UNITED STATES GEOLOGICAL SURVEY

Modifications to the Megatek Wand Software Package for the Model 3355 Color Graphics Workstation with IEEE-488 Bus Interface Option

Lawrence M. Baker

Open-File Report 86-212

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards and stratigraphic nomenclature. Any use of trade names is for descriptive purposes only and does not imply endorsement by the USGS.

Although this program has been tested by the Geological Survey, United States Department of the Interior, no warranty, expressed or implied, is made by the Geological Survey as to the accuracy and functioning of the program and related program material nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the Geological Survey in connection therewith.

1986

Abstract

This report describes changes made by USGS to the Megatek Wand subroutine library and the VAX/VMS IEU11 IEEE-488 bus interface device driver for the model 3355 color graphics workstation to eliminate the requirement for VMS LOG_IO privilege in user-level programs. Minor changes were made to the IEU11 device driver, IXDRIVER, to the hardware-dependent Wand module, WANDHD, and a new program was created, WINIT, to be invoked during the sytem startup procedure to initialize the IEEE-488 bus interface.

1 Introduction

The Wand software package for the Megatek model 3355 color graphics workstation consists of a graphics subroutine library, which must be linked with user-level programs, and an operating system device driver, which must be loaded and initialized for the proper hardware configuration by the system manager. The actual device driver shipped with the system depends on the hardware interface option ordered with the 3355. Additionally, the Wand library is divided into a set of hardware-independent subroutines and a set of hardware-dependent subroutines, which vary according to the hardware interface option selected.

In order to obtain the DEC VT100 terminal emulation feature, the USGS chose an intelligent controller interface called LIFE™ Interface for the 3355 which uses the IEEE-488 parallel bus for high speed transfers of graphical data. The corresponding interface supplied by Megatek for use inside the VAX-11/750 computer is a DEC IEU11 UNIBUS IEEE-488 bus controller. Megatek also supplies what appears to be an unmodified copy of the DEC VAX/VMS device driver for the IEU11, named IXDRIVER. This device driver requires that user-level programs making use of its features have the VMS LOG_IO privilege, which is highly unusual and extremely dangerous on a typical time-sharing VMS system. (A user with LOG_IO privilege can perform logical I/O operations on any device in the system, including disk drives, which effectively disables any operating system safeguards that prevent the accidental or deliberate corruption of, or unauthorized access to, files and volumes.)

Initially, only those programs which required the 3355 were allowed to have this privilege (by INSTALLing them with the VMS INSTALL utility). Gradually, though, it became necessary to grant LOG_IO privilege to selected user accounts to avoid requiring the intervention of the system manager every time a new program was generated. At this point a more permanent solution was sought that would both remove this obstacle and restore the security of the system.

The VAX/VMS I/O User's Guide contains a description of the different program logic used to validate physical, logical, and virtual I/O requests, which will not be elaborated here. The important features to note are that a program with LOG_IO privilege may perform physical I/O operations on an unmounted device (which is the state of the IEU11 when used with the Megatek 3355 workstation), and that no special privilege is required to perform logical I/O to an unmounted device. (Standard file I/O is called virtual I/O, which is normally performed only on a mounted volume, such as a disk or tape.)

2 Modifications

As mentioned above, the Wand subroutine library is divided into hardware-independent and hardware-dependent modules. The hardware-dependent module, WANDHD.FOR, was examined to locate all I/O requests to the IEEE-488 bus controller, IXA0:. This occurred in very few places, and most of the I/O function codes used to access the controller were declared using Fortran-77 PARAMETER statements, which made modifications relatively easy by localizing the changes in one place for each subroutine altered. The VMS IEU11 device driver, IXDRIVER.MAR, was examined to determine the source of the requirement for LOG_IO privilege.

It was found that virtually all of the I/O function codes used to implement the IEU11 device driver were physical I/O functions, which explained why LOG_IO privilege was required. The strategy chosen was to replace the physical I/O functions originally defined in IXDRIVER with suitable logical I/O functions that did not conflict with any standard VMS usage.

