]112 =&gt; QAC.</td>
</tr>
<tr>
<td>QFAD</td>
<td>-13410</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Quad fltg add. QAC + [EA]112 =&gt; QAC. (Ext: 2)</td>
</tr>
<tr>
<td>QFC</td>
<td>1156--</td>
<td>QAD</td>
<td>-</td>
<td>7</td>
<td>I</td>
<td>Quad floating compare QAF to [EA]112. (RI)</td>
</tr>
<tr>
<td>QFCM</td>
<td>140570</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Quad fltg complement. QAC =&gt; QAC</td>
</tr>
<tr>
<td>QFCS</td>
<td>-13410</td>
<td>QAD</td>
<td>6</td>
<td>5</td>
<td>I</td>
<td>Quad fltg divide. QAC / [EA]112 =&gt; QAC. (Ext: 6)</td>
</tr>
<tr>
<td>QFDV</td>
<td>1154--</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Quad fltg divide. QAC / [EA]112 =&gt; QAC.</td>
</tr>
<tr>
<td>QFDV</td>
<td>-13410</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Quad fltg divide. QAC / [EA]112 =&gt; QAC. (Ext: 5)</td>
</tr>
<tr>
<td>QFLD</td>
<td>0750--</td>
<td>QAD</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Quad fltg load. [EA]112/128 =&gt; QAC.</td>
</tr>
<tr>
<td>QFLD</td>
<td>-13410</td>
<td>QAD</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Quad fltg load. [EA]112/128 =&gt; QAC. (Ext: 0)</td>
</tr>
<tr>
<td>QFLX</td>
<td>-33414</td>
<td>QAD</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Quad fltg load index. [EA]*8 =&gt; X,Y. (No X)</td>
</tr>
<tr>
<td>QFMP</td>
<td>1152--</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Quad fltg multiply. QAC * [EA]112 =&gt; QAC.</td>
</tr>
<tr>
<td>QFMP</td>
<td>-13410</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Quad fltg mpy. QAC * [EA]112 =&gt; QAC. (Ext: 4)</td>
</tr>
<tr>
<td>QFSB</td>
<td>0756--</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Quad fltg subtruct. QAC - [EA]112 =&gt; QAC.</td>
</tr>
<tr>
<td>QFSB</td>
<td>-13410</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>V</td>
<td>Quad fltg sub. QAC - [EA]112 =&gt; QAC. (Ext: 3)</td>
</tr>
<tr>
<td>QFST</td>
<td>0752--</td>
<td>QAD</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Quad fltg store. QAC =&gt; [EA]128.</td>
</tr>
<tr>
<td>QFST</td>
<td>-13410</td>
<td>QAD</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>Quad fltg store. QAC =&gt; [EA]128. (Ext: 1)</td>
</tr>
<tr>
<td>QINQ</td>
<td>140572</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Convert quad to integer.</td>
</tr>
<tr>
<td>QIQR</td>
<td>140573</td>
<td>QAD</td>
<td>3</td>
<td>5</td>
<td>VI</td>
<td>Convert quad to integer rounded.</td>
</tr>
</tbody>
</table>
### 7.3.18. Register AP Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STEX</td>
<td>060027</td>
<td>RAP</td>
<td>6</td>
<td>5</td>
<td>I</td>
<td></td>
</tr>
</tbody>
</table>

### 7.3.19. Register Generic Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADLR</td>
<td>060014</td>
<td>RGN</td>
<td>-</td>
<td>7</td>
<td>I</td>
<td>Add LINK to R.</td>
</tr>
<tr>
<td>CGT</td>
<td>060026</td>
<td>RGN</td>
<td>6</td>
<td>5</td>
<td>I</td>
<td>Computed go to.</td>
</tr>
<tr>
<td>CHS</td>
<td>060040</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Change sign of R. ^R(1) =&gt; R(1).</td>
</tr>
<tr>
<td>CMH</td>
<td>060045</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Complement RH. ^R = RH.</td>
</tr>
<tr>
<td>CMR</td>
<td>060044</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Complement R. ^R =&gt; R.</td>
</tr>
<tr>
<td>CR</td>
<td>060059</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Clear R. 0 =&gt; R.</td>
</tr>
<tr>
<td>CRBL</td>
<td>060062</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Clear R left byte. 0 =&gt; R(1-8).</td>
</tr>
<tr>
<td>CRBR</td>
<td>060063</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Clear R right byte. 0 =&gt; R(9-16).</td>
</tr>
<tr>
<td>CRHL</td>
<td>060054</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Clear RH. 0 =&gt; RH.</td>
</tr>
<tr>
<td>CRHR</td>
<td>060055</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Clear R right halfword. 0 =&gt; R(17-32).</td>
</tr>
<tr>
<td>CSR</td>
<td>060041</td>
<td>RGN</td>
<td>5</td>
<td>-</td>
<td>I</td>
<td>Copy &amp; save sign. R(1) =&gt; C, 0 =&gt; R(1).</td>
</tr>
<tr>
<td>DCP</td>
<td>060150</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Decrement character pointer. (32IX)</td>
</tr>
<tr>
<td>DH1</td>
<td>060130</td>
<td>RGN</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Decr RH by 1. RH-1 =&gt; RH'.</td>
</tr>
<tr>
<td>DH2</td>
<td>060131</td>
<td>RGN</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Decr RH by 2. RH-2 =&gt; RH'.</td>
</tr>
<tr>
<td>DR1</td>
<td>060124</td>
<td>RGN</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Decr R by 1. R-1 =&gt; R.</td>
</tr>
<tr>
<td>DR2</td>
<td>060125</td>
<td>RGN</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Decr R by 2. R-2 =&gt; R.</td>
</tr>
<tr>
<td>ICBL</td>
<td>060065</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Exchange bytes. 0 =&gt; RH(1-8) =&gt; RH(9-16).</td>
</tr>
<tr>
<td>ICBR</td>
<td>060066</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Exchange bytes. 0 =&gt; RH(9-16) =&gt; RH(1-8).</td>
</tr>
<tr>
<td>ICHL</td>
<td>060060</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Interchange halfwords. RH =&gt; RL, 0 =&gt; RH.</td>
</tr>
<tr>
<td>ICHR</td>
<td>060061</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Interchange halfwords. RL =&gt; RH, 0 =&gt; RL.</td>
</tr>
<tr>
<td>ICP</td>
<td>060167</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Increment character pointer. (32IX)</td>
</tr>
<tr>
<td>IHI1</td>
<td>060126</td>
<td>RGN</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Incr halfword by 1. RH+1 =&gt; RH.</td>
</tr>
<tr>
<td>IHL2</td>
<td>060127</td>
<td>RGN</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Incr halfword by 2. RH+2 =&gt; RH.</td>
</tr>
<tr>
<td>INK</td>
<td>060070</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Input keys to RH.</td>
</tr>
<tr>
<td>IR1</td>
<td>060122</td>
<td>RGN</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Incr R by 1. R+1 =&gt; R.</td>
</tr>
<tr>
<td>IR2</td>
<td>060123</td>
<td>RGN</td>
<td>2</td>
<td>1</td>
<td>I</td>
<td>Incr R by 2. R+2 =&gt; R.</td>
</tr>
<tr>
<td>IRB</td>
<td>060064</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Interchange bytes. RH(1-8) =&gt; RH(9-16).</td>
</tr>
<tr>
<td>IRH</td>
<td>060057</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Interchange halves. RH =&gt; RL.</td>
</tr>
<tr>
<td>OTH</td>
<td>060071</td>
<td>RGN</td>
<td>7</td>
<td>6</td>
<td>I</td>
<td>Output keys from RH. [RH] =&gt; KEYS.</td>
</tr>
<tr>
<td>PID</td>
<td>060052</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Pos for int divide. R =&gt; R+1; w/ sign extend.</td>
</tr>
<tr>
<td>PIDH</td>
<td>060053</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Pos RH for div. RH =&gt; RL; RH(1) =&gt; RH(2-16).</td>
</tr>
<tr>
<td>PIM</td>
<td>060050</td>
<td>RGN</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Pos after int multiply. (R+1) =&gt; R.</td>
</tr>
<tr>
<td>PIMH</td>
<td>060051</td>
<td>RGN</td>
<td>3</td>
<td>5</td>
<td>I</td>
<td>Pos RH after int multiply. RL =&gt; RH.</td>
</tr>
<tr>
<td>SLLH</td>
<td>060076</td>
<td>RGN</td>
<td>4</td>
<td>-</td>
<td>I</td>
<td>Shift halfword left 1. LS(RH, 1) =&gt; RH.</td>
</tr>
<tr>
<td>SLLZ</td>
<td>060077</td>
<td>RGN</td>
<td>4</td>
<td>-</td>
<td>I</td>
<td>Shift halfword left 2. LS(RH, 2) =&gt; RH.</td>
</tr>
<tr>
<td>SHR1</td>
<td>060120</td>
<td>RGN</td>
<td>4</td>
<td>-</td>
<td>I</td>
<td>Shift halfword right 1. RS(RH, 1) =&gt; RH.</td>
</tr>
<tr>
<td>SHR2</td>
<td>060121</td>
<td>RGN</td>
<td>4</td>
<td>-</td>
<td>I</td>
<td>Shift halfword right 2. RS(RH, 2) =&gt; RH.</td>
</tr>
<tr>
<td>SL1</td>
<td>060072</td>
<td>RGN</td>
<td>4</td>
<td>-</td>
<td>I</td>
<td>Shift halfword left 1. LS(RH, 1) =&gt; R.</td>
</tr>
<tr>
<td>SL2</td>
<td>060073</td>
<td>RGN</td>
<td>4</td>
<td>-</td>
<td>I</td>
<td>Shift halfword left 2. LS(RH, 2) =&gt; R.</td>
</tr>
<tr>
<td>SR1</td>
<td>060074</td>
<td>RGN</td>
<td>4</td>
<td>-</td>
<td>I</td>
<td>Shift halfword right 1. RS(RH, 1) =&gt; R.</td>
</tr>
<tr>
<td>SR2</td>
<td>060075</td>
<td>RGN</td>
<td>4</td>
<td>-</td>
<td>I</td>
<td>Shift halfword right 2. RS(RH, 2) =&gt; R.</td>
</tr>
<tr>
<td>SSM</td>
<td>060042</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Set sign minus. 1 =&gt; R(1).</td>
</tr>
<tr>
<td>SSP</td>
<td>060043</td>
<td>RGN</td>
<td>-</td>
<td>-</td>
<td>I</td>
<td>Set sign plus. 0 =&gt; R(1).</td>
</tr>
<tr>
<td>Mnem</td>
<td>OpCode</td>
<td>Typ</td>
<td>C</td>
<td>cc</td>
<td>Modes</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>-----</td>
<td>---</td>
<td>----</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>TC</td>
<td>060046</td>
<td>RGN</td>
<td>3</td>
<td>1</td>
<td>I</td>
<td>Two's complement R. -R =&gt; R.</td>
</tr>
<tr>
<td>TCH</td>
<td>060047</td>
<td>RGN</td>
<td>3</td>
<td>1</td>
<td>I</td>
<td>Two's complement RH. -RH =&gt; RH.</td>
</tr>
</tbody>
</table>

### 7.3.20. Shift Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>0414--</td>
<td>SH</td>
<td>4</td>
<td>-</td>
<td>SRV</td>
<td>A left logical.</td>
</tr>
<tr>
<td>ALR</td>
<td>0416--</td>
<td>SH</td>
<td>4</td>
<td>-</td>
<td>SRV</td>
<td>A left rotate.</td>
</tr>
<tr>
<td>ALS</td>
<td>0415--</td>
<td>SH</td>
<td>3</td>
<td>-</td>
<td>SRV</td>
<td>A left shift (arith).</td>
</tr>
<tr>
<td>ARL</td>
<td>0404--</td>
<td>SH</td>
<td>4</td>
<td>-</td>
<td>SRV</td>
<td>A right logical.</td>
</tr>
<tr>
<td>ARR</td>
<td>0406--</td>
<td>SH</td>
<td>4</td>
<td>-</td>
<td>SRV</td>
<td>A right rotate.</td>
</tr>
<tr>
<td>ARS</td>
<td>0405--</td>
<td>SH</td>
<td>4</td>
<td>-</td>
<td>SRV</td>
<td>A right shift (arith).</td>
</tr>
<tr>
<td>LGL</td>
<td>0414--</td>
<td>SH</td>
<td>4</td>
<td>5</td>
<td>SRV</td>
<td>OBSOLETE. A left logical. (Use ALL)</td>
</tr>
<tr>
<td>LGR</td>
<td>0404--</td>
<td>SH</td>
<td>4</td>
<td>5</td>
<td>SRV</td>
<td>OBSOLETE. A right logical. (Use ARL)</td>
</tr>
<tr>
<td>LLL</td>
<td>0410--</td>
<td>SH</td>
<td>4</td>
<td>-</td>
<td>SRV</td>
<td>Long left logical.</td>
</tr>
<tr>
<td>LLR</td>
<td>0412--</td>
<td>SH</td>
<td>4</td>
<td>5</td>
<td>SRV</td>
<td>Long left rotate.</td>
</tr>
<tr>
<td>LLS</td>
<td>0411--</td>
<td>SH</td>
<td>3</td>
<td>5</td>
<td>SRV</td>
<td>Long left shift. (SR -&gt; B(1) ignored)</td>
</tr>
<tr>
<td>LRL</td>
<td>0400--</td>
<td>SH</td>
<td>4</td>
<td>-</td>
<td>SRV</td>
<td>Long right logical.</td>
</tr>
<tr>
<td>LRR</td>
<td>0402--</td>
<td>SH</td>
<td>4</td>
<td>-</td>
<td>SRV</td>
<td>Long right rotate.</td>
</tr>
<tr>
<td>LRS</td>
<td>0401--</td>
<td>SH</td>
<td>4</td>
<td>-</td>
<td>SRV</td>
<td>Long right shift. (SR -&gt; B(1) ignored)</td>
</tr>
</tbody>
</table>

