

MCAL REFERENCE LISTING

PROPRIETARY INFORMATION

This document contains information which is proprietary to and considered a trade secret of the ULTIMATE CORPORATION. It is expressly agreed that it shall not be reproduced in whole or part, disclosed, divulged, or otherwise made available to any third party either directly or indirectly. Reproduction of this document for any purpose is prohibited without the prior express written authorization of the ULTIMATE CORPORATION.

Copyright August, 1986, THE ULTIMATE CORP.
Operating System Release 10 - Rev 180

**CONFIDENTIAL MATERIAL
FOR INTERNAL COMPANY USE ONLY**

This document contains a listing of the MCALs (Monitor calls) that are in effect as of Release 10 Rev 180. The purpose of the document is to be a reference for Ultimate personnel who need to use and/or implement MCALs in either firmware or software implementations of the Ultimate operating system.

This document contains information which is strictly confidential to the Ultimate Corp. It should be read only by Ultimate Research and Development personnel who will be working with the Assembler and/or an Ultimate Kernel (Monitor), and not divulged outside the R & D Department.

MCALs on Firmware Implementations

Each MCAL is covered using the format in Figure A. The title of each MCAL contains its sequential hexadecimal number within class 11 (X'B'). The MCALs are presented in numerical order. MCALs have a synonym name, which is centered at the top of the page. All synonym forms of an MCAL are documented. Both the source and object codes are also given.

An index contains each MCAL synonym name in alphabetical order, and references the page on which it is discussed.

Understanding the MCAL Source and Object Formats

The general format of an MCAL source statement is:

MCAL Rr,nn,m

where:

Rr = the register number. In many MCALs this parameter is not used, but still must be specified (e.g., R0, R8). May be expressed in decimal (2, 15, etc.) or hex (X'2', X'F', etc.). The 'r' value (in hex) is assembled into the opcode's first byte, second nibble (e.g., 40, 48); the first nibble is the opcode identifier '4'.

nn = the sequential number of this MCAL within the class 'm' (e.g., 2, 14, 26). May be expressed in decimal (4, 14, 26, etc.) or hex (X'4', X'E', X'1A', etc.) The 'nn' value (in hex) is assembled into the opcode's second byte (e.g., 02, 0E, 1A).

m = the class number, which must be 11 (X'B'). The 'm' value is assembled into the opcode's third byte, second nibble (e.g., AB); the first nibble is the sub-opcode identifier, which is typically 'A', but may also be 8 or 9.*

* The Ultimate firmware interprets the following sub-opcodes to be MCALS: 8, 9, A, B, and F. The sub-opcodes B and F are reserved for VIQED and VIQ, respectively. An '8' is an MCAL which may be executed by the firmware rather than the kernel (Monitor). The 'A' is the preferred sub-opcode to use until all bits sequential 'nn' numbers have been implemented in MCALS (up to X'FF'). The '9' sub-opcode will be reserved for future use.

The Assembler assembles MCAL source statements into opcodes with the following general format (in hexadecimal):

```
4rnnsm
VVV
1 2 3 bytes
```

where:

- 4 = the opcode identifier
- r = register number 'n' from source statement (in hex)
- nn = the sequential number of this MCAL within 'm'
- s = the sub-opcode identifier.
- m = the class number 'm': X'B'.

Thus, an MCAL written as:

MCAL R7,4,11

would assemble into three bytes as follows:

```
-----
| 4 | 7 | 0 | 4 | A | B |
-----
```

MCALs on Software Implementations

The object code format discussed above and in subsequent sections does not apply to software implementations of the Ultimate instruction set, in which the object code may be whatever the implementors decide. However, because most machines are implemented in firmware and their opcodes are commonly known and used, this document still includes a complete discussion of the opcodes.

NOTE TO VIRTUAL PROGRAMMERS:

All MCALs have been assigned names. Writers of virtual code are hereby requested to use the named form. The numbered form may be used in the interim, while you are developing a name.

MCAL nn

(NAME)

MCAL Rr,nn,m opcode = 4rnnAm
MCAL Rr,nn,m (synonyms, if applicable)

PURPOSE:

statement

INPUT:

xxx.....

xxx.....

OUTPUT:

xxxxxx.....

DESCRIPTION:

additional explanation, if applicable

DATA STRUCTURES:

data structures modified, if applicable

Figure A. MCAL Model Format

MCAL REFERENCE LISTING

TABLE OF CONTENTS

Section and Topic

Page

Section 1: Sequential Listing of MCALS

MCAL 2 (MTBF)	1
MCAL 3 (LINK.CNT)	2
MCAL 4 (MTB)	3
MCAL 5 (CMD.STAT)	4
MCAL 6 (CMD.MAXFID)	5
MCAL 7 (CMD.FAKE.WT)	6
MCAL 9 (GET.ID)	7
MCAL C (TL.READ)	8
MCAL D (PANEL)	9
MCAL E (START.IO.PIB)	10
MCAL F (WARM.DUMP)	11
MCAL 10 (DB.ENT)	12
MCAL 11 (DB.LV)	13
MCAL 12 (PIB.AND)	14
MCAL 13 (PIB.OR)	15
MCAL 14 (FAKE.RD)	16
MCAL 15 (FAKE.WT)	17
MCAL 16 (WAIT)	18
MCAL 17 (QUERY)	20
MCAL 18 (PIB.PEEK)	22
MCAL 19 (PIB.POKE)	23
MCAL 1A (N.GET.ID)	24
MCAL 1C (ALARM.CLOCK)	25
MCAL 1D (CLOCK.CANCEL)	26
MCAL 1E (INT.CANCEL)	27
MCAL 1F (VMCAL)	28
MCAL 20 (FRM.UNLOCK)	29
MCAL 21 (FRM.LOCK)	30
MCAL 22 (SLEEP or SLEEP:)	31
MCAL 24 (DISK.ERR)	32
MCAL 25 (FORCE.WRITE)	33
MCAL 26 (SET.TIME)	34
MCAL 27 (TIME or GET.TIME)	35
MCAL 28 (RQM)	36
MCAL 29 (LOCK)	37
MCAL 2A (LOCK)	39
MCAL 2B (PIB.ATL)	41
MCAL 2C (DSABL.DSK)	42
MCAL 2D (QUEUE.READ)	43
MCAL 2F (RTC.CALIB)	44

MCAL REFERENCE LISTING

TABLE OF CONTENTS (cont.)