VMS allows a driver to define up to 32 physical, 16 logical, and 16 virtual I/O function codes. Each I/O request also carries with it a 10-bit I/O function modifier, which is typically used to distinguish from among several similar operations grouped together into one I/O function code (*e.g.*, reading from a terminal with or without a prompt specified as part of the read operation). Unfortunately, this convention is not followed in IXDRIVER, and there are not enough logical I/O function codes available to map the physical I/O function codes without redefining some of them as I/O function code/modifier pairs (*e.g.*, issue IEEE-488 bus command and issue auxiliary command). Fortunately, this had little effect on the code in either the driver or the Wand device-dependent subroutines. One physical I/O function code was left undisturbed, IO$_INITIALIZE, on the assumption that it should not be possible for a non-privileged user to change device configurations in a time-sharing system. This is the only I/O function used by Wand that still requires LOG_IO privilege, but since it need only be isued once after the IEU11 device is connected, a special program was written, WINIT, whose sole purpose is to issue an IO$_INITIALIZE to the IEEE-488 bus interface from the system startup command procedure.

3 Summary

The following table summarizes the changes made to the I/O function codes in IXDRIVER. The appendices contain the source code text and instructions for making these changes to IXDRIVER.MAR and WANDHD.FOR, and the source code text for the new module, WINIT.FOR.

I/O Function Code Symbol	Original Function Code	New Function Code/ Modifier Pair
IO$_SERVICE	1	22
IO$_SER_POLL	2	25
IO$_PAR_POLL	6	26
IO$_COMMAND	7	2A
IO$_GO_TO_CACS	8	23 ! 200
IO$_GO_TO_CSBS	9	23 ! 400
IO$_AUXILIARY	A	2A ! 40
IO$_COMMANDS	12	2B
IO$_SETEVENT	D	28
IO$_REC_EVENT	E	29
IO$_PASSCONTROL	F	2C
IO$_PARPOLLCON	13	2D
IO$_LOADPARPOLL	14	2E

Table 1. Modified IXDRIVER I/O Function Codes and Modifiers.

These modifications may be generally useful to any application making use of the DEC IEU11 IEEE-488 bus interface. However, the USGS has no such requirement and thus has not attempted to use any device other the the Megatek model 3355 on the IEEE-488 bus with the modified IXDRIVER.

4 References

4.1 DEC Documentation

1. VAX/VMS I/O User's Guide (Volume 1) (Order no. AA-M540A-TE)
2. VAX/VMS I/O User's Guide (Volume 2) (Order no. AA-M541A-TE)
3. VAX/VMS Guide to Writing a Device Driver (Order no. AA-H499C-TE)
4. VAX/VMS System Management and Operations Guide (Order no. AA-M547A-TE)

4.2 Megatek Documentation

1. Wand Installation Guide 3300 Parallel VAX/VMS (Document 0252-0072-00)
2. Wand Release Notes 3300 Parallel VAX/VMS (Document 0260-0071-00)

APPENDIX A

Modifications to IXDRIVER.MAR

A.1 Source Code Modifications

The following changes include those required to upgrade IXDRIVER to run under VMS version 4.x (identified by audit trail 1.1M). These additional changes have no adverse effect on IXDRIVER for systems still running VMS version 3.x.

A.1.1 Change the device driver identification code in line 2 from

```
        .IDENT  /1.0/
```

to

```
        .IDENT  /1.1M/
```

A.1.2 Change the comment in line 73 from

```
;       NOTE: THE ISSUING PROCESS MUST HAVE LOG_IO PRIVILEGE
```

to

```
;       NOTE: THE ISSUING PROCESS MUST HAVE LOG_IO PRIVILEGE
;             TO INITIALIZE THE DEVICE.                         ; 1.0M
```

A.1.3 Immediately before the comment in line 78 (IO$_WRITELBLK), add

```
;       IO$_WRITEVBLK           Same as IO$_WRITEPBLK           ; 1.0M
```

Modifications to IXDRIVER.MAR

A.1.4 Change the comment in line 90 from

```
;       IO$_AUXILIARY          Issue an auxiliary command
```

to

```
;               IO$M_AUXILIARY  Issue an auxiliary command                ; 1.0M
```

A.1.5 Move and change the comments in lines 105-107 from

```
;       IO$_GO_TO_CACS          Set unit to controller active state
;       IO$_GO_TO_CSBS          Set unit to controller standby state
;
```

to immediately after line 111 (IO$_SETMODE)

```
;               IO$M_GO_TO_CACS Set unit to controller active state       ; 1.0M
;               IO$M_GO_TO_CSBS Set unit to controller standby state      ; 1.0M
```