### 7.3.21. Skip Operations

<table>
<thead>
<tr>
<th>Mnem</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOP</td>
<td>101000</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>No operation (faster on certain machines).</td>
</tr>
<tr>
<td>SAR</td>
<td>10026-</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A(n) reset.</td>
</tr>
<tr>
<td>SAS</td>
<td>10126-</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A(n) set.</td>
</tr>
<tr>
<td>SEQ</td>
<td>100040</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Skip if A .EQ. 0. (Use SZE)</td>
</tr>
<tr>
<td>SGE</td>
<td>100400</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Skip if A .GE. 0. (Use SPL)</td>
</tr>
<tr>
<td>SGT</td>
<td>100220</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A .GT. 0.</td>
</tr>
<tr>
<td>SKP</td>
<td>100000</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip one word.</td>
</tr>
<tr>
<td>SLE</td>
<td>101220</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A .LE. 0.</td>
</tr>
<tr>
<td>SLN</td>
<td>101100</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A bit 16 set.</td>
</tr>
<tr>
<td>SLT</td>
<td>101400</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Skip if A .LT. 0. (Use SMI)</td>
</tr>
<tr>
<td>SLZ</td>
<td>100100</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A bit 16 .EQ. 0.</td>
</tr>
<tr>
<td>SMCR</td>
<td>100220</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if machine check reset.</td>
</tr>
<tr>
<td>SMCS</td>
<td>101200</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if machine check set.</td>
</tr>
<tr>
<td>SMI</td>
<td>101400</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A .LT. 0.</td>
</tr>
<tr>
<td>SNE</td>
<td>100140</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV*</td>
<td>OBSOLETE. Skip if A .NE. 0. (Use SNZ)</td>
</tr>
<tr>
<td>SNR</td>
<td>10024-</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV*</td>
<td>OBSOLETE. Skip if sense switch N reset.</td>
</tr>
<tr>
<td>SNS</td>
<td>10124-</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV*</td>
<td>OBSOLETE. Skip if sense switch N set.</td>
</tr>
<tr>
<td>SNZ</td>
<td>101040</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A .NE. 0.</td>
</tr>
<tr>
<td>SPL</td>
<td>100400</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>Skip if A .GE. 0.</td>
</tr>
<tr>
<td>SPN</td>
<td>100200</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Skip if machine check reset. (Use SMCR)</td>
</tr>
<tr>
<td>SPS</td>
<td>101200</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV</td>
<td>OBSOLETE. Skip if machine check set. (Use SMCS)</td>
</tr>
<tr>
<td>SR1</td>
<td>100020</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV*</td>
<td>OBSOLETE. Skip if sense switch 1 reset.</td>
</tr>
<tr>
<td>SR2</td>
<td>100010</td>
<td>SKP</td>
<td>-</td>
<td>-</td>
<td>SRV*</td>
<td>OBSOLETE. Skip if sense switch 2 reset.</td>
</tr>
<tr>
<td>Mнем</td>
<td>OpCode</td>
<td>Typ</td>
<td>C</td>
<td>cc</td>
<td>Modes</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>-----</td>
<td>---</td>
<td>----</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>SR3</td>
<td>100004</td>
<td>SKP</td>
<td>-</td>
<td></td>
<td>SRV*</td>
<td>OBSOLETE. Skip if sense switch 3 reset.</td>
</tr>
<tr>
<td>SR4</td>
<td>100002</td>
<td>SKP</td>
<td>-</td>
<td></td>
<td>SRV*</td>
<td>OBSOLETE. Skip if sense switch 4 reset.</td>
</tr>
<tr>
<td>SRC</td>
<td>100001</td>
<td>SKP</td>
<td>-</td>
<td></td>
<td>SRV</td>
<td>Skip if CBIT reset.</td>
</tr>
<tr>
<td>SS1</td>
<td>101020</td>
<td>SKP</td>
<td>-</td>
<td></td>
<td>SRV**</td>
<td>OBSOLETE. Skip if sense switch 1 set.</td>
</tr>
<tr>
<td>SS2</td>
<td>101010</td>
<td>SKP</td>
<td>-</td>
<td></td>
<td>SRV**</td>
<td>OBSOLETE. Skip if sense switch 2 set.</td>
</tr>
<tr>
<td>SS3</td>
<td>101004</td>
<td>SKP</td>
<td>-</td>
<td></td>
<td>SRV*</td>
<td>OBSOLETE. Skip if sense switch 3 set.</td>
</tr>
<tr>
<td>SS4</td>
<td>101002</td>
<td>SKP</td>
<td>-</td>
<td></td>
<td>SRV*</td>
<td>OBSOLETE. Skip if sense switch 4 set.</td>
</tr>
<tr>
<td>SSC</td>
<td>101001</td>
<td>SKP</td>
<td>-</td>
<td></td>
<td>SRV</td>
<td>Skip if CBIT set.</td>
</tr>
<tr>
<td>SSR</td>
<td>100036</td>
<td>SKP</td>
<td>-</td>
<td></td>
<td>SRV*</td>
<td>OBSOLETE. Skip if all sense switches reset.</td>
</tr>
<tr>
<td>SSS</td>
<td>101036</td>
<td>SKP</td>
<td>-</td>
<td></td>
<td>SRV**</td>
<td>OBSOLETE. Skip if all sense switches set.</td>
</tr>
<tr>
<td>SZE</td>
<td>100040</td>
<td>SKP</td>
<td>-</td>
<td></td>
<td>SRV</td>
<td>Skip if A.EQ. 0.</td>
</tr>
</tbody>
</table>

### 7.3.22. P300 Virtual Memory Operations

<table>
<thead>
<tr>
<th>Mнем</th>
<th>OpCode</th>
<th>Typ</th>
<th>C</th>
<th>cc</th>
<th>Modes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPMJ</td>
<td>000217</td>
<td>VM</td>
<td>-</td>
<td></td>
<td>SR</td>
<td>OBSOLETE. Enter page mode &amp; jump to microcode (P300).</td>
</tr>
<tr>
<td>EPMX</td>
<td>000237</td>
<td>VM</td>
<td>-</td>
<td></td>
<td>SR</td>
<td>OBSOLETE. Enter page mode &amp; jump to microcode (P300).</td>
</tr>
<tr>
<td>ERMJ</td>
<td>000701</td>
<td>VM</td>
<td>-</td>
<td></td>
<td>SR</td>
<td>OBSOLETE. Enter restricted mode &amp; jump (P300).</td>
</tr>
<tr>
<td>ERMex</td>
<td>000721</td>
<td>VM</td>
<td>-</td>
<td></td>
<td>SR</td>
<td>OBS. Enter restr'd mode &amp; jump to microcode (P300).</td>
</tr>
<tr>
<td>EVMJ</td>
<td>000703</td>
<td>VM</td>
<td>-</td>
<td></td>
<td>SR</td>
<td>OBSOLETE. Enter virtual mode &amp; jump to microcode (P300).</td>
</tr>
<tr>
<td>EVMX</td>
<td>000723</td>
<td>VM</td>
<td>-</td>
<td></td>
<td>SR</td>
<td>OBS. Enter virtual mode &amp; jump to microcode (P300).</td>
</tr>
<tr>
<td>LPMJ</td>
<td>000215</td>
<td>VM</td>
<td>-</td>
<td></td>
<td>SR</td>
<td>OBSOLETE. Leave page mode &amp; jump to microcode (P300).</td>
</tr>
<tr>
<td>LPMX</td>
<td>000235</td>
<td>VM</td>
<td>-</td>
<td></td>
<td>SR</td>
<td>OBS. Leave page mode &amp; jump to microcode (P300).</td>
</tr>
</tbody>
</table>
8. OPERATIONS

8.1. Front Panel Controls

<table>
<thead>
<tr>
<th>Switch</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>POWER</td>
<td>turns power on/off</td>
</tr>
<tr>
<td>KEY LOCK</td>
<td>locks/unlocks next 3 switches</td>
</tr>
<tr>
<td>MASTER CLEAR</td>
<td>initialize system</td>
</tr>
<tr>
<td>REMOTE ENABLE</td>
<td>permits remote access</td>
</tr>
<tr>
<td>REMOTE PRIVILEGE</td>
<td>selects remote privilege level</td>
</tr>
<tr>
<td>MULTI STREAM</td>
<td>select multiple stream mode (both ISUs)(P850)</td>
</tr>
<tr>
<td>ISU 1</td>
<td>select Instruction Stream Unit 1(P850)</td>
</tr>
<tr>
<td>ISU 2</td>
<td>select Instruction Stream Unit 2(P850)</td>
</tr>
</tbody>
</table>

8.2. Standard VCP Procedures

8.2.1. Cold start

1. Turn on power to equipment: supervisor terminal, CPU, disk drives, other peripherals.

2. For all machines (from first partition on drive 0, first controller):

```
SYSCLR
```

For Primos:

```
BOOT 14114
```

or, for Primos II:

```
BOOT 10114
```

3. For other than first partition:

```
SYSCLR
```

For Primos:

```
BOOT 4114
PHYSICAL_DEVICE=physical_device_number
```

or, for Primos II:

```
BOOT 114
PHYSICAL_DEVICE=physical_device_number
```

4. Add '20 to the device number for first partition on second controller ('27). See section 8.3 for other boot switch settings.

5. To bring up Primos from Primos II:

```
PRIMOS [directory_containing_Primos]
```

Option need not be specified if booting from same directory as last time.
For other boot options or devices see the Boot Device Table, section 8.3.

8.2.2. Warm Start
If a warm start is desired to reset a controller while the CPU is still running, hit the ESCAPE key twice to access the VCP/CP and then type STOP.

1. For 50 series machines, type in:
   
   \texttt{SYSCLR}
   
   RUN
   
   2. For 9000, 4000, 6000 and 2000 machines (CP), type in:
   
   \texttt{WARMstart}

8.2.3. Tape Dump
For 9000s, 4000s, and 6000s type:

\texttt{TAPEdump unit}

For 50 series, type:

\begin{tabular}{|c|c|c|c|}
  \hline
  \textbf{Drive} & \textbf{Drive} & \textbf{Drive} & \textbf{Drive} \\
  0 & 1 & 2 & 3 \\
  \texttt{SYSCLR} & \texttt{SYSCLR} & \texttt{SYSCLR} & \texttt{SYSCLR} \\
  \texttt{RUN 775} & \texttt{RUN 776} & \texttt{A 7} & \texttt{A 7} \\
  & & 775 & 775 \\
  & & / & / \\
  & & SS 2 & SS 3 \\
  & & RUN & RUN \\
  \hline
\end{tabular}

8.3. Boot Device Settings

8.3.1. Booting from SMDs

\begin{tabular}{|c|c|c|c|c|c|}
  \hline
  H & D & E & F & P & \textbf{unit} & \textbf{C} & \textbf{1} & \textbf{1} & \textbf{0} & \textbf{0} \textbf{1} & \textbf{1} & \textbf{0} \\
  0 & & & & & - & B & T & - & 101 & & & & \textbf{Storage Module} & \textbf{Magtape} \\
  \hline
\end{tabular}

\begin{tabular}{|c|c|c|}
  \hline
  Field & Description & Octal & Hex \\
  \hline
  H & \begin{itemize}
  \item Bypass CONFIG file; prompt for COMDEV & PAGING
  \end{itemize} & 100000 & 8000 \\
  D & \begin{itemize}
  \item Enable the ring 0 debugger
  \end{itemize} & 040000 & 4000 \\
  E & \begin{itemize}
  \item Enter the debugger during coldstart
  \end{itemize} & 020000 & 2000 \\
  F & \begin{itemize}
  \item Boot from first partition on drive 0, controller 0
  \end{itemize} & 010000 & 1000 \\
  P & \begin{itemize}
  \item Continue boot to PRIMOS
  \end{itemize} & 004000 & 0800 \\
  \textbf{unit} & \begin{itemize}
  \item Drive unit number
  \end{itemize} & 000600 & 0180 \\
  C & \begin{itemize}
  \item Controller number
  \end{itemize} & 000060 & 0030 \\
  \hline
\end{tabular}
Field | Description | Octal | Hex |
--- | --- | --- | --- |
Rel | Relocate boot file to ending address of:  
00 - end of physical memory  
01 - 16K  
10 - 32K  
11 - 48K | 000600 | 0180 |
A | Suppress auto-start of program | 000100 | 0040 |
B | Halt to allow baud rate change | 000040 | 0020 |
T | Drive type:  
0 - 9-track  
1 - 7-track | 000020 | 0010 |

8.4. Formatting disks: MAKE
To make a new disk from scratch (never on a Prime), use:

```
MAKE -PART partition-name -DISK pdev -DT device-type  
-FMT -NO_INIT -NEW_DISK
```

To remake an existing pack, use:

```
MAKE -PART partition-name -DISK pdev -DT drive-type  
-NO_INIT
```

Common device types are:

- SMD 80 Mb or 300 Mb removable packs.
- MODEL_4475 315 Mb fixed media (dark brown front, Century Data).
- MODEL_4735 500 Mb fixed media (pikeral).
- MODEL_4845 770 Mb fixed media (beluga).

Commonly needed options:

- -SPLIT [number-of-paging-records]  
  Split the disk into paging and file system parts. If not supplied, MAKE will ask for the number of paging records.
- -IC Make the disk for an intelligent controller (ICOP mode controller). Uses dynamic badspotting.
- -AC Make the disk compatible with all controllers. Can not use mirroring.

For further info, see MAKE in the commands chapter (2.7).

8.5. Disk maintenance: FIX_DISK
To check a disk for damage but do no correcting:

```
FIX_DISK -DISK pdev
```

To quickly fix quotas or robust partitions (fast mode), use:

```
FIX_DISK -DISK pdev -FIX -FAST
```

Otherwise, do a normal disk repair:
8.6. Adding & changing user configurations: EDIT_PROFILE
To invoke EDIT_PROFILE, enter:

EDIT_PROFILE

Then, to add a user, enter the underlined commands at the appropriate prompts:

> AU username -PW initial-password
 Groups: system-wide-groups
 Default login project: default-project
 Password lifetime in days: number-of-days
 > AU username -PROJ default-project -PROF
 Groups: project-related-groups
 Initial attach point: <partition>directory-path
 Create/change user attributes? Y
 Number of command levels: number-of-command-levels
 Number of live program invocations per command level: number-of-invocations
 Number of private, dynamic segments: number-of-segments
 Number of private, static segments: number-of-segments
 > Q

then attach to the partition given and create the user's directory (this may be a subdirectory
within another directory):

   A <partition>MFD
   CR directory-path
   SAC directory-path user ALL $REST LUR

The ACLs may be changed, the above is a typical setting.

Users may be changed by using the CU command in EDIT_PROFILE and deleted using the DU
sub-command. For more information on EDIT_PROFILE see its entry in the commands chapter
(2.7).

8.7. VCP Commands
Access [address | register][modes]
Subcommands:

   return
   Access next location.

   ^
   Access previous location.

   number
   Replace this location with number.

   / Return to VCP.

AWARMOFF

Don't warmstart on power return. (UPS, 9000 series)
AWARMON
Warmstart on power restore. (UPS, 9000 series)

BOOT device-number
Boot the CPU.

BOOTD
Boot CPU to PRIMOS II. (9000 series)

BOOTP
Boot CPU to PRIMOS. (9000 series)

Copy start end to
Copy memory block. Copies area between start and end to area starting at to.

DATE
Display the date. (9000 series)

DIRectory [:0 | :1]
Display VCP floppy disk directory contents. (9000 series) Default is last drive displayed.

DISPLAY address
Display virtual memory contents. (Only when PRIMOS is running.)

DISPLAYC address
Continuously display virtual memory contents. (Only when PRIMOS is running.)

DOS
Restart PRIMOS II after interruption. (9000 series)

Dump {register | start end}[modes]
Display the register or block of memory according to modes (see A).

FETCH
Display data according to previously set sense and data switches.

Fill start end number
Fill block of memory from start to end with number.