<u>Section and Topic</u>	<u>Page</u>
Section 1: Sequential Listing of MCALs (cont.)	
MCAL 30 (TEST.INP)	45
MCAL 31 (VOPT.OR)	46
MCAL 32 (VOPT.AND)	47
MCAL 33 (CLEAR.INP)	48
MCAL 34 (PERIPH.WRT.ONE)	49
MCAL 35 (PERIPH.RD.ONE)	50
MCAL 36 (CLR.OUT)	51
MCAL 37 (PIB.XPCB)	52
MCAL 38 (DISK.STAT)	53
MCAL 39 (WRITE.WAIT)	54
MCAL 3A (RCV.LEN)	55
MCAL 3D (SET.FL.DEN)	56
MCAL 3E (XFER.CLOCK)	57
MCAL 3F (SET.BATCH.TM)	58
MCAL 40 (PERIPH.RD)	59
MCAL 41 (PERIPH.WRT)	60
MCAL 44 (VMS.SPOOL)	61
MCAL 45 (VMS.TAPE)	62
MCAL 46 (VMS.OFF)	63
MCAL 47 (VMS.MSG)	64
MCAL 48 (PC.MSG)	66
MCAL 49 (FAKE.READ)	67
MCAL 4A (RFLAGS.CLR)	68
MCAL 4B (FLAGS.SET)	69

INDEX

MCAL 2

(MTBF)

MCAL R0,2,11

Opcode = 24002AB

PURPOSE:

Moves a buffer to the bottom of the age links.

INPUT:

T0 = Buffer number

OUTPUT:

none

DESCRIPTION:

This MCAL makes the specified buffer the first one to be used to satisfy frame faults or fake reads.

DATA STRUCTURES:

Age links are modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 3

(LINK.CNT)

MCAL R0,3,11

opcode = 4003AB

PURPOSE:

Counts forward and backward age links.

INPUT:

none

OUTPUT:

T0 = # of buffers in age links, counting in the forward direction.

T1 = # of buffers in age links, counting in the backward direction.

DESCRIPTION:

This may be used by virtual programs to test the integrity of the age links. The count should be the same from each direction. The count is the number of buffers available for paging.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 4

(MTB)

MCAL Rr,4,11

opcode = 4r04AB

PURPOSE:

Moves a frame (FID) to bottom of age links.

INPUT:

Rr = points to the buffer to move.

OUTPUT:

none

DESCRIPTION:

This MCAL makes the buffer the first one to be used to satisfy frame faults or fake reads.

DATA STRUCTURES:

Age links are modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 5

(CMD.STAT)

MCAL R0,5,11

opcode = 4005AB

PURPOSE:

Gets the copy protect and attention bits from last status to PCB.

INPUT:

none

OUTPUT:

D0

DESCRIPTION:

First, T0 and T1 are zeroed. If there is a CMD disk acting as a pseudo tape (applies to Honeywell systems only), then:

- the high bit of T1 is set.
- if the pack has an Ultimate label, the high bit of T0 is set.
- the write-protect status bit is moved to position X'2000' of T0.
- the attention status bit is moved to position X'4000' of T1.

On a VAX or DEC machine, this is an illegal opcode.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 6

(CMD.MAXFID)

MCAL R0,6,11

opcode = 4006AB

PURPOSE:

Returns pseudo-maxfid (PMAXFID).

INPUT:

none

OUTPUT:

D0 (contains PMAXFID without leading bit)

DESCRIPTION:

If there is a disk acting as a pseudo tape drive, the highest FID that can be written to the pack (PMAXFID) is returned in D0. Otherwise, '0' is returned.

Presently, the systems supporting the pseudo tape feature are:

Honeywell - CMD disk

PC - floppy

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 7

(CMD.FAKE.WT)

MCAL R0,7,11

opcode = 4007AB

PURPOSE:

Does a fake write (zeroes out the write required flag) on all pseudo-FIDs in buffer table.

INPUT:

none

OUTPUT:

none

DESCRIPTION:

Any FIDs that are pseudo-tape FIDs are fake-written. Pseudo-FIDs range from X'800000' to X'8 PMAXFID'.

Presently, the systems supporting the pseudo tape feature are:

Honeywell - CMD disk

PC - floppy

DATA STRUCTURES:

Each frame that is fake written is moved to the bottom of the age links.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 9

(GET.ID)

MCAL R0,9,11

opcode = 4009AB

PURPOSE:

Gets device-id for device number in T0.

INPUT:

T0 = Virtual device number (used as index into Virtual Device Table) in Honeywell form. That is, there is one number for the input function of a device and another number for the output function.

OUTPUT:

T0 = 0 or a device ID number. If 0, no device is configured on that channel. Otherwise, the device ID number is returned.

T1 = Channel address

DESCRIPTION:

If no device exists for that virtual device number, then T0 is zeroed. On the Honeywell, one of the channel address bits shows whether the channel is for reception or transmission. Virtual code uses this bit to select only one of the pair of entries associated with a given device.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL C

(TL.READ)

MCAL Rr,X'C',11
MCAL Rr,12,11

opcode = 4r0CAB

PURPOSE:

Transaction logger special READ.

INPUT:

Rr = specifies the address of the byte (tally) to clear.

OUTPUT:

The byte (tally)* pointed to by the register is zeroed.

DESCRIPTION:

The Transaction logger uses this. If any characters are in the terminal input buffer (typeahead buffer), the virtual process is allowed to resume. Otherwise, the process is detached and removed from the SNU links.

DATA STRUCTURES:

PIB may be removed from SNU links.

BT entry for frame pointed to by RA is marked write-required.

- * Ultimate software before Rev 180 supported no more than 255 processes, so process numbers (PIB numbers) could fit in one byte. Beginning with Rev 180, the number of processes supported is being increased. The process number must now be a tally.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL D

(PANEL)

MCAL R0,X'D',11
MCAL R0,13,11

opcode = 400DAB

PURPOSE:

Invokes panel (remote panel processor = :PANEL
verb).

INPUT:

T0 = Port number to use.

OUTPUT:

none

DESCRIPTION:

On Honeywell systems, this starts the PANEL
debugger program running on the port specified by
T0.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL E

(START.IO.PIB)

MCAL R0,X'E',11
MCAL R0,14,11

opcode = 400EAB

PURPOSE:

Starts MLCP I/O on the line (PIB port).

INPUT:

T0 = line number of PIB port to start I/O on.

OUTPUT:

none

DESCRIPTION:

This initializes and starts input on the terminal attached to the specified PIB.

DATA STRUCTURES:

Terminal input buffer initialized.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL F

(WARM.DUMP)

MCAL R0,X'F', 11
MCAL R0,15,11

opcode = 400FAB

PURPOSE:

Warmstarts, and dumps to tape.