A.1.6 Immediately after the comment in line 130 (; MODIFIED BY:), add

```
;
;       1.0M   3-Apr-85  L. M. Baker    Change all locally defined I/O function
;                                       codes to be logical functions instead
;                                       of physical functions (LOG_IO privilege
;                                       is required only for IO$_INITIALIZE).
;       1.1M   1-May-85  L. M. Baker    Upgrade to VMS V4.0.
```

A.1.7 Immediately after line 148 ($DEVDEF), add

```
        $DYNDEF                     ; Dynamic data structure type codes ; 1.1M
```

A.1.8 Immediately after line 151 ($IODEF), add

```
        .IF NDF IO$V_ENABL_ALT                                          ; 1.1M
 $EQU   IO$V_ENABL_ALT  11                                              ; 1.1M
 $EQU   IO$M_ENABL_ALT  <^X800>                                         ; 1.1M
 $EQU   IO$V_DSABL_ALT  12                                              ; 1.1M
 $EQU   IO$M_DSABL_ALT  <^X1000>                                        ; 1.1M
        .ENDC   ; NDF IO$V_ENABL_ALT                                    ; 1.1M
```

Modifications to IXDRIVER.MAR

A.1.9 Immediately after line 154 ($JIBDEF), add

```
        $PCBDEF                         ; Process control block offsets ; 1.1M
```

A.1.10 Immediately after line 155 ($PRIDEF), add

```
        $PRVDEF                         ; Privilege bits                ; 1.1M
```

A.1.11 Change the I/O Function Code definitions in lines 184-196 from

```
        IO$_SERVICE     =    1        ; Make a Service Request
        IO$_SER_POLL    =    2        ; Serial Poll
        IO$_PAR_POLL    =    6        ; Parallel Poll
        IO$_COMMAND     =    7        ; IEEE Bus Commands
        IO$_GO_TO_CACS =     8        ; Go to CACS
        IO$_GO_TO_CSBS  =    9        ; Go to CSBS
        IO$_AUXILIARY   =    ^XA      ; Issue an auxiliary command
        IO$_COMMANDS    =    ^X12     ; IEEE bus commands in string mode
        IO$_SETEVENT    =    ^XD      ; Set event mask
        IO$_REC_EVENT   =    ^XE      ; Recognize event
        IO$_PASSCONTROL =    ^XF      ; Pass control
        IO$_PARPOLLCON  =    ^X13     ; Parallel poll configure
        IO$_LOADPARPOLL =    ^X14     ; Load parallel poll
```

to

```
        IO$_SERVICE     =    ^X22     ; Make a Service Request           ; 1.0M
        IO$_SER_POLL    =    ^X25     ; Serial Poll                      ; 1.0M
        IO$_PAR_POLL    =     X26     ; Parallel Poll                    ; 1.0M
        IO$_COMMAND     =    ^X2A     ; IEEE Bus Commands                ; 1.0M
        IO$_GO_TO_CACS  =    ^X223    ; Go to CACS                       ; 1.0M
        IO$_GO_TO_CSBS  =    ^X423    ; Go to CSBS                       ; 1.0M
        IO$_AUXILIARY   =    ^X6A     ; Issue an auxiliary command       ; 1.0M
        IO$_COMMANDS    =    ^X2B     ; IEEE bus commands in string mode ; 1.0M
        IO$_SETEVENT    =    ^X28     ; Set event mask                   ; 1.0M
        IO$_REC_EVENT   =    ^X29     ; Recognize event                  ; 1.0M
        IO$_PASSCONTROL =    ^X2C     ; Pass control                     ; 1.0M
        IO$_PARPOLLCON  =    ^X2D     ; Parallel poll configure          ; 1.0M
        IO$_LOADPARPOLL =    ^X2E     ; Load parallel poll               ; 1.0M
```

A.1.12 Immediately before line 211 (IO$M_ENA_SRQ), add

```
        IO$M_GO_TO_CACS =       ^X200   ; Go to CACS                     ; 1.0M
        IO$V_GO_TO_CACS =       9                                        ; 1.0M
        IO$M_GO_TO_CSBS =       ^X400   ; Go to CSBS                     ; 1.0M
        IO$V_GO_TO_CSBS =       10                                       ; 1.0M
        IO$M_AUXILIARY  =       ^X40    ; Issue an auxiliary command     ; 1.0M
        IO$V_AUXILIARY   =      6                                    ; 1.0M
```

A.1.13 Remove lines 821 and 827-828 from the legal functions table

```
;       AUXILIARY,-                     ; Issue an auxiliary command    ; 1.0M

;       GO_TO_CACS,-                    ; Go to CACS                    ; 1.0M
;       GO_TO_CSBS,-                    ; Go to CSBS                    ; 1.0M
```