HALT
Stop the CPU. (9000 series)

HELP
Display list of DP commands. (9000 series)

HISTORY

Invoke history disk editor. (6000 series)
Subcommands ("HST>" prompt):
 P n  Print next n entries.
 N n  Move n entries from current and print it.
 ^   Print previous entry.
 E   Go to last entry.
 W   Write a comment (max 256 chars). Terminate with '\$'.
 F   Format the history disk floppy. Erases all data.
 Q   Exit to CP mode.

LDNET [filename]
Load a decode net file. (9000 and 6000 series).

Lights
Display the current value of the lights register. Abbreviation may only be used on 9000 series.

LightsC
Display the lights register whenever it changes. Abbreviation may only be used on 9000 series.

listREV
List CPU type, part number and required rev for each CPU board. (9000 series)

MO ABS
Enter absolute addressing mode.

MO BRIEF
Enter limited diagnostic message mode. (9000 series)

MO FULL
Enter full diagnostic message mode. (9000 series)

MO MAP
Enter mapped addressing mode. (Default condition.)

MO RFABS
Enter absolute register set addressing mode.
MO RFCRS
Enter current register set addressing mode.

MO RFH
Specify that high-order half of register is to be modified.

MO RFL
Specify that low-order half of register is to be modified.

MO ST
Place the terminal in supervisor terminal mode.

MO USER
Place the terminal in user terminal mode. (2250, 9000 series)

RCP [address]
Run without entering supervisor terminal mode.

REMPWD
Set password on remote port. (9000 series)

RUN [address]
Start the CPU running.

SD number
Set data switches. Except on 9000 series, this number is destroyed by any successive command that uses a number.

SEtime -mmddyy -hhmmww [-D]
Set the date and time for the DP. (9000 series) mmddyy is month(01-12), day(01-31), year(00-99). hhmmww is hour(00-23), minute(00-59), and day of week(1-7, 1 = Sunday). -D enables automatic daylight savings time change(last Sunday of April to last Sunday of October).

SPINDOWN
Instruct 68MB and 158MB (Winchester) drives to spin down. (2250) Must be issued before powering down 2250.

SS
Set sense switches.

STORE
Store specified data according to previously set sense and data switches.
Sysclr

Perform limited master clear. Resets CPU and I/O controllers. (Abbreviation valid only on 9000 series.)

SYSOUT {BUFF | IGN | INT}

Controls output to supervisor terminal. Output is either buffered (BUFF), ignored (IGN) or interleaved with interactive mode (INT). (2000 and 9000 series only.)

TAPEdump unit

Causes CPU to dump the current memory image on to the tape on drive unit. 9000 series only.

TRACE [number]

Single steps CPU for number of instructions. (2000 and 9000 series only.)

VIRY

Perform complete system master clear. Resets VCP, CPU and I/O controllers. Verifies VCP and CPU.

VPSD

Enter wired VPSD. VPSD directive must have been in CONFIG. Primos must have been running and machine must be halted. (2000 and 9000 only.) Old 50 series equivalent:

SYSCLR
RUN 600

WARMstart

Attempt warmstart of Primos. (9000 series.) Other machine equivalent:

SYSCLR
RUN
RUN
# 9. Peripheral I/O

## 9.1. Addresses

<table>
<thead>
<tr>
<th>Addr</th>
<th>Device</th>
<th>Addr</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>Polling</td>
<td>40</td>
<td>PRIMAD (AIS)</td>
</tr>
<tr>
<td>01</td>
<td>Paper Tape Reader</td>
<td>41</td>
<td>Digital Input 1 (DIS)</td>
</tr>
<tr>
<td>02</td>
<td>Paper Tape Punch</td>
<td>42</td>
<td>Digital Input 2</td>
</tr>
<tr>
<td>03</td>
<td>Unit Record Controller 1</td>
<td>43</td>
<td>Digital Output 1 (DOS)</td>
</tr>
<tr>
<td>04</td>
<td>STTY</td>
<td>44</td>
<td>Digital Output 2</td>
</tr>
<tr>
<td>05</td>
<td>Unit Record Controller 2</td>
<td>45</td>
<td>Disk Ctrlr (was AQS)</td>
</tr>
<tr>
<td>06</td>
<td>Interproc. Channel (IPC)</td>
<td>46</td>
<td>Disk Ctrlr (was CPI)</td>
</tr>
<tr>
<td>07</td>
<td>Primenet Node Controller 1</td>
<td>47</td>
<td>PNC 2</td>
</tr>
<tr>
<td>10</td>
<td>ICS2 1 or ICS1</td>
<td>50</td>
<td>HSSMLC 1</td>
</tr>
<tr>
<td>11</td>
<td>ICS2 2 or ICS1</td>
<td>51</td>
<td>HSSMLC 2 or MDLC</td>
</tr>
<tr>
<td>12</td>
<td>Floppy disk</td>
<td>52</td>
<td>AMLC 3 or ICS1</td>
</tr>
<tr>
<td>13</td>
<td>Magtape Controller 2</td>
<td>53</td>
<td>AMLC 2</td>
</tr>
<tr>
<td>14</td>
<td>Magtape Controller 1</td>
<td>54</td>
<td>AMLC 1</td>
</tr>
<tr>
<td>15</td>
<td>AMLC 5 or ICS1</td>
<td>55</td>
<td>MACI Autocall</td>
</tr>
<tr>
<td>16</td>
<td>AMLC 6 or ICS1</td>
<td>56</td>
<td>SMLC</td>
</tr>
<tr>
<td>17</td>
<td>AMLC 7 or ICS1</td>
<td>57</td>
<td>Gen. Purp. IF Board</td>
</tr>
<tr>
<td>20</td>
<td>Panel, Real Time Clock</td>
<td>60</td>
<td>GPIB</td>
</tr>
<tr>
<td>21</td>
<td>Disk option B' (4002)</td>
<td>61</td>
<td>GPIB</td>
</tr>
<tr>
<td>22</td>
<td>Disk Controller 3</td>
<td>62</td>
<td>GPIB</td>
</tr>
<tr>
<td>23</td>
<td>Disk Controller 4</td>
<td>63</td>
<td>GPIB</td>
</tr>
<tr>
<td>24</td>
<td>Disk Controller (was WCS)</td>
<td>64</td>
<td>GPIB</td>
</tr>
<tr>
<td>25</td>
<td>Disk Controller (was 4000)</td>
<td>65</td>
<td>GPIB</td>
</tr>
<tr>
<td>26</td>
<td>Disk Controller 1</td>
<td>66</td>
<td>GPIB</td>
</tr>
<tr>
<td>27</td>
<td>Disk Controller 2</td>
<td>67</td>
<td>GPIB</td>
</tr>
<tr>
<td>30</td>
<td>IOC 1 (Parallel I/O)</td>
<td>70</td>
<td>GPIB Test</td>
</tr>
<tr>
<td>31</td>
<td>IOC 2</td>
<td>71</td>
<td>ADAGE GP/400 IF</td>
</tr>
<tr>
<td>32</td>
<td>AMLC 8 or ICS1</td>
<td>72</td>
<td>--</td>
</tr>
<tr>
<td>33</td>
<td>Versatec</td>
<td>73</td>
<td>--</td>
</tr>
<tr>
<td>34</td>
<td>Versatec</td>
<td>74</td>
<td>--</td>
</tr>
<tr>
<td>35</td>
<td>AMLC 4 or ICS1</td>
<td>75</td>
<td>--</td>
</tr>
<tr>
<td>36</td>
<td>ICS1 1</td>
<td>76</td>
<td>--</td>
</tr>
<tr>
<td>37</td>
<td>ICS1 2</td>
<td>77</td>
<td>I/O Bus Test</td>
</tr>
</tbody>
</table>
### 9.2. AMLC

#### 9.2.1. OTA 01 – Set Line Configuration

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>Line Number</td>
<td>170000</td>
<td>F000</td>
</tr>
<tr>
<td>D</td>
<td>Data Set Control</td>
<td>002000</td>
<td>0400</td>
</tr>
<tr>
<td>L</td>
<td>Loop Line</td>
<td>001000</td>
<td>0200</td>
</tr>
<tr>
<td>Speed</td>
<td>Line Speed:</td>
<td>000700</td>
<td>01C0</td>
</tr>
<tr>
<td></td>
<td>0 - 110 BAUD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - 134.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - 300</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 - 1200</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 - Programed Clock</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 - Strap 1 (75)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 - Strap 2 (150)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 - Strap 3 (1800)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>ICS: reverse flow control; AMLC: Unused</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>S</td>
<td>Stop bits: 0 - 1, 1 - 2</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>P</td>
<td>Parity: 1 - Disable Parity</td>
<td>000010</td>
<td>0008</td>
</tr>
<tr>
<td>E</td>
<td>Parity: 0 - Odd Parity, 1 - Even</td>
<td>000004</td>
<td>0004</td>
</tr>
<tr>
<td>Len</td>
<td>Character Length:</td>
<td>000003</td>
<td>0003</td>
</tr>
<tr>
<td></td>
<td>0 - 5 Bits</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - 7 Bits</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - 6 Bits</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 - 8 Bits</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 9.2.2. OTA 02 – Set Line Control

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>Line Number</td>
<td>170000</td>
<td>F000</td>
</tr>
<tr>
<td>I</td>
<td>Interrupt: 1 - Char at a time</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>T</td>
<td>Transmit: 1 - Enable</td>
<td>000010</td>
<td>0008</td>
</tr>
<tr>
<td>E</td>
<td>Echo Back: 1 - Enable</td>
<td>000004</td>
<td>0004</td>
</tr>
<tr>
<td>B</td>
<td>Receive: 1 - Off, Report Break</td>
<td>000002</td>
<td>0002</td>
</tr>
<tr>
<td>R</td>
<td>Receive: 1 - Enable</td>
<td>000001</td>
<td>0001</td>
</tr>
</tbody>
</table>
### 9.3. ASR

<table>
<thead>
<tr>
<th>BAUD</th>
<th>OPTION-A</th>
<th>CTL 1</th>
<th>CTL 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>110</td>
<td>110</td>
<td>27</td>
<td>740**</td>
</tr>
<tr>
<td>300</td>
<td>1010</td>
<td>76</td>
<td>340**</td>
</tr>
<tr>
<td>1200</td>
<td>2010</td>
<td>373</td>
<td>340**</td>
</tr>
<tr>
<td>9600</td>
<td>3410</td>
<td>3735</td>
<td>340**</td>
</tr>
</tbody>
</table>

** = number of delays used by BOOT, PRIMOS

### 9.4. DISK CONTROLLERS

#### 9.4.1. Disk Channel Program Definitions

<table>
<thead>
<tr>
<th>Op Code</th>
<th>Mask</th>
<th>Various fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mnem</td>
<td>Op Code</td>
<td>Ex time (µsec)</td>
</tr>
<tr>
<td>DHLT</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>SFORM</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>SSEEK</td>
<td>3</td>
<td>7.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSEL</td>
<td>4</td>
<td>7.5</td>
</tr>
<tr>
<td>SREAD</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SWRITE</td>
<td>6</td>
<td>-</td>
</tr>
<tr>
<td>DSTALL</td>
<td>7</td>
<td>210</td>
</tr>
<tr>
<td>DSTAT</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>SSTOR</td>
<td>A</td>
<td>9</td>
</tr>
<tr>
<td>DOAR</td>
<td>B</td>
<td>9</td>
</tr>
<tr>
<td>SLOAD</td>
<td>C</td>
<td>9</td>
</tr>
<tr>
<td>SDMA</td>
<td>D</td>
<td>6</td>
</tr>
<tr>
<td>DINT</td>
<td>E</td>
<td>6+CPU</td>
</tr>
<tr>
<td>DTRAN</td>
<td>F</td>
<td>6</td>
</tr>
</tbody>
</table>
9.5. Disk Device Numbers (PDEV)

Rev 21.0:

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>If 0, do not execute inst if:</td>
<td>00400</td>
<td>0400</td>
</tr>
<tr>
<td></td>
<td>If 1, execute inst if:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>No function. Reserved for &quot;selected diskfile is write protected.&quot;</td>
<td>00200</td>
<td>0200</td>
</tr>
<tr>
<td>7</td>
<td>Last read or write record inst caused a DMA overrun, check error, controller parity error or header check failure (status word bits 2,4,5, or 6 set).</td>
<td>00100</td>
<td>0100</td>
</tr>
<tr>
<td>8</td>
<td>Selected MHD is seeking.</td>
<td>00040</td>
<td>0080</td>
</tr>
<tr>
<td>9</td>
<td>Selected diskfile has an error condition (status word bits 14, 15, or 16 are set).</td>
<td>00020</td>
<td>0040</td>
</tr>
<tr>
<td>10</td>
<td>For dual port operation only. Selected diskfile is busy servicing the &quot;other&quot; controller.</td>
<td>00010</td>
<td>0020</td>
</tr>
</tbody>
</table>

Field | Description | Octal | Hex   |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>(Offset to First Head)/2</td>
<td>170000</td>
<td>F000</td>
</tr>
<tr>
<td>NHeads</td>
<td>(Number of Heads)/2</td>
<td>007400</td>
<td>0F00</td>
</tr>
<tr>
<td>Ctr</td>
<td>Controller:</td>
<td>000340</td>
<td>00E0</td>
</tr>
<tr>
<td>0 - ('24)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 - ('26)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 - ('25)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 - ('22)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 - ('45)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 - ('27)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 - ('46)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7 - ('23)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td>Reserved. Must be 1.</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>Unit</td>
<td>Unit (Inc. bit 16 for Diskette)</td>
<td>000016</td>
<td>000E</td>
</tr>
<tr>
<td>N</td>
<td>LSB of Number Heads</td>
<td>000001</td>
<td>0001</td>
</tr>
</tbody>
</table>

Pre-rev 21.0:

<table>
<thead>
<tr>
<th>1</th>
<th>4</th>
<th>5</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>NHeads</td>
<td>Ctr</td>
<td>Type</td>
<td>Unit</td>
<td>N</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9-4 Prime Restricted
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>First</td>
<td>(Offset to First Head)/2</td>
<td>170000</td>
<td>F000</td>
</tr>
<tr>
<td>NHeads</td>
<td>(Number of Heads)/2</td>
<td>007400</td>
<td>0F00</td>
</tr>
<tr>
<td>Ctr</td>
<td>Controller:</td>
<td>000300</td>
<td>00C0</td>
</tr>
<tr>
<td></td>
<td>0 - 1 ('26)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - 3 ('22)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - 2 ('27)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 - 4 ('23)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Type of Controller:</td>
<td>000070</td>
<td>0038</td>
</tr>
<tr>
<td></td>
<td>0 - 4000 MHD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 - 4000 FHD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 - Diskette</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 - 4003 8 Sectors/Track</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 - 4003 FHD</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 - 4003 32 Sectors/Track</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 - 4004 Storage Module</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 - Reserved</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unit</td>
<td>Unit (Inc. bit 16 for Diskette)</td>
<td>000006</td>
<td>0006</td>
</tr>
<tr>
<td>N</td>
<td>Diskette: Low Bit of Unit</td>
<td>000001</td>
<td>0001</td>
</tr>
<tr>
<td></td>
<td>Storage Module: LSB of Number Heads</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 9.6. Disk Errors