INPUT:

T0 = Code for desired action. Valid codes are:

X'F511'
X'DEAD'
X'DC10'

OUTPUT:

none

DESCRIPTION:

If T0 = X'F511', then memory is flushed to disk and the system is warmstarted (a :WARMSTART).

If T0 = X'DEAD', then memory is written to tape and the system halts.

If T0 = X'DC10', then memory is flushed to disk and the system halts (a :WARMSTOP).

DATA STRUCTURES:

As part of flushing memory to disk:

- buffer table is rebuilt
- age links initialized and rebuilt
- all disk I/O data structures

If the system is restarted, all data structures are reinitialized, including those mentioned above.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 10

(DB.ENT)

MCAL R0,X'10',11
MCAL R0,16,11

opcode = 4010AB

PURPOSE:

Enter software debugger.

INPUT:

none

OUTPUT:

The debug bit (X'0080') in PIB word zero is set.

The debug bit (X'0080') in the PCB ACF field is set.

DESCRIPTION:

The two debug bits are set and the process is detached. When the process is next activated, the firmware will use the DCB instead of the PCB.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 11

(DB.LV)

MCAL Rr,X'11',11
MCAL Rr,17,11

opcode = 4r11A3

PURPOSE:

Leave software debugger.

INPUT:

Rr - byte zero of the PCB

OUTPUT:

The debug bit (X'0080') in PIB word zero is cleared.

The debug bit (X'0080') in the PCB ACF field is cleared.

DESCRIPTION:

The two debug bits are cleared and the process is detached. When the process is next activated, the firmware will use the PCB instead of the DCB.

DATA STRUCTURES:

BT entry for Rr (PCB) is set write-required.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 12

(PIB.AND)

MCAL R0,X'12',11
MCAL R0,18,11

opcode = 4012AB

PURPOSE:

'AND' bits in PIB word zero.

INPUT:

T0 = Tally with mask containing bits to be ANDed with word 0 of the PIB.

T1 = PIB line number. If negative, then use caller's own PIB.

OUTPUT:

T0 = Contains resultant PIB word 0.

DESCRIPTION:

This is typically used to clear roadblocks.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 13

(PIB.OR)

MCAL R0,X'13',11
MCAL R0,19,11

opcode = 4013AB

PURPOSE:

'OR' bits in PIB.

INPUT:

T0 = Tally with mask containing bits to be ORed with
word 0 of the PIB.

T1 = PIB line number. If negative, then use caller's
own PIB.

OUTPUT:

T0 = Contains resultant PIB word 0.

DESCRIPTION:

This is typically used to set roadblocks.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 14

(FAKE.RD)

MCAL R0,X'14',11
MCAL R0,20,11

opcode = 4014AB

PURPOSE:

Fake disk read (as if R15 has been frame faulted).
Assign the buffer; do not do the read.

INPUT:

R15FID = the frame number to be fake-read.

OUTPUT:

none

DESCRIPTION:

The memory map is modified so that the referenced frame is in memory, but it is not read from disk. It is assumed that the frame will be initialized after this.

DATA STRUCTURES:

Modifies these data structures:

- buffer table
- age links
- disk I/O queues (on some systems)

CAUTION: This monitor call is obsolete. MCAL 49 should be used instead. Some software machines force all registers to be attached. Therefore, R15 would be attached, and the frame would be read before the MCAL is executed by the kernel.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 15

(FAKE.WT)

MCAL R0,X'15'11
MCAL R0,21,11

opcode = 4015AB

PURPOSE:

Fake write. Zero the buffer table write required bit of the buffer pointed to by R15.

INPUT:

R15FID = the frame number to be fake-written.

OUTPUT:

none

DESCRIPTION:

If the specified frame is in memory, it is marked as not write-required and its buffer is made the first buffer to be used to satisfy frame faults and fake reads.

DATA STRUCTURES:

Modifies these data structures:

- buffer table
- age links

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 16

(WAIT)

MCAL R0,X'16',11
MCAL R0,22,11

opcode = 4016AB

PURPOSE:

Suspend process until virtual interrupt.

INPUT:

T0 = Virtual device number.

OUTPUT:

T0 = Device number (if 0 or greater). Otherwise:
-1 (no entry for this device)
-2 (clock timeout)
-3 (semaphore timeout)

T1 = If a virtual device interrupted, this is the number of interrupts received. Otherwise, zero (0).

DESCRIPTION:

If any interrupts have been received from any device that had an I/O operation started by this process:

- the virtual device number of the interrupting device is placed in T0.
- the number of interrupts received is placed in T1.
- if the device has any I/Os still outstanding, the count of outstanding I/Os is reduced by the number received; otherwise, the device is marked inactive.
- the virtual process is resumed.

NOTE: If the process has interrupts outstanding for multiple devices, only the interrupts for one device will be returned and there is no priority ordering of devices.

If no device interrupts have been received:

- if an alarm clock that the process had armed has expired, then -2 is returned in T0, the clock request block is marked inactive, and the process is resumed.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

- if an alarm clock that the process had armed was cancelled because another process attempted to activate this process, -3 is returned in T0, the clock request block is marked inactive, and the process is resumed.
- if no alarm clock had expired or was cancelled, and there were no outstanding I/O operations, then -1 is returned in T0 and the process is detached.
- If no alarm clock had expired or was cancelled and there is at least one outstanding I/O operation, and no interrupts have been received (a -4 condition in MCAL QUERY), the process is removed from the SNU links. In addition, the program counter is backed up to the WAIT MCAL, and the process is detached. This prevents activation of the process until an interrupt occurs.

DATA STRUCTURES:

Modifies these data structures:

- SNU links
- Virtual device table
- Clock request block and links

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 17

(QUERY)

MCAL R0,X'17',11
MCAL R0,23,11

opcode = 4017AB

PURPOSE:

Query or Query virtual interrupt. Same as MCAL 16 (WAIT) except never suspends process.

INPUT:

none

OUTPUT:

T0 = Device number (if 0 or greater). Otherwise:
-1 (no entry for this device)
-2 (clock timeout)
-3 (semaphore timeout)
-4 (no clock timeout and outstanding I/O)

T1 = If a virtual device interrupted, this is the number of interrupts received. Otherwise, zero (0).

DESCRIPTION:

If any interrupts have been received from any device that had an I/O operation started by this process:

- the virtual device number of the interrupting device is placed in T0.
- the number of interrupts received is placed in T1.
- if the device has any I/Os still outstanding, the count of outstanding I/Os is reduced by the number received; otherwise, the device is marked inactive.
- the virtual process is resumed.