A.1.14 Remove lines 854 and 860-861 from the FDT_INIT dispatch table

```
;       AUXILIARY,-                     ; Issue an auxiliary command    ; 1.0M

;       GO_TO_CACS,-                    ; Go to CACS                    ; 1.0M
;       GO_TO_CSBS,-                    ; Go to CSBS                    ; 1.0M
```

A.1.15 Remove lines 904 and 909-910 from the EXE$QIODRVPKT dispatch table

```
;       AUXILIARY,-                     ; Issue an auxiliary command    ; 1.0M

;       GO_TO_CACS,-                    ; Go to CACS                    ; 1.0M
;       GO_TO_CSBS,-                    ; Go to CSBS                    ; 1.0M
```

A.1.16 Change lines 1335-1336 in IX_START (after label 10$) from

```
        CMPB    R2,#IO$_AUXILIARY       ; Test is this is a holdoff release
        BNEQ    20$                     ; No, then automatically generate release
to
;                                                                       ; 1.0M
; Test if this is a holdoff release.  If not, then automatically        ; 1.0M
; generate release.                                                     ; 1.0M
;                                                                       ; 1.0M
      CMPB    R2,#IO$_COMMAND         ; Command request?                ; 1.0M
      BNEQ    20$                     ; No                              ; 1.0M
      BBC     #IO$V_AUXILIARY,R1,20$  ; Branch if not auxiliary command ; 1.0M
```

A.1.17 Change the I/O function dispatcher case selector in lines 1348-1374 from

```
60$:    CASEW   R2,#1,#:X27-1           ; Function computed go to
```

Modifications to IXDRIVER.MAR

```
70$:    .WORD   IX_SERVICE-70$          ; 01 - SERVICE REQUEST
        .WORD   IX_SER_POLL-70$         ; 02 - SERIAL POLL
        .WORD   80$-70$                 ; 03 - Not used
        .WORD   IX_INITIALIZE-70$       ; 04 - INITIALIZE
        .WORD   80$-70$                 ; 05 - Not used
        .WORD   IX_PAR_POLL-70$         ; 06 - PARALLEL POLL
        .WORD   IX_COMMAND-70$          ; 07 - COMMAND
        .WORD   IX_GO_TO_CACS-70$       ; 08 - GO TO CACS
        .WORD   IX_GO_TO_CSBS-70$       ; 09 - GO TO CSBS
        .WORD   IX_AUXILIARY-70$        ; 0A - AUXILIARY COMMAND
        .WORD   IX_WRITEPBLK-70$        ; 0B - WRITE
        .WORD   IX_READPBLK-70$         ; 0C - READ
        .WORD   IX_SETEVENT-70$         ; 0D - SET EVENT MASK
        .WORD   IX_REC_EVENT-70$        ; 0E - RECOGNIZE EVENT
        .WORD   IX_PASSCONTROL-70$      ; 0F - PASS CONTROL
        .WORD   80$-70$[2]
        .WORD   IX_COMMANDS-70$         ; 12 - COMMAND
        .WORD   IX_PARPOLLCON-70$       ; 13 - PARALLEL POLL CONFIGURE
        .WORD   IX_LOADPARPOLL-70$      ; 14 - LOAD PARALLEL POLL REGISTER
        .WORD   80$-70$[11]
        .WORD   IX_WRITEPBLK-70$        ; 20 - WRITE
        .WORD   IX_READPBLK-70$         ; 21 - READ
        .WORD   80$-70$
        .WORD   IX_SETMODE-70$          ; 23 - SETMODE
        .WORD   80$-70$[3]
        .WORD   IX_SENSEMODE-70$        ; 27 - SENSEMODE
```