#### 9.6.1. Diskette Controller

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>Bad Record Identifier</td>
<td>177777</td>
<td>FFFF</td>
</tr>
<tr>
<td>-</td>
<td>Device Not Ready</td>
<td>177776</td>
<td>FFFE</td>
</tr>
<tr>
<td>1</td>
<td>Normal End of Instruction</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>2</td>
<td>Sector Not Found</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>3</td>
<td>Checksum Error on Sector ID</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>Track Error (head misposition)</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>5</td>
<td>Bad OTA or Not Ready</td>
<td>004000</td>
<td>0800</td>
</tr>
<tr>
<td>6</td>
<td>Deleted Data Mark Read</td>
<td>002000</td>
<td>0400</td>
</tr>
<tr>
<td>7</td>
<td>DMx Overrun</td>
<td>001000</td>
<td>0200</td>
</tr>
<tr>
<td>8</td>
<td>Chksum err. Write Prot. Violation, Inoperable on</td>
<td>000400</td>
<td>0100</td>
</tr>
<tr>
<td></td>
<td>Write or Format</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9-15</td>
<td>Unused</td>
<td>000376</td>
<td>00FE</td>
</tr>
<tr>
<td>16</td>
<td>Not Ready</td>
<td>000001</td>
<td>0001</td>
</tr>
<tr>
<td>-</td>
<td>Redundant Int. (Warm Start)</td>
<td>000000</td>
<td>0000</td>
</tr>
</tbody>
</table>
### 9.6.2. Storage Module (4004 Controller)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>Bad Record Identifier</td>
<td>177777</td>
<td>FFFF</td>
</tr>
<tr>
<td>-</td>
<td>Device Not Ready (DOS)</td>
<td>177776</td>
<td>FFFE</td>
</tr>
<tr>
<td>-</td>
<td>Memory Parity Error During DMx</td>
<td>177775</td>
<td>FFD</td>
</tr>
<tr>
<td>-</td>
<td>No controller</td>
<td>177774</td>
<td>FFFC</td>
</tr>
<tr>
<td>-</td>
<td>Hung controller</td>
<td>177773</td>
<td>FFFB</td>
</tr>
<tr>
<td>1</td>
<td>Bit 1 Always On</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>2</td>
<td>DMA Overrun</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>3</td>
<td>Write Protect</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>Read Check</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>5</td>
<td>Data Parity Error</td>
<td>004000</td>
<td>0800</td>
</tr>
<tr>
<td>6</td>
<td>Header Check</td>
<td>002000</td>
<td>0400</td>
</tr>
<tr>
<td>7-10</td>
<td>Unused</td>
<td>001700</td>
<td>0360</td>
</tr>
<tr>
<td>11</td>
<td>Busy(Dual Port Only)</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>12</td>
<td>Unused</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>13</td>
<td>Seeking</td>
<td>000010</td>
<td>0008</td>
</tr>
<tr>
<td>14</td>
<td>Illegal Seek</td>
<td>000004</td>
<td>0004</td>
</tr>
<tr>
<td>15</td>
<td>Select Error</td>
<td>000002</td>
<td>0002</td>
</tr>
<tr>
<td>16</td>
<td>Not Ready (hardware)</td>
<td>000001</td>
<td>0001</td>
</tr>
<tr>
<td>-</td>
<td>Redundant Int. (Warm Start)</td>
<td>000000</td>
<td>0000</td>
</tr>
</tbody>
</table>

### 9.7. DMx control words

#### 9.7.1. DMA

<table>
<thead>
<tr>
<th>0</th>
<th>12 13 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>Word Count Segment</td>
</tr>
<tr>
<td>1</td>
<td>Starting Address</td>
</tr>
</tbody>
</table>

OTA '14dd:

<table>
<thead>
<tr>
<th>1</th>
<th>5 6 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nchan</td>
<td>Chan Address</td>
</tr>
</tbody>
</table>

Nchan = Number of channels - 1.
9.7.2. DMC

<table>
<thead>
<tr>
<th>Address</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Start Address</td>
</tr>
<tr>
<td>1</td>
<td>End Address</td>
</tr>
</tbody>
</table>

OTA '14dd:

<table>
<thead>
<tr>
<th>Address</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 4 5 6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Nchan 1</td>
</tr>
<tr>
<td></td>
<td>Chan Address</td>
</tr>
</tbody>
</table>

Nchan = Number of channels - 1.

9.7.3. DMQ

<table>
<thead>
<tr>
<th>Address</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Read Pointer</td>
</tr>
<tr>
<td>1</td>
<td>Write Pointer</td>
</tr>
<tr>
<td>2 6</td>
<td>- HO bits phy addr</td>
</tr>
<tr>
<td>3</td>
<td>Mask</td>
</tr>
</tbody>
</table>

Mask = length (of Queue) - 1.
length = \(2^K\) (4 \(\leq K \leq 10\))
(Queue must be on \(2^K\) boundary.)

INPUT: End of Range if no room.
OUTPUT: EOR if empty (not w/last entry).

9.7.4. DMT
Device Defined.
## 9.8. Magtape

### 9.8.1. Command Bit Definitions

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select Transport (bits 9-12)</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>2</td>
<td>0=&gt;File Operation, 1=&gt;Record Op</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>3</td>
<td>0=&gt;Read/Write Op, 1=&gt;Spacing Op</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>1=&gt;9-Track Read and Correct</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>5</td>
<td>0=&gt;Binary, 1=&gt;BCD (7-track only)</td>
<td>004000</td>
<td>0800</td>
</tr>
<tr>
<td>6</td>
<td>0=&gt;7-Track Transport, 1=&gt;9-Track</td>
<td>002000</td>
<td>0400</td>
</tr>
<tr>
<td>7</td>
<td>Unused</td>
<td>001000</td>
<td>0200</td>
</tr>
<tr>
<td>8</td>
<td>1=&gt;2 Characters per Word</td>
<td>000400</td>
<td>0100</td>
</tr>
<tr>
<td>9</td>
<td>1=&gt;Forward Motion (bits 10,11=0)</td>
<td>000200</td>
<td>0080</td>
</tr>
<tr>
<td>10</td>
<td>1=&gt;Reverse Motion (bits 9,11,12=0)</td>
<td>000100</td>
<td>0040</td>
</tr>
<tr>
<td>11</td>
<td>1=&gt;Rewind (bits 9,10,12=0)</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>12</td>
<td>1=&gt;Write Order</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>13</td>
<td>Select Transport 0</td>
<td>000010</td>
<td>0008</td>
</tr>
<tr>
<td>14</td>
<td>Select Transport 1</td>
<td>000004</td>
<td>0004</td>
</tr>
<tr>
<td>15</td>
<td>Select Transport 2</td>
<td>000002</td>
<td>0002</td>
</tr>
<tr>
<td>16</td>
<td>Select Transport 3</td>
<td>000001</td>
<td>0001</td>
</tr>
</tbody>
</table>

### 9.8.2. Magtape Commands

<table>
<thead>
<tr>
<th>Octal</th>
<th>Hex</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100000</td>
<td>8000</td>
<td>Select Transport (7 and 9 track)</td>
</tr>
<tr>
<td>000040</td>
<td>0020</td>
<td>Rewind to BOT (7 and 9 track)</td>
</tr>
<tr>
<td>022100</td>
<td>2440</td>
<td>Backspace File Mark, 9-track</td>
</tr>
<tr>
<td>020100</td>
<td>2040</td>
<td>Backspace File Mark, 7-track</td>
</tr>
<tr>
<td>062100</td>
<td>6440</td>
<td>Backspace Record, 9-track</td>
</tr>
<tr>
<td>060100</td>
<td>6040</td>
<td>Backspace Record, 7-track</td>
</tr>
<tr>
<td>022220</td>
<td>2490</td>
<td>Write File Mark, 9-track</td>
</tr>
<tr>
<td>020220</td>
<td>2090</td>
<td>Write File Mark, 7-track</td>
</tr>
<tr>
<td>062200</td>
<td>6480</td>
<td>Forward Space Record, 9-track</td>
</tr>
<tr>
<td>060200</td>
<td>6080</td>
<td>Forward Space Record, 7-track</td>
</tr>
<tr>
<td>022200</td>
<td>2480</td>
<td>Forward Space File Mark, 9-track</td>
</tr>
<tr>
<td>Octal</td>
<td>Hex</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>020200</td>
<td>2080</td>
<td>Forward Space File Mark, 7-track</td>
</tr>
<tr>
<td>042220</td>
<td>4490</td>
<td>Write Record One Char/Word, 9-track</td>
</tr>
<tr>
<td>042620</td>
<td>4590</td>
<td>Write Record Two Char/Word, 9-track</td>
</tr>
<tr>
<td>042200</td>
<td>4490</td>
<td>Read Record One Char/Word, 9-track</td>
</tr>
<tr>
<td>042600</td>
<td>4580</td>
<td>Read Record Two Char/Word, 9-track</td>
</tr>
<tr>
<td>052200</td>
<td>5480</td>
<td>Read/Correct Record One Char/Word, 9-track</td>
</tr>
<tr>
<td>052600</td>
<td>5580</td>
<td>Read/Correct Record Two Char/Word, 9-track</td>
</tr>
<tr>
<td>040220</td>
<td>4090</td>
<td>Write Binary Record One Char/Word, 7-track</td>
</tr>
<tr>
<td>040620</td>
<td>4190</td>
<td>Write Binary Record Two Char/Word, 7-track</td>
</tr>
<tr>
<td>044220</td>
<td>4890</td>
<td>Write BCD Record One Char/Word, 7-track</td>
</tr>
<tr>
<td>044620</td>
<td>4990</td>
<td>Write BCD Record Two Char/Word, 7-track</td>
</tr>
<tr>
<td>040200</td>
<td>4080</td>
<td>Read Binary Record One Char/Word, 7-track</td>
</tr>
<tr>
<td>040600</td>
<td>4180</td>
<td>Read Binary Record Two Char/Word, 7-track</td>
</tr>
<tr>
<td>044200</td>
<td>4880</td>
<td>Read BCD Record One Char/Word, 7-track</td>
</tr>
<tr>
<td>044600</td>
<td>4980</td>
<td>Read BCD Record Two Char/Word, 7-track</td>
</tr>
<tr>
<td>140000</td>
<td>C000</td>
<td>Return controller ID</td>
</tr>
<tr>
<td>100020</td>
<td>8010</td>
<td>Erase 3 inch gap (vers. 2 or 3 controller)</td>
</tr>
<tr>
<td>100040</td>
<td>8020</td>
<td>Unload; rewind and plac offline (2, 3)</td>
</tr>
<tr>
<td>100060</td>
<td>8030</td>
<td>Set density to 800 bpi (2 only)</td>
</tr>
<tr>
<td>100100</td>
<td>8040</td>
<td>Set density to 1600 bpi (2, 3)</td>
</tr>
<tr>
<td>100120</td>
<td>8050</td>
<td>Set density to 6250 bpi (3 only)</td>
</tr>
<tr>
<td>100140</td>
<td>8060</td>
<td>Enable front panel density select (3)</td>
</tr>
<tr>
<td>100160</td>
<td>8070</td>
<td>Set speed to 25 IPS (future)</td>
</tr>
<tr>
<td>100200</td>
<td>8080</td>
<td>Set speed to 100 IPS (future)</td>
</tr>
<tr>
<td>043500</td>
<td>4740</td>
<td>Read record backwards (3 only)</td>
</tr>
</tbody>
</table>
9.8.3. Magtape Status

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>Octal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Parity Error</td>
<td>100000</td>
<td>8000</td>
</tr>
<tr>
<td>2</td>
<td>Runaway Tape</td>
<td>040000</td>
<td>4000</td>
</tr>
<tr>
<td>3</td>
<td>CRC Error</td>
<td>020000</td>
<td>2000</td>
</tr>
<tr>
<td>4</td>
<td>LRC Error</td>
<td>010000</td>
<td>1000</td>
</tr>
<tr>
<td>5</td>
<td>Low DMx Range</td>
<td>004000</td>
<td>0800</td>
</tr>
<tr>
<td>6</td>
<td>Permanent Error</td>
<td>002000</td>
<td>0400</td>
</tr>
<tr>
<td>7</td>
<td>Read-After-Write (RAW) Error</td>
<td>001000</td>
<td>0200</td>
</tr>
<tr>
<td>8</td>
<td>File Mark Detected</td>
<td>000400</td>
<td>0100</td>
</tr>
<tr>
<td>9</td>
<td>Ready</td>
<td>000200</td>
<td>0080</td>
</tr>
<tr>
<td>10</td>
<td>Online</td>
<td>000100</td>
<td>0040</td>
</tr>
<tr>
<td>11</td>
<td>End of Tape Detected</td>
<td>000040</td>
<td>0020</td>
</tr>
<tr>
<td>12</td>
<td>Rewinding</td>
<td>000020</td>
<td>0010</td>
</tr>
<tr>
<td>13</td>
<td>Beginning of Tape (at Load Point)</td>
<td>000010</td>
<td>0008</td>
</tr>
<tr>
<td>14</td>
<td>Tape is Write-Protected</td>
<td>000004</td>
<td>0004</td>
</tr>
<tr>
<td>15</td>
<td>DMx Overrun</td>
<td>000002</td>
<td>0002</td>
</tr>
<tr>
<td>16</td>
<td>Rewind Complete</td>
<td>000001</td>
<td>0001</td>
</tr>
</tbody>
</table>

Normal Completion: 000300 or 000304 (00C0 or 00C4)

9.9. PROGRAMMED I/O (PIO)

9.9.1. OCP — Output Control Pulse
03FFDD FF=Function, DD=Device Address

9.9.2. SKS — Skip on Condition
07CCDD CC=Condition, DD=Device Address

9.9.3. INA — Input to A-Register
13FFDD FF=Function, DD=Device Address
No skip for device '20 Always skips if status register input.