NOTE: If the process has interrupts outstanding for multiple devices, only the interrupts for one device will be returned and there is no priority ordering of devices.

If no device interrupts have been received:

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

- if an alarm clock that the process had armed has expired, then -2 is returned in T0, the clock request block is marked inactive, and the process is resumed.
- if an alarm clock that the process had armed was cancelled because another process attempted to activate this process, -3 is returned in T0, the clock request block is marked inactive, and the process is resumed.
- if no alarm clock has expired or was cancelled, and there was at least one outstanding I/O operation, then -4 is returned in T0 and the process is detached.
- if no alarm clock had expired or was cancelled, and there were no outstanding I/O operations, then -1 is returned in T0 and the process is detached.

DATA STRUCTURES:

Modifies these data structures:

- SNU links
- Virtual device table
- Clock request block and links

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 18

(PIB.PEEK)

MCAL R0,X'18',11
MCAL R0,24,11

opcode = 4018AB

PURPOSE:

Returns the value of a specified word in a PIB.

INPUT:

T1*= PIB number. Negative means self.

H4*= PIB word number.

OUTPUT:

T0 = value of the specified word.

DESCRIPTION:

Note that virtual programmers should be aware that the kernel or firmware may change the values of certain PIB words at any time.

DATA STRUCTURES:

No data structures modified.

- * Ultimate software before Rev 180 supported no more than 255 processes, so process numbers (PIB numbers) could fit in one byte. Beginning with Rev 180, the number of processes supported is being increased. The process number must now be a tally.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 19

(PIB.POKE)

MCAL R0,X'19',11
MCAL R0,25,11

opcode = 4019AB

PURPOSE:

Replaces a specified word in a PIB.

INPUT:

T1*= PIB number. Negative means self.

H4*= PIB word number.

T0 = new value to poke into the PIB.

OUTPUT:

none

DESCRIPTION:

Note: Virtual programmers should be aware that the kernel or firmware may change the values of certain PIB words at any time. Care must be taken to not damage the state of the operating system by inappropriate use of PIB.POKE.

DATA STRUCTURES:

Depends on what word is modified.

- * Ultimate software before Rev 180 supported no more than 255 processes, so process numbers (PIB numbers) could fit in one byte. Beginning with Rev 180, the number of processes supported is being increased. The process number must now be a tally.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 1A

(N.GET.ID)

MCAL R0,X'1A',11
MCAL R0,26,11

opcode = 401AAB

PURPOSE:

Gets the device ID.

INPUT:

T3 = virtual device number in Honeywell format (same as
GET.ID).

OUTPUT:

T0 = device ID or zero if device number is too big

T1 = channel address

T2 = buffer number, in a format appropriate for VIOLD
instructions

DESCRIPTION:

Gets device ID.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 1C

(ALARM.CLOCK)

MCAL R0,X'1C',11
MCAL R0,28,11

opcode = 401CAB

PURPOSE:

Enables alarm clock request for specified time.

INPUT:

D0 = amount of time until expiration, in milliseconds

OUTPUT:

none

DESCRIPTION:

A timer is initialized and started. This may later be used by the WAIT and QUERY MCALs.

DATA STRUCTURES:

Modifies clock request block and links.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 1D

(CLOCK.CANCEL)

MCAL R0,X'1D',11
MCAL R0,29,11

opcode = 401DAB

PURPOSE:

Resets an alarm clock request.

INPUT:

none

OUTPUT:

none

DESCRIPTION:

A previously set timer, if present, is cleared.

DATA STRUCTURES:

Modifies clock request block and links.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 1E

(INT.CANCEL)

MCAL R0,X'1E',11
MCAL R0,30,11

opcode = 401EAB

PURPOSE:

Resets virtual interrupt request.

INPUT:

T0 = virtual device number (Honeywell format) or, if
X'FFFF', all virtual devices for the process.

OUTPUT:

none

DESCRIPTION:

The interrupts from any I/O operations that the process had started (on the specified device or on any device) are cancelled. Interrupts from the device(s) may still come to the kernel, but the virtual device table is marked in a way that will cause them to be ignored.

DATA STRUCTURES:

Modifies the virtual device table.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 1F

(VMCAL)

MCAL Rr,X'1F',11
MCAL Rr,31,11

opcode = 4r1FAB

PURPOSE:

Kernel subroutine monitor call.

INPUT:

Rr - address of kernel code

OUTPUT:

none

DESCRIPTION:

This is used to execute kernel code (native CPU instructions) that resides in a virtual frame. This is presently only supported on Honeywell machines and is intended to be used for special purpose patches.

DATA STRUCTURES:

Modification depends on the kernel code.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 20

(FRM.UNLOCK)

MCAL Rr,X'20',11
MCAL Rr,32,11

opcode = 4r20AB

PURPOSE:

Unlocks designated frame.

INPUT:

Rr - address of any byte within the frame

OUTPUT:

none

DESCRIPTION:

None needed.

DATA STRUCTURES:

Modifies buffer table - the corelock bit is cleared.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 21

(FRM.LOCK)

MCAL Rr,X'21',11
MCAL Rr,33,11

opcode = 4r21AB

PURPOSE:

Locks designated frame in memory.

INPUT:

Rr - address of any byte within the frame

OUTPUT:

H4 = High byte of 24-bit byte address of byte 0 of the frame

H5 = Middle byte of 24-bit byte address of byte 0 of the frame

DESCRIPTION:

This locks the frame in memory and returns the memory address (in a "funny format") where it was locked. This may later be used by the VM or MV instructions. The frame will stay in that memory location until either:

- the FRM.UNLOCK MCAL is used
- the system is restarted

If the frame is modified by virtual software, the modification will be reflected on disk.

DATA STRUCTURES:

Modifies the buffer table.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 22

(SLEEP or SLEEP:)

MCAL Rr,X'22',11
MCAL Rr,34,11

opcode = 4r22AB

PURPOSE:

Puts process to sleep for specified time.

INPUT:

D0 = Time to wake up (in milliseconds after midnight)

Rr = Address of byte (tally) to clear

OUTPUT:

The byte (tally)* at Rr is zeroed.

DESCRIPTION:

The tally pointed to by Rr is zeroed. This is in case the spooler, which must clear a lock in synchronization with being deactivated, is executing the sleep. Ordinarily, the register is set to a scratch tally. If the SLEEP opcode is used, the register defaults to R0. The first tally of the PCB is scratch. To specify another register, use the SLEEP: opcode.