to

```
60$:    CASEW   R2,#4,#^X2E-1         ; Function computed go to           ; 1.0M
70$:    .WORD   IX_INITIALIZE-70$     ; 04 - INITIALIZE                   ; 1.0M
        .WORD   80$-70$[6]            ; 05-0A - Not used                  ; 1.0M
        .WORD   IX_WRITEPBLK-70$      ; 0B - WRITE                        ; 1.0M
        .WORD   IX_READPBLK-70$       ; 0C - READ                         ; 1.0M
        .WORD   80$-70$[19]           ; 0D-1F - Not used                  ; 1.0M
        .WORD   IX_WRITEPBLK-70$      ; 20 - WRITE                        ; 1.0M
        .WORD   IX_READPBLK-70$       ; 21 - READ                         ; 1.0M
        .WORD   IX_SERVICE-70$        ; 22 - SERVICE REQUEST              ; 1.0M
        .WORD   IX_SETMODE-70$        ; 23 - SETMODE                      ; 1.0M
        .WORD   80$-70$               ; 24 - Not used                     ; 1.0M
        .WORD   IX_SER_POLL-70$       ; 25 - SERIAL POLL                  ; 1.0M
        .WORD   IX_PAR_POLL-70$       ; 26 - PARALLEL POLL                ; 1.0M
        .WORD   IX_SENSEMODE-70$      ; 27 - SENSEMODE                    ; 1.0M
        .WORD   IX_SETEVENT-70$       ; 28 - SET EVENT MASK               ; 1.0M
        .WORD   IX_REC_EVENT-70$      ; 29 - RECOGNIZE EVENT              ; 1.0M
        .WORD   IX_COMMAND-70$        ; 2A - COMMAND                      ; 1.0M
        .WORD   IX_COMMANDS-70$       ; 2B - COMMAND                      ; 1.0M
        .WORD   IX_PASSCONTROL-70$    ; 2C - PASS CONTROL                 ; 1.0M
        .WORD   IX_PARPOLLCON-70$     ; 2D - PARALLEL POLL CONFIGURE      ; 1.0M
        .WORD   IX_LOADPARPOLL-70$    ; 2E - LOAD PARALLEL POLL REGISTER ; 1.0M
```

A.1.18 Immediately after line 2049 in IX_COMMAND (labelled IX_COMMAND:), add

```
        BITW    #IO$M_AUXILIARY,R1      ; Auxiliary command?            ; 1.0M
        BNEQ    IX_AUXILIARY            ; If set, yes                   ; 1.0M
```

A.1.19 Change line 3002 in IX_SETMODE from

```
50$:    MOVZWL  #SS$_NORMAL,-(SP)       ; Retrurn success
```

to

```
;                                                                       ; 1.0M
; Test for controller state change requests                             ; 1.0M
;                                                                       ; 1.0M
50$:    BBC     #IO$V_GO_TO_CACS,R1,52$ ; Set unit to controller        ; 1.0M
        BRW     IX_GO_TO_CACS           ;   active state                ; 1.0M
52$:    BBC     #IO$V_GO_TO_CSBS,R1,54$ ; Set unit to controller        ; 1.0M
        BRW     IX_GO_TO_CSBS           ;   standby state               ; 1.0M
54$:    MOVZWL  #SS$_NORMAL,-(SP)       ; Return success                ; 1.0M
```

A.2 DCL Commands

The following DCL commands may be issued to assemble and link the new version of IXDRIVER. (Section 3.3 of the Wand Installation Guide 3300 Parallel contains more explicit information on installation and testing of IXDRIVER.)

```
$ Macro /List IXDRIVER + Sys$Library:Lib.mlb/Library
$ Link  /Map /NoTrace IXDRIVER,Sys$Input/Options,-
        Sys$System:Sys.stb/Selective_Search
BASE=0
```

APPENDIX B

Modifications to WANDHD.FOR

B.1 Source Code Modifications

B.1.1 Immediately after line 12 in Subroutine MDA488, add

```
C       *       01      03-APR-85       MODIFIED I/O FUNCTION CODES   *
C       *       02      03-APR-85       REMOVED NEED FOR LOG_IO PRIV; *
C       *                                 DEVICE MUST BE INITIALIZED  *
C       *                                 BY RUNNING WINIT, WHICH     *
C       *                                 DOES REQUIRE LOG_IO PRIV    *
C       *       03      03-APR-85       REFERENCE CORRECT FORMAT STMT *
```

B.1.2 Immediately after line 335 in Subroutine MDD488, line 543 in Subroutine MDW488, and line 881 in Subroutine MDX488, add

```
C       *       01      03-APR-85       MODIFIED I/O FUNCTION CODES   *
```