9.9.4. OTA — Output from A=Register
17FFDD FF=Function, DD=Device Address
No skip if device '20.
9.9.5. Standard Functions

<table>
<thead>
<tr>
<th>FF</th>
<th>OCP</th>
<th>SKS</th>
<th>INA</th>
<th>OTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td></td>
<td>Ready</td>
<td>Data Reg</td>
<td></td>
</tr>
<tr>
<td>01</td>
<td></td>
<td>Not Busy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>02</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>03</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>04</td>
<td></td>
<td>Not Interrupting</td>
<td></td>
<td></td>
</tr>
<tr>
<td>05</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>06</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>Normal Mode</td>
<td></td>
<td>DMx Channel</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>Diagnostic Mode</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>Ack Interrupt</td>
<td></td>
<td>Int Vect Addr</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>Set Int Mask</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>Reset Int Mask</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>Initialize</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9.10. RS-232-C pin-outs

<table>
<thead>
<tr>
<th>Pin</th>
<th>Abbrev</th>
<th>Description</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FG</td>
<td>Protective (frame) ground</td>
<td>DTE</td>
</tr>
<tr>
<td>2</td>
<td>TxD</td>
<td>Transmitted data</td>
<td>DCE</td>
</tr>
<tr>
<td>3</td>
<td>RxD</td>
<td>Received data</td>
<td>DCE</td>
</tr>
<tr>
<td>4</td>
<td>RTS</td>
<td>Request To Send</td>
<td>DTE</td>
</tr>
<tr>
<td>5</td>
<td>CTS</td>
<td>Clear To Send</td>
<td>DCE</td>
</tr>
<tr>
<td>6</td>
<td>DSR</td>
<td>Data Set Ready</td>
<td>DCE</td>
</tr>
<tr>
<td>7</td>
<td>SG</td>
<td>Signal ground</td>
<td>DCE</td>
</tr>
<tr>
<td>8</td>
<td>CD</td>
<td>Data Carrier Detect</td>
<td>DCE</td>
</tr>
<tr>
<td>9</td>
<td>-</td>
<td>Reserved for test</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>-</td>
<td>Reserved for test</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>-</td>
<td>Unassigned</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>SCD</td>
<td>Sec. Carrier Detect</td>
<td>DCE</td>
</tr>
<tr>
<td>13</td>
<td>SCTS</td>
<td>Sec. Clear to Send</td>
<td>DCE</td>
</tr>
<tr>
<td>14</td>
<td>STxD</td>
<td>Sec. Transmitted Data</td>
<td>DTE</td>
</tr>
<tr>
<td>15</td>
<td>TxC</td>
<td>Trans. Signal Element Timing</td>
<td>DCE</td>
</tr>
<tr>
<td>16</td>
<td>S RxD</td>
<td>Sec. Received Data</td>
<td>DCE</td>
</tr>
<tr>
<td>17</td>
<td>RxC</td>
<td>Rec. Signal Element</td>
<td>DTE</td>
</tr>
<tr>
<td>18</td>
<td>-</td>
<td>Unassigned</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>S RTS</td>
<td>Sec. Request to Send</td>
<td>DTE</td>
</tr>
<tr>
<td>20</td>
<td>DTR</td>
<td>Data Terminal Ready</td>
<td>DTE</td>
</tr>
<tr>
<td>21</td>
<td>SQ</td>
<td>Data Signal Quality</td>
<td>DCE</td>
</tr>
<tr>
<td>22</td>
<td>RI</td>
<td>Ring Indicator</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>-</td>
<td>Data Rate Selector</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>ETxC</td>
<td>Trans. Signal Element Timing</td>
<td>DTE</td>
</tr>
<tr>
<td>25</td>
<td>-</td>
<td>Unassigned</td>
<td></td>
</tr>
</tbody>
</table>
## Appendix A

### ASCII character set

<table>
<thead>
<tr>
<th>Octal Left</th>
<th>Hex</th>
<th>Dec</th>
<th>Char</th>
<th>Octal Left</th>
<th>Hex</th>
<th>Dec</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>000</td>
<td>0</td>
<td>NUL @</td>
<td>200</td>
<td>100</td>
<td>80</td>
<td>128</td>
</tr>
<tr>
<td>001</td>
<td>004</td>
<td>1</td>
<td>SOH ^A</td>
<td>201</td>
<td>1004</td>
<td>81</td>
<td>129</td>
</tr>
<tr>
<td>002</td>
<td>010</td>
<td>2</td>
<td>STX ^B</td>
<td>202</td>
<td>1010</td>
<td>82</td>
<td>130</td>
</tr>
<tr>
<td>003</td>
<td>014</td>
<td>3</td>
<td>ETX ^C</td>
<td>203</td>
<td>1014</td>
<td>83</td>
<td>131</td>
</tr>
<tr>
<td>004</td>
<td>020</td>
<td>4</td>
<td>EOT ^D</td>
<td>204</td>
<td>1020</td>
<td>84</td>
<td>132</td>
</tr>
<tr>
<td>005</td>
<td>024</td>
<td>5</td>
<td>ENQ ^E</td>
<td>205</td>
<td>1024</td>
<td>85</td>
<td>133</td>
</tr>
<tr>
<td>006</td>
<td>030</td>
<td>6</td>
<td>ACK ^F</td>
<td>206</td>
<td>1030</td>
<td>86</td>
<td>134</td>
</tr>
<tr>
<td>007</td>
<td>034</td>
<td>7</td>
<td>BEL ^G</td>
<td>207</td>
<td>1034</td>
<td>87</td>
<td>135</td>
</tr>
<tr>
<td>010</td>
<td>040</td>
<td>8</td>
<td>BS ^H</td>
<td>210</td>
<td>1040</td>
<td>88</td>
<td>136</td>
</tr>
<tr>
<td>011</td>
<td>044</td>
<td>9</td>
<td>HT ^I</td>
<td>211</td>
<td>1044</td>
<td>89</td>
<td>137</td>
</tr>
<tr>
<td>012</td>
<td>050</td>
<td>10</td>
<td>LF ^J</td>
<td>212</td>
<td>1050</td>
<td>90</td>
<td>138</td>
</tr>
<tr>
<td>013</td>
<td>054</td>
<td>11</td>
<td>VT ^K</td>
<td>213</td>
<td>1054</td>
<td>91</td>
<td>139</td>
</tr>
<tr>
<td>014</td>
<td>060</td>
<td>12</td>
<td>FF ^L</td>
<td>214</td>
<td>1060</td>
<td>92</td>
<td>140</td>
</tr>
<tr>
<td>015</td>
<td>064</td>
<td>13</td>
<td>CR ^M</td>
<td>215</td>
<td>1064</td>
<td>93</td>
<td>141</td>
</tr>
<tr>
<td>016</td>
<td>070</td>
<td>14</td>
<td>LS1 ^N</td>
<td>216</td>
<td>1070</td>
<td>94</td>
<td>142</td>
</tr>
<tr>
<td>017</td>
<td>074</td>
<td>15</td>
<td>LS0 ^O</td>
<td>217</td>
<td>1074</td>
<td>95</td>
<td>143</td>
</tr>
<tr>
<td>020</td>
<td>100</td>
<td>16</td>
<td>DLE ^P</td>
<td>220</td>
<td>1100</td>
<td>96</td>
<td>144</td>
</tr>
<tr>
<td>021</td>
<td>104</td>
<td>17</td>
<td>DC1 ^Q</td>
<td>221</td>
<td>1104</td>
<td>97</td>
<td>145</td>
</tr>
<tr>
<td>022</td>
<td>110</td>
<td>18</td>
<td>DC2 ^R</td>
<td>222</td>
<td>1108</td>
<td>98</td>
<td>146</td>
</tr>
<tr>
<td>023</td>
<td>114</td>
<td>19</td>
<td>DC3 ^S</td>
<td>223</td>
<td>1112</td>
<td>99</td>
<td>147</td>
</tr>
<tr>
<td>024</td>
<td>120</td>
<td>20</td>
<td>DC4 ^T</td>
<td>224</td>
<td>1116</td>
<td>100</td>
<td>148</td>
</tr>
<tr>
<td>025</td>
<td>124</td>
<td>21</td>
<td>NAK ^U</td>
<td>225</td>
<td>1120</td>
<td>101</td>
<td>149</td>
</tr>
<tr>
<td>026</td>
<td>130</td>
<td>22</td>
<td>SYN ^V</td>
<td>226</td>
<td>1124</td>
<td>102</td>
<td>150</td>
</tr>
<tr>
<td>027</td>
<td>134</td>
<td>23</td>
<td>ETB ^W</td>
<td>227</td>
<td>1128</td>
<td>103</td>
<td>151</td>
</tr>
<tr>
<td>030</td>
<td>140</td>
<td>24</td>
<td>CAN ^X</td>
<td>230</td>
<td>1132</td>
<td>104</td>
<td>152</td>
</tr>
<tr>
<td>031</td>
<td>144</td>
<td>25</td>
<td>EM ^Y</td>
<td>231</td>
<td>1136</td>
<td>105</td>
<td>153</td>
</tr>
<tr>
<td>032</td>
<td>150</td>
<td>26</td>
<td>SUB ^Z</td>
<td>232</td>
<td>1140</td>
<td>106</td>
<td>154</td>
</tr>
<tr>
<td>033</td>
<td>154</td>
<td>27</td>
<td>ESC ^[</td>
<td>233</td>
<td>1144</td>
<td>107</td>
<td>155</td>
</tr>
<tr>
<td>034</td>
<td>160</td>
<td>28</td>
<td>FS ^| 234</td>
<td>1148</td>
<td>108</td>
<td>156</td>
<td></td>
</tr>
<tr>
<td>035</td>
<td>164</td>
<td>29</td>
<td>GS ^</td>
<td>235</td>
<td>1152</td>
<td>109</td>
<td>157</td>
</tr>
<tr>
<td>036</td>
<td>170</td>
<td>30</td>
<td>RS ^A</td>
<td>236</td>
<td>1156</td>
<td>110</td>
<td>158</td>
</tr>
<tr>
<td>037</td>
<td>174</td>
<td>31</td>
<td>US ^B</td>
<td>237</td>
<td>1160</td>
<td>111</td>
<td>159</td>
</tr>
<tr>
<td>040</td>
<td>200</td>
<td>32</td>
<td>Space</td>
<td>240</td>
<td>1200</td>
<td>112</td>
<td>160</td>
</tr>
<tr>
<td>041</td>
<td>204</td>
<td>33</td>
<td>&quot;</td>
<td>241</td>
<td>1204</td>
<td>113</td>
<td>161</td>
</tr>
<tr>
<td>042</td>
<td>210</td>
<td>34</td>
<td>!</td>
<td>242</td>
<td>1208</td>
<td>114</td>
<td>162</td>
</tr>
<tr>
<td>043</td>
<td>214</td>
<td>35</td>
<td>#</td>
<td>243</td>
<td>1212</td>
<td>115</td>
<td>163</td>
</tr>
<tr>
<td>044</td>
<td>220</td>
<td>36</td>
<td>$</td>
<td>244</td>
<td>1216</td>
<td>116</td>
<td>164</td>
</tr>
<tr>
<td>045</td>
<td>224</td>
<td>37</td>
<td>%</td>
<td>245</td>
<td>1220</td>
<td>117</td>
<td>165</td>
</tr>
<tr>
<td>046</td>
<td>230</td>
<td>38</td>
<td>&amp;</td>
<td>246</td>
<td>1224</td>
<td>118</td>
<td>166</td>
</tr>
<tr>
<td>047</td>
<td>234</td>
<td>39</td>
<td>(</td>
<td>247</td>
<td>1228</td>
<td>119</td>
<td>167</td>
</tr>
<tr>
<td>050</td>
<td>240</td>
<td>40</td>
<td>)</td>
<td>248</td>
<td>1232</td>
<td>120</td>
<td>168</td>
</tr>
<tr>
<td>051</td>
<td>244</td>
<td>41</td>
<td>_</td>
<td>249</td>
<td>1236</td>
<td>121</td>
<td>169</td>
</tr>
<tr>
<td>052</td>
<td>250</td>
<td>42</td>
<td>+</td>
<td>250</td>
<td>1240</td>
<td>122</td>
<td>170</td>
</tr>
<tr>
<td>053</td>
<td>254</td>
<td>43</td>
<td>=</td>
<td>251</td>
<td>1244</td>
<td>123</td>
<td>171</td>
</tr>
<tr>
<td>054</td>
<td>260</td>
<td>44</td>
<td>\</td>
<td>252</td>
<td>1248</td>
<td>124</td>
<td>172</td>
</tr>
<tr>
<td>055</td>
<td>264</td>
<td>45</td>
<td>(</td>
<td>253</td>
<td>1252</td>
<td>125</td>
<td>173</td>
</tr>
<tr>
<td>056</td>
<td>270</td>
<td>46</td>
<td>)</td>
<td>254</td>
<td>1256</td>
<td>126</td>
<td>174</td>
</tr>
<tr>
<td>057</td>
<td>274</td>
<td>47</td>
<td>/</td>
<td>255</td>
<td>1260</td>
<td>127</td>
<td>175</td>
</tr>
<tr>
<td>060</td>
<td>300</td>
<td>48</td>
<td>0</td>
<td>256</td>
<td>1264</td>
<td>128</td>
<td>176</td>
</tr>
<tr>
<td>061</td>
<td>304</td>
<td>49</td>
<td>1</td>
<td>257</td>
<td>1268</td>
<td>129</td>
<td>177</td>
</tr>
<tr>
<td>062</td>
<td>310</td>
<td>50</td>
<td>2</td>
<td>258</td>
<td>1272</td>
<td>130</td>
<td>178</td>
</tr>
<tr>
<td>063</td>
<td>314</td>
<td>51</td>
<td>3</td>
<td>259</td>
<td>1276</td>
<td>131</td>
<td>179</td>
</tr>
<tr>
<td>064</td>
<td>320</td>
<td>52</td>
<td>4</td>
<td>260</td>
<td>1280</td>
<td>132</td>
<td>180</td>
</tr>
<tr>
<td>065</td>
<td>324</td>
<td>53</td>
<td>5</td>
<td>261</td>
<td>1284</td>
<td>133</td>
<td>181</td>
</tr>
<tr>
<td>066</td>
<td>330</td>
<td>54</td>
<td>6</td>
<td>262</td>
<td>1288</td>
<td>134</td>
<td>182</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Octal</th>
<th>Octal Left</th>
<th>Hex</th>
<th>Dec</th>
<th>Char</th>
<th>Octal</th>
<th>Octal Left</th>
<th>Hex</th>
<th>Dec</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>167</td>
<td>0734</td>
<td>77</td>
<td>119</td>
<td>w</td>
<td>367</td>
<td>1734</td>
<td>F7</td>
<td>247</td>
<td></td>
</tr>
<tr>
<td>170</td>
<td>0740</td>
<td>78</td>
<td>120</td>
<td>x</td>
<td>370</td>
<td>1740</td>
<td>F8</td>
<td>248</td>
<td></td>
</tr>
<tr>
<td>171</td>
<td>0744</td>
<td>79</td>
<td>121</td>
<td>y</td>
<td>371</td>
<td>1744</td>
<td>F9</td>
<td>249</td>
<td></td>
</tr>
<tr>
<td>172</td>
<td>0750</td>
<td>7A</td>
<td>122</td>
<td>z</td>
<td>372</td>
<td>1750</td>
<td>FA</td>
<td>250</td>
<td></td>
</tr>
<tr>
<td>173</td>
<td>0754</td>
<td>7B</td>
<td>123</td>
<td>{</td>
<td>373</td>
<td>1754</td>
<td>FB</td>
<td>251</td>
<td>R</td>
</tr>
<tr>
<td>174</td>
<td>0760</td>
<td>7C</td>
<td>124</td>
<td>}</td>
<td>374</td>
<td>1760</td>
<td>FC</td>
<td>252</td>
<td>R</td>
</tr>
<tr>
<td>175</td>
<td>0764</td>
<td>7D</td>
<td>125</td>
<td></td>
<td>375</td>
<td>1764</td>
<td>FD</td>
<td>253</td>
<td>R</td>
</tr>
<tr>
<td>176</td>
<td>0770</td>
<td>7E</td>
<td>126</td>
<td>~</td>
<td>376</td>
<td>1770</td>
<td>FE</td>
<td>254</td>
<td>R</td>
</tr>
<tr>
<td>177</td>
<td>0774</td>
<td>7F</td>
<td>127</td>
<td>DEL</td>
<td>377</td>
<td>1774</td>
<td>FF</td>
<td>255</td>
<td></td>
</tr>
</tbody>
</table>
### Appendix B
Conversion tables