The process will be deactivated until one of these occurs:

- the wakeup time is reached.
- another process wakes it up, either by the PIB.AND (not recommended) or PIB.ATL MCAL, or by trying to set a lock that the process has set.
- the break key is pressed.

DATA STRUCTURES:

Modifies the clock request block and links.

- * Ultimate software before Rev 180 supported no more than 255 processes, so process numbers (PIB numbers) could fit in one byte. Beginning with Rev 130, the number of processes supported is being increased. The process number must now be a tally.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 24

(DISK.ERR)

MCAL Rr,X'24',11
MCAL Rr,36,11

opcode = 4r24AB

PURPOSE:

Reports disk error from 'stack' to virtual process.

INPUT:

Rr - address where disk error information should be put.
This must be at least 32 bytes before the end of a
frame.

OUTPUT:

32 bytes of disk error information are copied to
Rr.

DESCRIPTION:

This copies the kernel's disk error table to a
virtual frame and clears the table. If the table
was empty, zeroes are copied to the virtual frame.

DATA STRUCTURES:

Modifies the buffer table (write-required bit set).

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 25

(FORCE.WRITE)

MCAL Rr,X'25',11
MCAL Rr,37,11

opcode = 4r25AB

PURPOSE:

Forces write of designated frame by enqueueing.

INPUT:

Rr = address of frame to write

OUTPUT:

none

DESCRIPTION:

The purpose of this MCAL is to checkpoint a particular frame.

If the frame is not write-required, this is a NOP.

If the frame is being written, the program counter is backed up to the beginning of the MCAL and an RQM is executed. This makes the MCAL wait for previously started writes to complete.

Otherwise, the frame is scheduled to be written to disk.

DATA STRUCTURES:

Modifies disk queues.

CAUTION:

Not all kernels support multiple force-writes being active concurrently, due to a limited number of disk queue entries.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 26

(SET.TIME)

MCAL R0,X'26',11
MCAL R0,38,11

opcode = 4026AB

PURPOSE:

Sets the system time and date.

INPUT:

D0 = New system time, in milliseconds since midnight

T2 = New system date

OUTPUT:

none

DESCRIPTION:

Note: This is a NOP on the VAX because the timekeeping is done by VMS.

Opcode 40FFAA is an obsolete synonym. Support for this opcode will be withdrawn in the future.

DATA STRUCTURES:

No data structures are modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 27

(TIME)
(GET.TIME)

MCAL R0,X'27',11
MCAL R0,39,11

opcode = 4027AB

PURPOSE:

Gets the system time and date.

INPUT:

none

OUTPUT:

T2 = system date

D0 = system time in milliseconds since midnight

DESCRIPTION:

Opcode 4000AA is an obsolete synonym. Support for this opcode will be withdrawn in the future.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 28

(RQM)

MCAL R0,X'28',11
MCAL R0,40,11

opcode = 4028AB

PURPOSE:

Releases time quantum.

INPUT:

none

OUTPUT:

none

DESCRIPTION:

This deactivates a process for approximately 100 milliseconds. Any alarm clock that was set is not disturbed.

Opcode 4000A9 is an obsolete synonym. Support for this opcode will be withdrawn in the future.

DATA STRUCTURES:

Modifies clock request block and links.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 29

(LOCK)

opcode = 4r298D

PURPOSE:

Locks a system resource, with an ELSE clause.

INPUT:

Rr = address of a tally to be used as a lock.

OUTPUT:

Tally at Rr may contain process number + 1, with the bytes in swapped order.

INHIBITH is incremented if the resource is obtained or already owned by the process.

DESCRIPTION:

This is used to try to lock a system resource. If the resource is already locked by another process, a branch instruction that is assembled immediately following the MCAL is taken.

In assembly language, this is coded as:

LOCK REGISTER, LABEL

The assembler uses the label to construct the branch instruction, and to know what to put in byte 2 of the opcode.

The tally pointed to by the register is used as a lock. Zero is the unlocked condition. When the tally is locked, it contains the PIB number plus one of the process that set the lock (owns the resource). The PIB number + 1 is stored with the high and low order bytes reversed so that the DEC machines run more efficiently:

```

Rr -->  -----
        | low order byte | high order byte |
        | of PIB # + 1  | of PIB # + 1  |
        -----

```

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

The firmware in the DEC machines enables the kernel to efficiently process the first byte pointed to by Rr, but not the second byte. On the DEC machines (or any machines that cannot support more than 254 processes) the high order byte will always be zero, and the lock may be treated as a byte by the kernel. Virtual software should always consider the lock to be a tally whenever it reads it or initializes it.

If the tally contains a zero, the PIB number + 1 is put into the tally in byte-swapped order and execution resumes after the branch instruction that follows the MCAL.

If the tally is not zero, the lock is owned by the PIB represented by the tally. If this process already owned the lock, execution is the same as if the tally were zero. If the PIB owning the lock is roadblocked by disk, terminal I/O, or a trap, or it is active in the other processor of a dual processor system, the LOCK MCAL is treated as if it were an RQM. In this case, the virtual program counter is left pointing to the branch instruction, so the ELSE clause will be taken after the RQM. If the PIB owning the lock is able to be activated, an attempt is made to activate it by doing what the PIB.ATL MCAL does. The virtual program counter is left pointing at the branch instruction that follows the MCAL.

NOTE: Before release 180, the LOCK instruction generated opcode X'4r01AD' and the lock was a byte, not a tally. Opcode X'4r29AB' was defined as a synonym in the kernels, but was not in OSYM.

DATA STRUCTURES:

Modifies the following data structures:

- clock request block and links
- SNU links
- buffer table write required flag is set

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 2A

(LOCK)

opcode = 4r2A8D

PURPOSE:

Locks a system resource.

INPUT:

Rr = address of tally to be used as a lock

OUTPUT:

Tally at Rr will contain process number + 1, with the bytes in swapped order.

INHIBITH is incremented if the resource is obtained or already owned by the process.

DESCRIPTION:

This is used to try to lock a system resource. If the resource is already locked by another process, a branch instruction that is assembled immediately following the MCAL is taken.

In assembly language, this is coded as:

LOCK REGISTER

The lack of a label (vs. MCAL 29) causes the assembler to put the correct code in byte 2 of the opcode.

The tally pointed to by the register is used as a lock. Zero is the unlocked condition. When the tally is locked, it contains the PIB number plus one of the process that set the lock (owns the resource). The PIB number + 1 is stored with the high and low order bytes reversed so that the DEC machines run more efficiently:

Rr -->	low order byte	high order byte
	of PIB # + 1	of PIB # + 1

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

The firmware in the DEC machines enables the kernel to efficiently process the first byte pointed to by Rr, but not the second byte. On the DEC machines (or any machines that cannot support more than 254 processes) the high order byte will always be zero, and the lock may be treated as a byte by the kernel. Virtual software should always consider the lock to be a tally whenever it reads it or initializes it.

If the tally contains a zero, the PIB number + 1 is put into the tally in byte-swapped order and execution resumes with the instruction after the MCAL.

If the tally is not zero, the lock is owned by the PIB represented by the tally. If this process already owned the lock, execution is the same as if the tally were zero. If the PIB owning the lock is roadblocked by disk, terminal I/O, or a trap, or it is active in the other processor of a dual processor system, the LOCK MCAL is treated as if it were an RQM. However, the virtual program counter is backed up to the beginning of the MCAL. If the PIB owning the lock is able to be activated, an attempt is made to activate it by doing what the PIB.ATL MCAL does. The virtual program counter is backed up to the beginning of the MCAL instruction.

NOTE: Before release 180, the LOCK instruction generated opcode X'4r00AD' and the lock was a byte, not a tally. Opcode X'4r2AAB' was defined as a synonym in the kernels but was not in OSYM.

DATA STRUCTURES:

Modifies the following data structures:

- clock request block and links
- SNU links
- buffer table write required flag is set

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 2B

(PIB.ATL)

MCAL R0,X'2B',11
MCAL R0,43,11

opcode = 402BAB

PURPOSE:

Activate a process (add to top of PIB links).

INPUT:

T0 = PIB number to be activated

OUTPUT:

none

DESCRIPTION:

The SLEEP roadblock is cleared. If the CRB was in the links (meaning an alarm clock or sleep or RQM had not yet expired), the CRB is removed from the links and the semaphore flag is set (this causes a -3 to be returned by WAIT or QUERY). The PIB is moved to the top of the SNU links. The current process is detached and the target PIB is activated if there are no roadblocks.

The R0,00,15 form is an obsolete synonym. Support for it will be withdrawn in the future.

DATA STRUCTURES:

Modifies clock request block and links, and SNU links.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 2C

(DSABL.DSK)

MCAL R0,X'2C',11
MCAL R0,44,11

opcode = 402CAB

PURPOSE:

Disables disk set.

INPUT:

none

OUTPUT:

none

DESCRIPTION:

This is only used by the Honeywell offline monitor.
It switches the interrupt level of each disk from
the disk level to the virtual device level.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 2D

(QUEUE.READ)

MCAL R0,X'2D',11
MCAL R0,45,11

opcode = 402DAB

PURPOSE:

Queues a read (frame fault).

INPUT:

D0 = frame to read

OUTPUT:

none

DESCRIPTION:

If the frame is in memory, this is a NOP.
Otherwise, a disk read is started, but the PIB is
not marked disk roadblocked.

DATA STRUCTURES:

Modifies disk queues.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 2F

(RTC.CALIB)

MCAL R0,X'2F',11
MCAL R0,47,11

opcode = 402FAB

PURPOSE:

Changes the RTC tick count from 6 to 5 to accommodate systems that operate on 50 hertz power.

INPUT:

none

OUTPUT:

none

DESCRIPTION:

This is used only on DEC-based systems. Changes the real time (time of day) clock refresh value in the kernel to 5. Five ticks equal 100 ms.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 30

(TEST.INP)

MCAL R0,X'30',11
MCAL R0,48,11

opcode = 4030AB

PURPOSE:

Tests for characters in terminal input buffer.

INPUT:

T1 = PIB number

OUTPUT:

T0 = number of characters in input buffer

DESCRIPTION:

This may send an XON if the buffer is empty and
typeahead is enabled.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 31

(VOPT.OR)

MCAL R0,X'31',11
MCAL R0,49,11

opcode = 4031AB

PURPOSE:

Virtual option flag 'OR'.

INPUT:

T0 = mask to OR with options

OUTPUT:

T0 = new options value

DESCRIPTION:

This is used to change certain global system parameters.

DATA STRUCTURES:

No data structures modified.

MCAL 32

(VOPT.AND)

MCAL R0,X'32',11
MCAL R0,50,11

opcode = 4032AB

PURPOSE:

Virtual option flag 'AND'.

INPUT:

T0 = mask to AND with options

OUTPUT:

T0 = new options value

DESCRIPTION:

This is used to change certain global system parameters.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 33

(CLEAR.INP)

MCAL R0,X'33',11
MCAL R0,51,11

opcode = 4033AB

PURPOSE:

Resets (clears) terminal input buffer.

INPUT:

none

OUTPUT:

none

DESCRIPTION:

None needed.

DATA STRUCTURES:

Modifies terminal input buffer pointers.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 34

(PERIPH.WRT.ONE)

MCAL Rr,X'34',11
MCAL Rr,52,11

opcode = 4r34AB

PURPOSE:

Writes data byte to another line's port.

INPUT:

T0 = PIB number

Rr = address of the byte to write

OUTPUT:

none

DESCRIPTION:

If T0 refers to a PIB other than the one making the monitor call, the TRAP roadblock is set.

The specified byte is output as if the specified process had executed a WRITE instruction. Control returns immediately (that is, there is no wait for the output to complete). If the byte could not be written, the program counter is backed up to the MCAL instruction and an RQM is done; this causes the process to wait about 100 ms. before re-executing the MCAL.

DATA STRUCTURES:

Modifies the following data structures:

- terminal output buffer
- clock request block and links

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 35

(PERIPH.RD.ONE)

MCAL Rr,X'35',11

opcode = 4r35AB

MCAL Rr,53,11

PURPOSE:

Reads data byte from another line's port.

INPUT:

T0 = PIB number

Rr = address to copy byte to

OUTPUT:

Byte at Rr copied from designated input buffer.

DESCRIPTION:

This removes one byte from the designated PIB's terminal input buffer and copies it to virtual space. If T0 refers to a PIB other than the one making the monitor call, the TRAP roadblock is set.

If the PIB's terminal input buffer is not empty, the first byte in it is removed and copied to where Rr points.

If the input buffer is empty, the program counter is set back to the MCAL instruction and an RQM is done; this causes the process to wait for about 100 ms. before re-executing the MCAL.

DATA STRUCTURES:

Modifies the following data structures:

- terminal input buffer
- clock request block and links

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 36

(CLR.OUT)

MCAL R0,X'36',11
MCAL R0,54,11

opcode = 4036AB

PURPOSE:

Clears terminal output.

INPUT:

T1 = PIB number (negative means self)

OUTPUT:

none

DESCRIPTION:

This cancels any terminal output and clears the output roadblock.