#### B.1. Octal-Decimal Conversion Table

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Prime Restricted
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>720</td>
<td>464</td>
<td>465</td>
<td>466</td>
<td>467</td>
<td>468</td>
<td>469</td>
<td>470</td>
</tr>
<tr>
<td>730</td>
<td>472</td>
<td>473</td>
<td>474</td>
<td>475</td>
<td>476</td>
<td>477</td>
<td>478</td>
</tr>
<tr>
<td>740</td>
<td>480</td>
<td>481</td>
<td>482</td>
<td>483</td>
<td>484</td>
<td>485</td>
<td>486</td>
</tr>
<tr>
<td>750</td>
<td>488</td>
<td>489</td>
<td>490</td>
<td>491</td>
<td>492</td>
<td>493</td>
<td>494</td>
</tr>
<tr>
<td>760</td>
<td>496</td>
<td>497</td>
<td>498</td>
<td>499</td>
<td>500</td>
<td>501</td>
<td>502</td>
</tr>
<tr>
<td>770</td>
<td>504</td>
<td>505</td>
<td>506</td>
<td>507</td>
<td>508</td>
<td>509</td>
<td>510</td>
</tr>
<tr>
<td>1000</td>
<td>512</td>
<td>513</td>
<td>514</td>
<td>515</td>
<td>516</td>
<td>517</td>
<td>518</td>
</tr>
<tr>
<td>1010</td>
<td>520</td>
<td>521</td>
<td>522</td>
<td>523</td>
<td>524</td>
<td>525</td>
<td>526</td>
</tr>
<tr>
<td>1020</td>
<td>528</td>
<td>529</td>
<td>530</td>
<td>531</td>
<td>532</td>
<td>533</td>
<td>534</td>
</tr>
<tr>
<td>1030</td>
<td>536</td>
<td>537</td>
<td>538</td>
<td>539</td>
<td>540</td>
<td>541</td>
<td>542</td>
</tr>
<tr>
<td>1040</td>
<td>544</td>
<td>545</td>
<td>546</td>
<td>547</td>
<td>548</td>
<td>549</td>
<td>550</td>
</tr>
<tr>
<td>1050</td>
<td>552</td>
<td>553</td>
<td>554</td>
<td>555</td>
<td>556</td>
<td>557</td>
<td>558</td>
</tr>
<tr>
<td>1060</td>
<td>560</td>
<td>561</td>
<td>562</td>
<td>563</td>
<td>564</td>
<td>565</td>
<td>566</td>
</tr>
<tr>
<td>1070</td>
<td>568</td>
<td>569</td>
<td>570</td>
<td>571</td>
<td>572</td>
<td>573</td>
<td>574</td>
</tr>
<tr>
<td>1100</td>
<td>576</td>
<td>577</td>
<td>578</td>
<td>579</td>
<td>580</td>
<td>581</td>
<td>582</td>
</tr>
<tr>
<td>1150</td>
<td>584</td>
<td>585</td>
<td>586</td>
<td>587</td>
<td>588</td>
<td>589</td>
<td>590</td>
</tr>
<tr>
<td>1160</td>
<td>592</td>
<td>593</td>
<td>594</td>
<td>595</td>
<td>596</td>
<td>597</td>
<td>598</td>
</tr>
<tr>
<td>1170</td>
<td>600</td>
<td>601</td>
<td>602</td>
<td>603</td>
<td>604</td>
<td>605</td>
<td>606</td>
</tr>
<tr>
<td>1180</td>
<td>608</td>
<td>609</td>
<td>610</td>
<td>611</td>
<td>612</td>
<td>613</td>
<td>614</td>
</tr>
<tr>
<td>1190</td>
<td>616</td>
<td>617</td>
<td>618</td>
<td>619</td>
<td>620</td>
<td>621</td>
<td>622</td>
</tr>
<tr>
<td>1200</td>
<td>620</td>
<td>621</td>
<td>622</td>
<td>623</td>
<td>624</td>
<td>625</td>
<td>626</td>
</tr>
<tr>
<td>1210</td>
<td>628</td>
<td>629</td>
<td>630</td>
<td>631</td>
<td>632</td>
<td>633</td>
<td>634</td>
</tr>
<tr>
<td>1220</td>
<td>636</td>
<td>637</td>
<td>638</td>
<td>639</td>
<td>640</td>
<td>641</td>
<td>642</td>
</tr>
<tr>
<td>1230</td>
<td>644</td>
<td>645</td>
<td>646</td>
<td>647</td>
<td>648</td>
<td>649</td>
<td>650</td>
</tr>
<tr>
<td>1240</td>
<td>656</td>
<td>657</td>
<td>658</td>
<td>659</td>
<td>660</td>
<td>661</td>
<td>662</td>
</tr>
<tr>
<td>1250</td>
<td>664</td>
<td>665</td>
<td>666</td>
<td>667</td>
<td>668</td>
<td>669</td>
<td>670</td>
</tr>
<tr>
<td>1270</td>
<td>672</td>
<td>673</td>
<td>674</td>
<td>675</td>
<td>676</td>
<td>677</td>
<td>678</td>
</tr>
<tr>
<td>1280</td>
<td>680</td>
<td>681</td>
<td>682</td>
<td>683</td>
<td>684</td>
<td>685</td>
<td>686</td>
</tr>
<tr>
<td>1290</td>
<td>688</td>
<td>689</td>
<td>690</td>
<td>691</td>
<td>692</td>
<td>693</td>
<td>694</td>
</tr>
<tr>
<td>1300</td>
<td>696</td>
<td>697</td>
<td>698</td>
<td>699</td>
<td>700</td>
<td>701</td>
<td>702</td>
</tr>
<tr>
<td>1330</td>
<td>704</td>
<td>705</td>
<td>706</td>
<td>707</td>
<td>708</td>
<td>709</td>
<td>710</td>
</tr>
<tr>
<td>1340</td>
<td>712</td>
<td>713</td>
<td>714</td>
<td>715</td>
<td>716</td>
<td>717</td>
<td>718</td>
</tr>
<tr>
<td>1350</td>
<td>720</td>
<td>721</td>
<td>722</td>
<td>723</td>
<td>724</td>
<td>725</td>
<td>726</td>
</tr>
<tr>
<td>1370</td>
<td>728</td>
<td>729</td>
<td>730</td>
<td>731</td>
<td>732</td>
<td>733</td>
<td>734</td>
</tr>
<tr>
<td>1380</td>
<td>736</td>
<td>737</td>
<td>738</td>
<td>739</td>
<td>740</td>
<td>741</td>
<td>742</td>
</tr>
<tr>
<td>1400</td>
<td>744</td>
<td>745</td>
<td>746</td>
<td>747</td>
<td>748</td>
<td>749</td>
<td>750</td>
</tr>
<tr>
<td>1430</td>
<td>752</td>
<td>753</td>
<td>754</td>
<td>755</td>
<td>756</td>
<td>757</td>
<td>758</td>
</tr>
<tr>
<td>1440</td>
<td>760</td>
<td>761</td>
<td>762</td>
<td>763</td>
<td>764</td>
<td>765</td>
<td>766</td>
</tr>
<tr>
<td>1450</td>
<td>768</td>
<td>769</td>
<td>770</td>
<td>771</td>
<td>772</td>
<td>773</td>
<td>774</td>
</tr>
<tr>
<td>1470</td>
<td>776</td>
<td>777</td>
<td>778</td>
<td>779</td>
<td>780</td>
<td>781</td>
<td>782</td>
</tr>
<tr>
<td>1490</td>
<td>784</td>
<td>785</td>
<td>786</td>
<td>787</td>
<td>788</td>
<td>789</td>
<td>790</td>
</tr>
<tr>
<td>1500</td>
<td>792</td>
<td>793</td>
<td>794</td>
<td>795</td>
<td>796</td>
<td>797</td>
<td>798</td>
</tr>
<tr>
<td>1520</td>
<td>800</td>
<td>801</td>
<td>802</td>
<td>803</td>
<td>804</td>
<td>805</td>
<td>806</td>
</tr>
<tr>
<td>1550</td>
<td>808</td>
<td>809</td>
<td>810</td>
<td>811</td>
<td>812</td>
<td>813</td>
<td>814</td>
</tr>
<tr>
<td>1570</td>
<td>816</td>
<td>817</td>
<td>818</td>
<td>819</td>
<td>820</td>
<td>821</td>
<td>822</td>
</tr>
<tr>
<td>1590</td>
<td>824</td>
<td>825</td>
<td>826</td>
<td>827</td>
<td>828</td>
<td>829</td>
<td>830</td>
</tr>
<tr>
<td>1600</td>
<td>832</td>
<td>833</td>
<td>834</td>
<td>835</td>
<td>836</td>
<td>837</td>
<td>838</td>
</tr>
<tr>
<td>1610</td>
<td>840</td>
<td>841</td>
<td>842</td>
<td>843</td>
<td>844</td>
<td>845</td>
<td>846</td>
</tr>
<tr>
<td>1620</td>
<td>848</td>
<td>849</td>
<td>850</td>
<td>851</td>
<td>852</td>
<td>853</td>
<td>854</td>
</tr>
<tr>
<td>1650</td>
<td>856</td>
<td>857</td>
<td>858</td>
<td>859</td>
<td>860</td>
<td>861</td>
<td>862</td>
</tr>
<tr>
<td>1660</td>
<td>864</td>
<td>865</td>
<td>866</td>
<td>867</td>
<td>868</td>
<td>869</td>
<td>870</td>
</tr>
<tr>
<td>1670</td>
<td>872</td>
<td>873</td>
<td>874</td>
<td>875</td>
<td>876</td>
<td>877</td>
<td>878</td>
</tr>
<tr>
<td>1680</td>
<td>880</td>
<td>881</td>
<td>882</td>
<td>883</td>
<td>884</td>
<td>885</td>
<td>886</td>
</tr>
<tr>
<td>1700</td>
<td>888</td>
<td>889</td>
<td>890</td>
<td>891</td>
<td>892</td>
<td>893</td>
<td>894</td>
</tr>
<tr>
<td>1710</td>
<td>896</td>
<td>897</td>
<td>898</td>
<td>899</td>
<td>900</td>
<td>901</td>
<td>902</td>
</tr>
<tr>
<td>1720</td>
<td>904</td>
<td>905</td>
<td>906</td>
<td>907</td>
<td>908</td>
<td>909</td>
<td>910</td>
</tr>
<tr>
<td>1740</td>
<td>912</td>
<td>913</td>
<td>914</td>
<td>915</td>
<td>916</td>
<td>917</td>
<td>918</td>
</tr>
<tr>
<td>1760</td>
<td>920</td>
<td>921</td>
<td>922</td>
<td>923</td>
<td>924</td>
<td>925</td>
<td>926</td>
</tr>
<tr>
<td>1770</td>
<td>928</td>
<td>929</td>
<td>930</td>
<td>931</td>
<td>932</td>
<td>933</td>
<td>934</td>
</tr>
<tr>
<td>1780</td>
<td>936</td>
<td>937</td>
<td>938</td>
<td>939</td>
<td>940</td>
<td>941</td>
<td>942</td>
</tr>
<tr>
<td>1790</td>
<td>944</td>
<td>945</td>
<td>946</td>
<td>947</td>
<td>948</td>
<td>949</td>
<td>950</td>
</tr>
<tr>
<td>1800</td>
<td>952</td>
<td>953</td>
<td>954</td>
<td>955</td>
<td>956</td>
<td>957</td>
<td>958</td>
</tr>
<tr>
<td>1820</td>
<td>960</td>
<td>961</td>
<td>962</td>
<td>963</td>
<td>964</td>
<td>965</td>
<td>966</td>
</tr>
<tr>
<td>1830</td>
<td>968</td>
<td>969</td>
<td>970</td>
<td>971</td>
<td>972</td>
<td>973</td>
<td>974</td>
</tr>
<tr>
<td>1840</td>
<td>976</td>
<td>977</td>
<td>978</td>
<td>979</td>
<td>980</td>
<td>981</td>
<td>982</td>
</tr>
<tr>
<td>1860</td>
<td>984</td>
<td>985</td>
<td>986</td>
<td>987</td>
<td>988</td>
<td>989</td>
<td>990</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>---</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>1740</td>
<td>992</td>
<td>993</td>
<td>994</td>
<td>995</td>
<td>996</td>
<td>997</td>
<td>998</td>
</tr>
<tr>
<td>1750</td>
<td>1000</td>
<td>1001</td>
<td>1002</td>
<td>1003</td>
<td>1004</td>
<td>1005</td>
<td>1006</td>
</tr>
<tr>
<td>1760</td>
<td>1008</td>
<td>1009</td>
<td>1010</td>
<td>1011</td>
<td>1012</td>
<td>1013</td>
<td>1014</td>
</tr>
<tr>
<td>1770</td>
<td>1016</td>
<td>1017</td>
<td>1018</td>
<td>1019</td>
<td>1020</td>
<td>1021</td>
<td>1022</td>
</tr>
</tbody>
</table>
## Appendix C
### Powers of Two