DATA STRUCTURES:

Modifies the terminal output buffer.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 37

(PIB.XPCB)

MCAL Rr,X'37',11
MCAL Rr,55,11

opcode = 4r37AB

PURPOSE:

Changes PCB FID in PIB.

INPUT:

H2:H1:H0 = the new PCB FID

H3 = flags

Rr = some address in the new PCB

OUTPUT:

none

DESCRIPTION:

Sets bit in position X'20' of ACF of current PCB. Moves B30 of old PCB (bit position X'40' of H3) to PIB debug bit. Moves new PCB FID from the accumulator to the PIB. Zeroes bit in position X'20' of ACF of new PCB (for warmstart). The process is deactivated. The next time it is activated, the new PCB will be used.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 38

(DISK.STAT)

MCAL Rr,X'38',11
MCAL Rr,56,11

opcode = 4r38AB

PURPOSE:

Reports disk I/O statistics.

INPUT:

Rr = address of buffer to copy statistics to

OUTPUT:

none

DESCRIPTION:

Copies disk statistical counts to Rr.

****More to be supplied****

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 39

(WRITE.WAIT)

MCAL R0,X'39',11
MCAL R0,57,11

opcode = 4039AB

PURPOSE:

Wait for a frame to be written to disk.

INPUT:

R15FID = frame being written

OUTPUT:

none

DESCRIPTION:

This is used to synchronize writes to disk.

If the frame has been written to disk since the last FORCE.WRITE MCAL for the frame, this is a NOP. (The frame not being in memory or not being write-required satisfies this condition.) Otherwise, the program counter is backed up to the beginning of the MCAL and an RQM is performed.

DATA STRUCTURES:

Modifies clock request block and links.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 3A

(RCV.LEN)

MCAL R0,X'3A',11
MCAL R0,58,11

opcode = 403AAB

PURPOSE:

Gets residual UltiNet range.

INPUT:

none

OUTPUT:

If successful:

IO.BIT= set to indicate success

T0 = length

If unsuccessful:

IO.BIT= zeroed to indicate failure

T0 = specifies which I/O code to the controller (1-6)
failed.

DESCRIPTION:

This is used on Honeywell systems to get the residual range from the UltiNet controller. It is equivalent to a series of 6 VIO instructions, but faster.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 3D

(SET.FL.DEN)

MCAL R0,X'3D',11
MCAL R0,61,11

opcode = 403DAB

PURPOSE:

Set density of floppy disk. Used on PC implementation only.

INPUT:

none

OUTPUT:

none

DESCRIPTION:

On a PC, causes floppy diskette parameters to be set properly for the type of diskette inserted into the drive. (Presently used only on PC-AT which handles diskettes of two formats.)

A NOP on all machines other than a PC.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 3E

(XFER.CLOCK)

MCAL R0,X'3E',11
MCAL R0,62,11

opcode = 403EAB

PURPOSE:

Sets time and date from internal clock. Used on PC implementation only.

INPUT:

none

OUTPUT:

none

DESCRIPTION:

On a PC, this is called at system initialization time on processors having a hardware time of day clock.

A NOP on all machines other than a PC.

DATA STRUCTURES:

data structures modified, if applicable

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 3F

(SET.BATCH.TM)

MCAL R0,X'3F',11
MCAL R0,63,11

opcode = 403FAB

PURPOSE:

Sets batch time limit to use all of memory.

INPUT:

T0 = new time limit

OUTPUT:

T0 = current time limit

DESCRIPTION:

This sets or returns the maximum number of seconds that must pass with no interactive jobs running in order for batch jobs to be allowed to use all of memory. If T0 is non-zero, it contains the new value, in seconds. If T0 is zero, the current value is not changed. In either case, the (new) current value is returned in T0.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 40

(PERIPH.RD)

MCAL Rr,X'40',11
MCAL Rr,64,11

opcode = 4r40AB

PURPOSE:

Multi-byte peripheral read.

INPUT:

T0 = PIB number
H2 = count
H3 = flags
Rr = address to start copying bytes to

OUTPUT:

T1 = number of bytes copied

DESCRIPTION:

This copies multiple bytes from the designated PIB's terminal input buffer to where the register points.

If T0 refers to a PIB other than the one making the monitor call, the TRAP roadblock is set.

If the input buffer is empty, zero is put in T1 and execution resumes at the instruction after the MCAL.

Characters are removed from the terminal input buffer to the locations pointed to by Rr until one of the following occurs:

1. The input buffer is emptied.
2. The end of the frame that Rr points to is reached.
3. As many characters as specified by the count in H2 have been moved.
4. If virtual B24 (the low bit of H3) is set, a control character (one less than X'20') is moved.

DATA STRUCTURES:

Modifies the terminal input buffer.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 41

(PERIPH.WRT)

MCAL Rr,X'41',11
MCAL Rr,65,11

opcode = 4r41AB

PURPOSE:

Multi-byte peripheral write.

INPUT:

T0 = PIB number
Rr = address of string to write

OUTPUT:

T1 = number of characters actually written

DESCRIPTION:

If T0 refers to a PIB other than the one making the monitor call, the TRAP roadblock is set.

The string of bytes pointed to by Rr is output (or buffered for output) on the port specified by T0, as if the WRITE instruction had been executed for each byte. The end of the string is marked by the first of the following conditions that is encountered:

1. No more characters may be buffered by the terminal I/O interface. Typically, the output buffer is full.
2. A segment mark is found. (The segment mark is not transmitted.)
3. The string reaches the end of its frame.

The number of characters sent is stored in T1. If this is not the entire string, it is the responsibility of the virtual code to send more later.

Execution resumes with the next instruction.

DATA STRUCTURES:

Modifies the terminal output buffer.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 44

(VMS.SPOOL)

MCAL Rr,X'44',11
MCAL Rr,68,11

opcode = 4r44AB

PURPOSE:

Passes spool file to VMS for printing. Used on VAX implementations only.

INPUT:

Rr = first FID of spool file

OUTPUT:

none

DESCRIPTION:

On a VAX, causes the VMS detached process to create a VMS file, to move Ultimate spool file data into the VMS file, and to send it to the VMS job controller for printing.

A NOP on all machines other than VAX.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 45

(VMS.TAPE)

MCAL R0,X'45',11
MCAL Rr,X'45',11
MCAL R0,69,11

opcode = 4045AB
opcode = 4r45AB

PURPOSE:

Mechanism to issue tape commands to various tape drivers in the VAX. Used on VAX implementations only.