<table>
<thead>
<tr>
<th>$n$</th>
<th>$2^n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
</tr>
<tr>
<td>8</td>
<td>256</td>
</tr>
<tr>
<td>9</td>
<td>512</td>
</tr>
<tr>
<td>10</td>
<td>1024</td>
</tr>
<tr>
<td>11</td>
<td>2048</td>
</tr>
<tr>
<td>12</td>
<td>4096</td>
</tr>
<tr>
<td>13</td>
<td>8192</td>
</tr>
<tr>
<td>14</td>
<td>16384</td>
</tr>
<tr>
<td>15</td>
<td>32768</td>
</tr>
<tr>
<td>16</td>
<td>65536</td>
</tr>
<tr>
<td>17</td>
<td>131072</td>
</tr>
<tr>
<td>18</td>
<td>262144</td>
</tr>
<tr>
<td>19</td>
<td>524288</td>
</tr>
<tr>
<td>20</td>
<td>1048576</td>
</tr>
<tr>
<td>21</td>
<td>2097152</td>
</tr>
<tr>
<td>22</td>
<td>4194304</td>
</tr>
<tr>
<td>23</td>
<td>8388608</td>
</tr>
<tr>
<td>24</td>
<td>16777216</td>
</tr>
<tr>
<td>25</td>
<td>33554432</td>
</tr>
<tr>
<td>26</td>
<td>67108864</td>
</tr>
<tr>
<td>27</td>
<td>134217728</td>
</tr>
<tr>
<td>28</td>
<td>268435456</td>
</tr>
<tr>
<td>29</td>
<td>536870912</td>
</tr>
<tr>
<td>30</td>
<td>1073741824</td>
</tr>
<tr>
<td>31</td>
<td>2147483648</td>
</tr>
<tr>
<td>32</td>
<td>4294967296</td>
</tr>
<tr>
<td>33</td>
<td>8589934592</td>
</tr>
<tr>
<td>34</td>
<td>17179869184</td>
</tr>
<tr>
<td>35</td>
<td>34359738368</td>
</tr>
<tr>
<td>36</td>
<td>68719476736</td>
</tr>
<tr>
<td>37</td>
<td>137438953472</td>
</tr>
<tr>
<td>38</td>
<td>274877906944</td>
</tr>
<tr>
<td>39</td>
<td>549755813888</td>
</tr>
<tr>
<td>40</td>
<td>1099511627776</td>
</tr>
<tr>
<td>41</td>
<td>2199023255552</td>
</tr>
<tr>
<td>42</td>
<td>4398046511104</td>
</tr>
<tr>
<td>43</td>
<td>8796093022208</td>
</tr>
<tr>
<td>44</td>
<td>17592186044416</td>
</tr>
<tr>
<td>45</td>
<td>35184372088832</td>
</tr>
<tr>
<td>46</td>
<td>70368744177664</td>
</tr>
<tr>
<td>47</td>
<td>140737488355328</td>
</tr>
<tr>
<td>48</td>
<td>281474976710656</td>
</tr>
<tr>
<td>49</td>
<td>562949953421312</td>
</tr>
<tr>
<td>50</td>
<td>1125899906842624</td>
</tr>
<tr>
<td>51</td>
<td>2251799813685248</td>
</tr>
<tr>
<td>52</td>
<td>4503599627370496</td>
</tr>
<tr>
<td>53</td>
<td>9007199254740992</td>
</tr>
<tr>
<td>54</td>
<td>18014398509481984</td>
</tr>
<tr>
<td>55</td>
<td>36028797018963968</td>
</tr>
<tr>
<td>56</td>
<td>72057594037927936</td>
</tr>
<tr>
<td>57</td>
<td>144115188075855872</td>
</tr>
<tr>
<td>58</td>
<td>288230376151711744</td>
</tr>
<tr>
<td>59</td>
<td>576460752303423488</td>
</tr>
<tr>
<td>60</td>
<td>1152921504606846976</td>
</tr>
<tr>
<td>$n$</td>
<td>$2^n$</td>
</tr>
<tr>
<td>-----</td>
<td>------------------------</td>
</tr>
<tr>
<td>61</td>
<td>2305 84300 92136 93952</td>
</tr>
<tr>
<td>62</td>
<td>4611 68601 84273 87904</td>
</tr>
<tr>
<td>63</td>
<td>9223 37203 66547 75808</td>
</tr>
<tr>
<td>64</td>
<td>18446 74407 37095 51616</td>
</tr>
<tr>
<td>n</td>
<td>2^n</td>
</tr>
<tr>
<td>----</td>
<td>------</td>
</tr>
<tr>
<td>0</td>
<td>1.0</td>
</tr>
<tr>
<td>1</td>
<td>2.0</td>
</tr>
<tr>
<td>2</td>
<td>4.0</td>
</tr>
<tr>
<td>3</td>
<td>0.125</td>
</tr>
<tr>
<td>4</td>
<td>0.625</td>
</tr>
<tr>
<td>5</td>
<td>0.3125</td>
</tr>
<tr>
<td>6</td>
<td>0.15625</td>
</tr>
<tr>
<td>7</td>
<td>0.078125</td>
</tr>
<tr>
<td>8</td>
<td>0.0390625</td>
</tr>
<tr>
<td>9</td>
<td>0.01953125</td>
</tr>
<tr>
<td>10</td>
<td>0.009765625</td>
</tr>
<tr>
<td>11</td>
<td>0.0048828125</td>
</tr>
<tr>
<td>12</td>
<td>0.0024140625</td>
</tr>
<tr>
<td>13</td>
<td>0.00120703125</td>
</tr>
<tr>
<td>14</td>
<td>0.000603515625</td>
</tr>
<tr>
<td>15</td>
<td>0.00030157578125</td>
</tr>
<tr>
<td>16</td>
<td>0.000150787515625</td>
</tr>
<tr>
<td>17</td>
<td>0.0000753937578125</td>
</tr>
<tr>
<td>18</td>
<td>0.00003769687515625</td>
</tr>
<tr>
<td>19</td>
<td>0.0000188484375</td>
</tr>
<tr>
<td>20</td>
<td>0.00000942421875</td>
</tr>
<tr>
<td>21</td>
<td>0.000004712109375</td>
</tr>
<tr>
<td>22</td>
<td>0.0000023560546875</td>
</tr>
<tr>
<td>23</td>
<td>0.00000117802734375</td>
</tr>
<tr>
<td>24</td>
<td>0.000000589013671875</td>
</tr>
<tr>
<td>25</td>
<td>0.0000002945068359375</td>
</tr>
<tr>
<td>26</td>
<td>0.00000014725341771875</td>
</tr>
<tr>
<td>27</td>
<td>0.00000007362670888847</td>
</tr>
<tr>
<td>28</td>
<td>0.000000036813354444235</td>
</tr>
<tr>
<td>29</td>
<td>0.0000000184066772221775</td>
</tr>
<tr>
<td>30</td>
<td>0.00000000920333861108875</td>
</tr>
<tr>
<td>31</td>
<td>0.000000004601669305544375</td>
</tr>
<tr>
<td>32</td>
<td>0.000000002300834652768175</td>
</tr>
<tr>
<td>33</td>
<td>0.0000000115041723133808875</td>
</tr>
<tr>
<td>34</td>
<td>0.00000000575208615669044375</td>
</tr>
<tr>
<td>35</td>
<td>0.00000000287604297834822175</td>
</tr>
<tr>
<td>36</td>
<td>0.00000000143802148917411108875</td>
</tr>
<tr>
<td>37</td>
<td>0.000000000719010744587055544375</td>
</tr>
<tr>
<td>38</td>
<td>0.0000000003595053722935277722</td>
</tr>
<tr>
<td>39</td>
<td>0.0000000001797526861467638861</td>
</tr>
<tr>
<td>40</td>
<td>0.0000000000898763430733819396</td>
</tr>
<tr>
<td>41</td>
<td>0.0000000000449381715366909698</td>
</tr>
<tr>
<td>42</td>
<td>0.0000000000224690857683454899</td>
</tr>
<tr>
<td>43</td>
<td>0.0000000000112345428842724999</td>
</tr>
<tr>
<td>44</td>
<td>0.00000000000561727444129124999</td>
</tr>
<tr>
<td>45</td>
<td>0.00000000000280863722061822499</td>
</tr>
<tr>
<td>46</td>
<td>0.0000000000014043186103091124999</td>
</tr>
<tr>
<td>47</td>
<td>0.0000000000007021593051545604999</td>
</tr>
<tr>
<td>48</td>
<td>0.0000000000003510796525772824999</td>
</tr>
<tr>
<td>49</td>
<td>0.0000000000001755398263841214999</td>
</tr>
<tr>
<td>50</td>
<td>0.00000000000008776991319260724999</td>
</tr>
<tr>
<td>51</td>
<td>0.000000000000043884956596303624999</td>
</tr>
<tr>
<td>52</td>
<td>0.0000000000000219424777981518124999</td>
</tr>
<tr>
<td>53</td>
<td>0.00000000000001097123890088990624999</td>
</tr>
<tr>
<td>54</td>
<td>0.000000000000005485619450444953124999</td>
</tr>
<tr>
<td>55</td>
<td>0.0000000000000027428097252223965624999</td>
</tr>
<tr>
<td>56</td>
<td>0.000000000000001371404862611111823124999</td>
</tr>
<tr>
<td>57</td>
<td>0.000000000000000685702431305555416624999</td>
</tr>
<tr>
<td>58</td>
<td>0.0000000000000003428512156777771083124999</td>
</tr>
<tr>
<td>59</td>
<td>0.0000000000000001714256078388885416624999</td>
</tr>
<tr>
<td>60</td>
<td>0.00000000000000008571280391944427083124999</td>
</tr>
<tr>
<td>61</td>
<td>0.000000000000000042856401959722135416624999</td>
</tr>
<tr>
<td>62</td>
<td>0.0000000000000000214282009798610677083124999</td>
</tr>
<tr>
<td>63</td>
<td>0.00000000000000001071410048993053385416624999</td>
</tr>
<tr>
<td>$n$</td>
<td>$2^n$</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
</tr>
<tr>
<td>64</td>
<td>0.00000 00000 00000 00005 42101 08624 27522 17003 72640 04349 70855 71289 0625</td>
</tr>
</tbody>
</table>
Appendix D
IOA$ usage

Declarations for ioa$, ioa$rs and arguments:

dcl ioa$  entry options(variable);
dcl ioa$rs entry options(variable);
dcl control  char(*); /* control string */
dcl control_length bin; /* length of control string */
dcl output_buffer char(*); /* buffer for ioa$rs */
dcl output_buffer_size bin; /* length of output buffer */
dcl rtn_buffer_length bin; /* chars put into buffer */

    call ioa$(control, control_length [, arg1, ..., arg99]);
    call ioa$rs(output_buffer, output_buffer_size,
              rtn_buffer_length, control, control_length,
              [ arg1, ..., arg99]);

Conversion string format:

%[fw][.s][prec][z][r]type

fw  field width (default 1)
 s  scaling factor (default 0)
 prec  precision (values: 0, 1, 2, 3; default 1)
z  character z, zero fill (default is blank fill)
r  character r, reverse justification (default is right justify)

<table>
<thead>
<tr>
<th>item</th>
<th>type</th>
<th>fw</th>
<th>s</th>
<th>prec</th>
<th>z</th>
<th>r</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>literal %</td>
<td>%</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>decimal</td>
<td>d</td>
<td>0</td>
<td>0</td>
<td>0,1,2,3</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>octal</td>
<td>o</td>
<td>0</td>
<td>0</td>
<td>0,1,2,3</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>hex</td>
<td>h</td>
<td>0</td>
<td>0</td>
<td>0,1,2,3</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>logical</td>
<td>l</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>note 3</td>
</tr>
<tr>
<td>word</td>
<td>w</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>ASCII (non-var)</td>
<td>a</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6,7,9</td>
</tr>
<tr>
<td>ASCII (non-var)</td>
<td>c</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6,8,9</td>
</tr>
<tr>
<td>ASCII (var)</td>
<td>v</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8,9</td>
</tr>
<tr>
<td>pointer</td>
<td>p</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>filler</td>
<td>x</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>new line</td>
<td>/</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>form feed</td>
<td>^</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>reposition</td>
<td>y</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>repeat group</td>
<td>$</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10,12</td>
</tr>
<tr>
<td>end repeat</td>
<td>\</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>terminate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>terminate w/newline</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- not applicable, usually ignored.
  o optional

Notes:

1 Integer precision values are:

0  fixed bin(16,0) unsigned; SHORT_CARDINAL
1  fixed bin(15,0) signed; SHORT_INTEGER
2  fixed bin(31,0) signed; LONG_INTEGER
3  fixed bin(32,0) unsigned; LONG_CARDINAL
z implies TRUE/FALSE as opposed to T/F
works on bit(16)aligned (FORTRAN LOGICAL)
same as :0zo
takes two arguments: char(*), fixed bin(15,0) (string, length)
strips trailing blanks
doesn't strip trailing blanks
default justification is left
fw is repeat count (default 1)
fw is argument number (default 1)
repeat groups cannot nest

Appendix E
References

[1] Burley, J. C.


BASIC/VM Programmer’s Guide.

[5] Desmond, Ellen S.
C User’s Guide.

CPL User’s Guide.

Data Backup and Recovery Guide.

DBMS Administrator’s Guide.

DBMS Data Description Language Reference Guide.

[10] Kingsbury, B. & Wilson, A. C.
DISCOVER Reference Guide.

DISCOVER User’s Guide.

Distributed Processing Terminal Executive Guide.

DSM User’s Guide.

[14] Shepp, Marion.
EMACS Reference Guide.
FORTRAN 77 Reference Guide.  

FED User's Guide.  

FORMS Programmer's Guide.  

[18] Lewis, Anthony.  
FORTRAN Reference Guide.  

Instruction Sets Guide.  

Interpretive BASIC User's Guide.  


[22] Prime.  

[23] Ladd, Anne P.  
SEG and LOAD Reference Guide.  

Magnetic Tape User's Guide.  

MIDAS/PLUS User's Guide.  

Modula-2 Reference Guide.  

[27] Shores, Andrew.  
Network Planning and Administration Guide.  

[28] Derr, Daniel.  
New User’s Guide to EDITOR and RUNOFF.  

[29] Shores, Andrew.  
NTS Planning and Configuration Guide.  
*OAS System Administrator's Guide.*

[31] Perry, Elizabeth Hanes.
*Operator's Guide to the Batch Subsystem.*


[33] Rose, Tom.

[34] Alley, Stephen.
*Operator's Guide to System Backup.*

[35] Forbes, J., Landy, A., Miles, C.

[36] Zegarra, Sonya.

[37] Hasse, Camilla B.
*Pascal Reference Guide.*

[38] Spector, D.
*The DEREMER Parser Generator.*

*SPL Reference Guide.*

[40] Rand, D.
*BUIL: a Tool for Program Building.*

[41] Ullmann, R.
*PDN Mailer User's Guide.*

[42] Xenakis, J. & Haase, C.
*PL/I Reference Guide.*

[43] Unknown.

*Assembly Language Programmer's Guide.*
*PRIMENET Guide.*  

[46] Shores, Andrew.  
*PRIMENET Planning and Configuration Guide.*  

*Prime User's Guide.*  

[48] Unknown.  
*PRIMEWORD Administrator's Guide.*  

*PRIMOS Commands Reference Guide.*  

*PRISAM User's Guide.*  

*Programmer's Guide to BIND and EPFs.*  

[52] Ryan, David.  

[53] Munro, Andrew.  
*ROAM Administrator's Guide.*  

[54] McKenzie, Charles D.  
*RPG II V-Mode Compiler Reference Guide.*  

[55] Ryan, David.  
*PRIME/SNA Administrator's Guide.*  

[56] Ryan, David.  
*PRIME/SNA Operator's Guide.*  

[57] Cioto, Paul.  
*Source Level Debugger User's Guide.*  

[58] Breithaupt, J.  
*Subroutines Reference Guide; Vol I.*  

[59] Breithaupt, J.  
*Subroutines Reference Guide; Vol II.*  
[60] Breithaupt, J.  
Subroutines Reference Guide; Vol III.  

[61] Breithaupt, J.  
Subroutines Reference Guide; Vol IV.  


[63] Conrad, Lois Anne.  

[64] Frost, Dick.  

[65] Hammond, M. & Landy, A.  
Index

$ 2-10

ABBREV 2-10
ABBRSW 4-3
Abort Flags 4-1
ABS H E 4-1
ACCESS command 8-4
Access Controls 3-26
ADD_REMOTE_ID 2-11
ADDISK 2-10
ADMIN_LOG 2-11
AIDS 2-11
Alarms 4-1
AMLC 2-12, 9-2
AMLC Process 3-19
AP 3-1
ARCHIVE 2-12
ARCHIVE_RELEASE 2-13
ARCHIVE_RESTORE 2-13
Argument Pointer 3-1
ARID 2-11
ASCII A-1
ASR Control Words 9-3
ASRCWD 2-14
ASSIGN 2-14
ATM 2-14
ATM_ADMIN 2-14
ATTACH 2-14
AUTOPSY 2-15
AVAIL 2-16
AWARMOFF command 8-4
AWARMON command 8-4
backup 2-16
BACKUP_RESTORE 2-17
BASIC 2-18
BASICV 2-18
BASINP 2-18
BATCH 2-18
BATGEN 2-18
BINARY 2-18
BIND 2-18
BOOT 8-2
BOOT command 8-5
BOOT_ATTACH 2-20
BOOT_CREATE 2-20
BOOT_IMP CODE 2-20
BOOT_RESTORE 2-20
BOOT_SAVE 2-20
BOOT_TREE 2-20
BOOTD command 8-5
BOOTP command 8-5
CBL 2-20
CBLDML 2-20
CBLSUBS 2-21
CC 2-21
CDML 2-21
CE-opt 2-96
CHANGE_PASSWORD 2-21
CHAP 2-22
Character Set A-1
Check header 3-2
Checks 3-1
Clock Process 3-19
CLOSE 2-22
CLUP 2-22
CMPPF 2-22
CN_RBF 2-22
CNAME 2-22
CNVTMA 2-22
COBOL 2-20, 2-22, 2-65
COMINPUT 2-23
Command Input 2-23
Command Output 2-23
Commands 2-10
Common LISP 2-50
COMOUTPUT 2-23
Compiler options 2-96
CONCAT 2-23
Concealed Stack 3-2
Condition Code 3-16
CONFIG 2-24
CONFIG_DSM 2-26
CONFIG_NET 2-26
CONFIG_NTS 2-27
CONFIG UM 2-27
Conversion Tables B-1
COPY 2-27
COPY command 8-5
COPY_DISK 2-27
COPY_RBF 2-27
CPL 2-29
CPPMC 2-28
CPU 3-1
CPW 2-21
CRASH_AUDIT 2-28
CREATE 2-28
CREATK 2-28
CRMPC 2-28
CRSER 2-28
CSUBS 2-28
DATE 2-28
DATE command 8-5
Date format 5-14
DBACP 2-28
DBASIC 2-29
DBG 2-29
DBUTL 2-33
DEFINE_GVAR 2-33
DELAY 2-33
DELETE 2-34
DELETE_RBF 2-34
DELETE_VAR 2-34
DELSSEG 2-34
DENOTE 2-34
DEREMER 2-34
Device Addresses 9-1
DIAG 2-34
DIRECTORY command 8-5
DISCOVER 2-34
DISCOVER_TCB 2-35
Disk 9-3
Disk Addresses 9-4
Disk Errors 9-5
Diskette 9-5
DISKS command 2-35
DISPLAY command 8-5
DISPLAY_LOG 2-35
DISPLAYO command 8-5
DISTRIBUTE_DSM 2-35
DLGEN 2-35
DMC 9-7
DMPU 2-36
DMQ 9-7
DMSTK 2-36
DMT 9-7
DMx 9-6
DG command 8-5
DPTCFG 2-36
DPTX 2-36
DPTXMTR 2-36
DROPDT 2-36
DSW 3-2
DSWSTAT 3-3
DTAR 3-11
DUMP command 8-5
DUMP_USER 2-36
DUMPSTACK 2-36

ECB 3-12
ECL 2-39
ED 2-37
EDAC 2-39
EDB 2-39
EDIT_ACCESS 2-39
EDIT_COMMAND_LINE 2-39
EDIT_EFU 2-40
EDIT_PROFILE 2-40
ELIGHTS 2-41
EMACS 2-41
Entry 2-54
Entry Control Block 3-12
EPF Commands 2-10
ESR 2-41
EVENT LOG 2-41
EXPAND_SEARCH_RULES 2-41
External Commands 2-10

F77 2-41
F77DML 2-42
F77SUBS 2-42
FADDR 3-12
FAP 2-42
FAU 2-42
Fault table entry 3-12
Faults 3-12
FCODE 3-12
FDL 2-42
FDMJ 2-42
FED 2-42
FETCH command 8-5
FIGCOM 4-3
File System 5-1
File system date 5-14
File types 5-13
FILL command 8-5
FILMEM 2-43
FILVER 2-43
FIND_RING_BREAK 2-43
FIX_DISK 2-43
FIXBAT 2-43
FIXRAT 2-43
Floating point 3-13
Floppy 9-5
FSUBS 2-45
FTGEN 2-45
FTN 2-45
FTOP 2-46
FTR 2-46
FUTIL 2-46
GENERATE_CATALOG 2-47
HALT command 8-5
HDXSTAT 2-47
HELP 2-47
HELP command 8-5
HISTORY 2-47
HISTORY command 8-6
HMAP 3-17
HPSD 2-47
I/O 9-1
ICE 2-48
IDBMS 2-47
Indirect Pointer 3-14
INFO 2-47
INFORM 2-47
Information 2-47
INIT 2-48
INITIALIZE_COMMAND_ENVIRONMENT 2-48
INPUT 2-48
Instruction formats 7-1
Instruction Set 7-1
Interlude (SVG) 4-11
Internal Commands 2-10
IP 3-14
IPC Process 3-19
IROAM 2-48
JOB 2-48
KBUILD 2-49
Keys 3-15
KIDDEL 2-49
LABEL 2-49
LATE 2-49
LD 2-49
LDMP 2-52
LDNET command 8-6
LE 2-52
LEM 2-50
LIGHTS command 8-6
LIGHTSC command 8-6
LISP 2-50
LIST_ACCESS 2-50
LIST_CATALOG 2-51
LIST_DISKS 2-52
LIST_DUMP 2-52
LIST_ENDPOINT 2-52
LIST_ENDPOINT 2-52
LIST_HEAD 2-52
LIST_LIBRARY_ENTRIES 2-53
LIST_LIMITS 2-53
LIST_LIBRARY 2-53
LIST_MINI_COMMANDS 2-54
LIST_PRIMENET_NODES 2-54
LIST_PRIMENET_PORTS 2-55
LIST_PRIORITY_ACCESS 2-55
LIST_PROCESS 2-55
LIST_QUOTA 2-55
LIST_RBF 2-55
LIST_REMOTE_ID 2-56
LIST_SEARCH_RULES 2-56
LIST_SEGMENT 2-56
LIST_SEMAPHORES 2-56
LIST_SYNC 2-56
LIST_TAPE 2-56
LIST_UNITS 2-57
LIST_USERS 2-57
LIST_VAR 2-57
LIST_VCS 2-57
LISTF 2-50
LISTING 2-50
LL 2-53
LENT 2-53
LMAP 3-17
LMC 2-54
LOAD 2-57
LOGIN 2-58
LOGOUT 2-59
LOGPRT 2-59
LON 2-59
LOOK 2-59
LPAC 2-55
LS 2-56
LSR 2-56
LWORD 2-12

Machine Checks 3-1
MAGNET 2-59
MAGRST 2-60
MAGSAV 2-60
Magnetape 9-6
Magnetape Commands 9-8
Magnetape status 9-10
MAIL 2-60, 2-61
MAKE 2-62
MAXSCH 2-63
MAXUSR 2-63
MCLUP 2-63
MDUMP 2-63
MED_SPOOL 2-63
MEDCONFIG 2-63
MEDUSA 2-63
MESSAGE 2-64
MIRROR_OFF 2-64
MIRROR_ON 2-64
MMAP 3-18
MO_FULL command 8-6
MO_MAP command 8-6
MO_RFABS command 8-6
MO_RFCS command 8-6
MO_RFH command 8-7
MO_RFL command 8-7
MO_RST command 8-7
MO_USER command 8-7
Models 3-15
MODULA 2-64
MODULA-2 2-64
MOFF 2-64
MON 2-64
MONITOR_NET 2-64
MONITOR_RING 2-64
MP2 Process 3-19
MPACK 2-65
MPC Process 3-19
MPLUSCLUP 2-65
MRGF 2-65
MTDENS 2-65
MTRESUME 2-65
NCOBOL 2-65
NET 2-65
NETCFG 2-66
NETLINK 2-66

Prime Restricted
NETLOG 2-66
NETLVL 2-66
NSED 2-66
NTS_ASSOCIATE 2-66
NTS_LINE 2-66
NTS_LIST_ASSOCIATE 2-66
NTS_UNASSOCIATE 2-66
NUMBER 2-66

OA_ADMIN 2-67
OA_TERM 2-67
OAS 2-67
Octal/Decimal B-1
OPEN 2-67
OPRPRI 2-67
OPTION-A 9-3
ORIGIN 2-67
OWLDSC 2-68

Page Maps 3-17
PASCAL 2-68
PASSWD 2-68
PASSWORD_DIRS 2-68
PCB 3-18
PCBs 3-19
PDEV 9-4
PDNMail 2-60
PHANTOM 2-68
PHYRST 2-68
PHYSAV 2-68
PIO 9-10
PL1 2-69
PLIG 2-69
PUB 2-69
PLOT 2-69
PLP 2-69
PM 2-69
PMA 2-69
POWER 2-70
Powers of Two C-1
PPA 3-19
PPB 3-19
PRERR 2-70
PRIMEAIDS 2-11
Primeword 2-95
PRIMIX 2-70
PRIMOS 2-70, 4-1
PRINT_KSR 2-70
PRINT_NETLOG 2-70
PRINT_SCS 2-71
PRINT_SECURITY_LOG 2-71
PRINT_SYSLOG 2-72
PRMPC 2-72
Process Control Block 3-18
Programmed I/O 9-10
PROP 2-72
PROTEC 2-73
PROTECT 2-73
PRSER 2-73
PRTDSC 2-73
PRVER 2-73
PSD 2-73
PSD20 2-75
PSLOG 2-71
PST100DSC 2-75
PT45DSC 2-76
PT46DSC 2-76
PTDSC 2-75
PE-T 500
Prime Engineering Handbook

PTELE 2-75
PTUSEG 4-4
PUDCOM 4-4
PWDIR 2-68

QCB 3-19
Quad floating point 3-13
Queue Control Block 3-19

RCP command 8-7
RDMP 2-77
RDY 2-76
Ready List 3-19
Record Headers 5-4
REFORM 2-76
Register File 3-20
Registers 3-20
RELEASE_LEVEL 2-76
REMEPF 2-76
REMOTE 2-76
REMOVE_EPF 2-76
REMOVE_PRIORITY_ACCESS 2-76
REMOVE_REMOTE_ID 2-76
RMPWD command 8-7
REN 2-77
REPLY 2-77
RESET_DUMP 2-77
RESTATE 2-77
RESTOR 2-77
RESTORE_RBF 2-77
RESUME 2-77
RESUS 2-77
REVERT_PASSWORD 2-78
RJ1004 2-78
RJ2200UT 2-78
RJ7020 2-78
RJORTS 2-78
RJHASP 2-78
RJOJ 2-78
RJQ 2-78
RJX80 2-78
RLS 2-76, 2-78
RO_TRACE_EVENTS 2-78
ROSUE 2-78
ROXTL 2-79
RPAC 2-76
RPG 2-79
RRID 2-76
RSAV format 3-24
RSTEM 2-79
RUN command 8-7
RUNOFF 2-79
RWLOCK 2-81

SAC 2-85
SAVE 2-81
Save mask 3-24
SAVE_RBF 2-81
SCHDEC 2-81
SCHED 2-81
SCHEMA 2-81
SD command 8-7
SDW 3-26
SECMON 2-82
SECST 2-82
SECURITY_MONITOR 2-82
SECURITY_STATUS 2-82
SEG 2-82
Segment Descriptor Word 3-26
Semaphore 3-26
SET_ACCESS 2-85
SET_ASYNC 2-85
SET_DELETE 2-85
SET_PRIORITY_ACCESS 2-85
SET_QUOTA 2-85
SET_RBF 2-85
SET_SEARCH_RULES 2-86
SET_TIME 2-86
SET_TIME_INFO 2-86
SET_VAR 2-86
SETIME 2-84
SETIME command 8-7
SETMOD 2-85
SHARE 2-86
SHUTDOWN 2-86
SIZE 2-86
SLIST 2-86
SMLC Process 3-19
SNA_3270 2-87
SNA_3270_CONFIG 2-87
SNA_PRINT 2-87
SNA_SERVER 2-88
SNA_SERVER_CONFIG 2-88
SOC 9-3
Software interrupts 4-9
SORT 2-88
SPAC 2-85
SPINDOWN command 8-7
SPL 2-88
SPOOL 2-89
SPSS 2-90
SPSSX 2-90
SPY 2-90
SQ 2-85
SSR 2-86
Stack 4-10
Stack Extension 3-27
Stack Frame 3-27
Stack Header 3-27
Stack Root 3-27
Stack, concealed 3-2
START 2-90
START_DSM 2-90
START_LSR 2-91
START_NET 2-91
START_NTS 2-91
STARTUP 2-91
STATUS 2-91
STATUS_DSM 2-91
STI 2-86
STLB 3-28
STOP_DSM 2-91
STOP_LSR 2-91
STOP_NET 2-91
STOP_NTS 2-91
Storage Module 9-6
SVC Interlude 4-11
SVCSET 2-92
SYSLOG 2-92
TA_ADMIN 2-92
TAP 2-92
TCF 2-92
TDOS64 2-92
TEMPLATE 2-92
TERM 2-92
TIME 2-92
TIMER 2-93
TLOG 2-93
TP 2-93
TP_EKO 2-93
TPBE 2-93
TPLINK 2-93
TRACE_RQ 2-93
TRAMLČ 2-93
TRANSFER_LOG 2-93
TRANSPORT 2-93
TRANSPORT_RELEASE 2-94
TRANSPORT_RESTORE 2-94
TSEALM 4-1
TYPE 2-94

UFD Entry 5-9
UFD Header 5-7
UI Requirements 2-58
UPLOAD 2-94
UNASSIGN 2-94
UPCASE 2-94
UPCOM 4-11
USAGE 2-95
User profile 4-11
USERS 2-95
USRASR 2-95

VERSATEC Process 3-19
VISTA 2-95
VPSS 2-95
VRPG 2-95
VRTSSW 2-95

Wildcards 2-1
WORD 2-95
WP_ADMIN 2-96
WPS 2-96
WS1004 2-96
WS200UT 2-96
WS7070 2-96
WSGRTS 2-96
WSHASP 2-96
WSX50 2-96

X.MAIL 2-61

Z80MA 2-96
Z8KMA 2-96