INPUT:

For label I/O:

Rr = points to label buffer in tape control block

For non-label I/O:

Rr = R0

OUTPUT:

none

DESCRIPTION:

On a VAX, this gives information on the type of I/O, size of the transfer, which device to use, etc. This information is used to build I/O packets to issue to VMS tape drivers.

A NOP on all machines other than VAX.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 46

(VMS.OFF)

MCAL R0,X'46',11
MCAL R0,70,11

opcode = 4046AB

PURPOSE:

VMS Logoff. Used on VAX implementation only.

INPUT:

none

OUTPUT:

none

DESCRIPTION:

On a VAX, this returns to VMS the control of the terminal attached to the PIB making the MCAL.

A NOP on all machines other than VAX.

DATA STRUCTURES:

Modifies the following data structures:

- terminal I/O table
- SNU links
- clock request block and links

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 47

(VMS.MSG)

MCAL Rr,X'47',11
MCAL Rr,71,11

opcode = 4r47AB

PURPOSE:

File transfer. Used on VAX implementations only.

INPUT:

Rr = address of byte zero of the frame to be copied

OUTPUT:

T0 = zero (0) to indicate success

DESCRIPTION:

On a VAX, this copies data between Ultimate memory and VAX memory. To use this MCAL, the virtual process fills a frame with the following information:

Bytes Use

- 0-1 Number of bytes of meaningful data in frame.
- 2 Function code.
- 3 Zero (0). This is a first time flag that is altered by the monitor.
- 4-511 The data to transfer.

If byte 3 of the frame contains a zero (0):

- 1. As many words as indicated by bytes 0-1 are copied from the frame to a corresponding I/O buffer in VMS memory.
- 2. The SLEEP roadblock for the Ultimate process is set, the program counter is backed up to the MCAL instruction, and the process is detached.
- 3. Event flag 2 (file transfer attention) is set for the corresponding VMS process.
- 4. If byte 2 contains a 4 or 5, the VMS process is awakened.
- 5. A one (1) is written into byte 3 of the frame and the frame's write required flag is set.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

If byte 3 of the frame does not contain a zero and the function code is negative, this indicates that the VMS process is done:

1. As many words as indicated by the byte count in bytes 0-1 of the corresponding VMS I/O buffer are copied into the frame.
2. The frame's write-required flag is set.
3. T0 in the accumulator is zeroed to indicate success.
4. The Ultimate process is resumed.

If byte 3 of the frame does not contain a zero and the function code is not negative, this means the operator pressed the BREAK key, etc.:

1. The PC is backed up and the process is put to sleep again.

An illegal opcode on all machines other than VAX.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 48

(PC.MSG)

MCAL R0,X'48',11
MCAL R0,72,11

opcode = 4048AB

PURPOSE:

File transfer on PC. Used on PC implementation only.

INPUT:

Read or write flag, drive #, filename, directory

OUTPUT:

none

DESCRIPTION:

On a PC, performs a read or write (as specified by the flag). On PC implementations, the disk space is partitioned for DOS and Ultimate operating systems. This MCAL calls a DOS routine to cross the partition and move the file data in or out of virtual memory. One item at a time is transferred, in 1024-byte increments.

An illegal opcode on machines other than a PC.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 49

(FAKE.READ)

MCAL R0,X'49',11
MCAL R0,73,11

opcode = 4049AB

PURPOSE:

Fake read. FID in D0.

INPUT:

D0 = frame to fake read

OUTPUT:

none

DESCRIPTION:

The memory map is modified so that the referenced frame is in memory, but it is not read from disk. It is assumed that the frame will be initialized after this.

DATA STRUCTURES:

Modifies the following data structures:

- buffer table
- age links
- disk I/O queues (on some systems)

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 4A

(RFLAGS.CLR)

MCAL R0,X'4A',11
MCAL R0,74,11

opcode = 404AAB

PURPOSE:

Clears bits in Rflags.

INPUT:

T0 = Mask containing bits to clear

T1 = PIB number (negative means self)

OUTPUT:

T0 = new value of RFLAGS

DESCRIPTION:

This is the method of enabling and disabling XON/XOFF, typeahead, etc. If the PIB number is negative, the currently executing PIB is used.

Each bit that is set in the mask is cleared in RFLAGS.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

MCAL 4B

(RFLAGS.SET)

MCAL R0,X'4B',11
MCAL R0,75,11

opcode = 404BAB

PURPOSE:

Sets bits in Rflags.

INPUT:

T0 = mask containing bits to set

T1 = PIB number (negative means self)

OUTPUT:

T0 = new value of RFLAGS

DESCRIPTION:

This is the method of enabling and disabling XON/XOFF, typeahead, etc. If the PIB number is negative, the currently executing PIB is used.

Each bit that is set in the mask is set in RFLAGS.

DATA STRUCTURES:

No data structures modified.

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY

INDEX

ALARM.CLOCK	25	TL.READ	8
CLEAR.INP	48	VMCAL	28
CLOCK.CANCEL	26	VMS.MSG	64
CLR.OUT	51	VMS.OFF	6
CMD.FAKE.WT	6	VMS.SPOOL	61
CMD.MAXFID	5	VMS.TAPE	62
CMD.STAT	4	VOPT.AND	47
DB.ENT	12	VOPT.OR	46
DB.LV	13	WAIT	18
DISK.ERR	32	WARM.DUMP	11
DISK.STAT	53	WRITE.WAIT	54
DSABL.DSK	42	XFER.CLOCK	57
FAKE.RD	16		
FAKE.READ	67		
FAKE.WT	17		
FORCE.WRITE	33		
FRM.LOCK	30		
FRM.UNLOCK	29		
GET.ID	7		
GET.TIME	35		
INT.CANCEL	27		
LINK.CNT	2		
LOCK	37,39		
MTB	3		
MTBF	1		
N.GET.ID	24		
PANEL	9		
PC.MSG	66		
PERIPH.RD	59		
PERIPH.RD.ONE	50		
PERIPH.WRT	60		
PERIPH.WRT.ONE	49		
PIB.AND	14		
PIB.ATL	41		
PIB.OR	15		
PIB.PEEK	22		
PIB.POKE	23		
PIB.XPCB	52		
QUERY	20		
QUEUE.READ	43		
RCV.LEN	55		
RFLAGS.CLR	68		
RFLAGS.SET	69		
ROM	36		
RTC.CALIB	44		
SET.BATCH.TM	58		
SET.FL.DEN	56		
SET.TIME	34		
SLEEP or SLEEP:	31		
START.IO.PIB	10		
TEST.INP	45		
TIME	35		

CONFIDENTIAL MATERIAL
FOR INTERNAL USE ONLY