B.1.3 Change the I/O Function Code definitions in lines 64-75 in Subroutine MDA488, lines 388-399 in Subroutine MDD488, lines 669-680 in Subroutine MDW488, and lines 996-1007 in Subroutine MDX488 from

```
PARAMETER (IO$_SERVICE     = '00000001'X)
PARAMETER (IO$_SER_POLL    = '00000002'X)
PARAMETER (IO$_GO_TO_CACS  = '00000008'X)
PARAMETER (IO$_GO_TO_CSBS  = '00000009'X)
PARAMETER (IO$_AUXILIARY   = '0000000A'X)
PARAMETER (IO$_SETEVENT    = '0000000D'X)
PARAMETER (IO$_REC_EVENT   = '0000000E'X)
PARAMETER (IO$_PASSCONTROL = '0000000F'X)
PARAMETER (IO$_COMMANDS    = '00000012'X)
PARAMETER (IO$_PARPOLLCON  = '00000013'X)
PARAMETER (IO$_LOADPARPOLL = '00000014'X)
PARAMETER (IO$_PAR_POLL    = '00000006'X)
```

to

```
        PARAMETER (IO$_SERVICE     = '00000022'X)                        ! 01
        PARAMETER (IO$_SER_POLL    = '00000025'X)                        ! 01
        PARAMETER (IO$_GO_TO_CACS  = '00000223'X)                        ! 01
        PARAMETER (IO$_GO_TO_CSBS  = '00000423'X)                        ! 01
        PARAMETER (IO$_AUXILIARY   = '0000006A'X)                        ! 01
        PARAMETER (IO$_SETEVENT    = '00000028'X)                        ! 01
        PARAMETER (IO$_REC_EVENT   = '00000029'X)                        ! 01
        PARAMETER (IO$_PASSCONTROL = '0000002C'X)                        ! 01
        PARAMETER (IO$_COMMANDS    = '0000002B'X)                        ! 01
        PARAMETER (IO$_PARPOLLCON  = '0000002D'X)                        ! 01
        PARAMETER (IO$_LOADPARPOLL = '0000002E'X)                        ! 01
        PARAMETER (IO$_PAR_POLL    = '00000026'X)                        ! 01
```

B.1.4 Remove the IO$_INITIALIZE request from lines 242-247 in Subroutine MDA488

```
C               IMASK = '0037'X                                          ! 02
C               ISTATUS = SYS$QIOW (,%VAL(CHAN),%VAL(IO$_INITIALIZE),    ! 02
C       1                           IOSB,,,                              ! 02
C       1                       %VAL(IMASK),%VAL(0),,,,)                 ! 02
C               IF ((ISTATUS .EQ. SS$_NORMAL) .AND.                      ! 02
C       1           (IOSB(1) .EQ. SS$_NORMAL)) THEN                      ! 02
```

B.1.5 Remove the IO$_INITIALIZE failure code from lines 269-279 in Subroutine MDA488

```
C               ELSE                                                     ! 02
C ** INITIALIZATION FAILURE
C                 IF (ISTATUS .EQ. SS$_NORMAL) THEN                      ! 02
C                    CALL SYS$GETMSG(%VAL(IOSB(1)), MSGLEN, MESSAGE,     ! 02
C       1                 %VAL(15),)                                     ! 02
C                 ELSE                                                   ! 02
C                    CALL SYS$GETMSG(%VAL(ISTATUS), MSGLEN, MESSAGE,     ! 02
C       1                 %VAL(15),)                                     ! 02
C                 END IF                                                 ! 02
C                 WRITE (UNIT=SRCMSG,FMT=6100) GXNAM                     ! 02
C               END IF                                                   ! 02
```

B.1.6 Change the format statement reference in line 288 in Subroutine MDA488 from

```
            WRITE (UNIT=SRCMSG,FMT=6100) GXNAM
```

to

```
            WRITE (UNIT=SRCMSG,FMT=6300) GXNAM                          ! 03
```

B.1.7 Remove the IO$_INITIALIZE failure message from line 317 in Subroutine MDA488

```
C6100   FORMAT ('INITIALIZE FAILURE ON DEVICE: ',A)                     ! 02
```

B.2 DCL Commands

The following DCL commands may be used to compile the new version of WANDHD and replace the old modules in the Wand Workstation level and System level routines library, WAND.OLB.

```
$ Fortran /List    WANDHD
$ Library /Replace WAND WANDHD
```

APPENDIX C

New Module WINIT.FOR

C.1 New Source Code

The following is the VMS Fortran-77 source code for the new program, WINIT, which is executed by the system manager to initialize the IEU11 IEEE-488 bus controller.

```
CBEGIN PROGRAM WINIT.FOR
C       ****************************************************************
C       *                                                              *
C       *                       MEGATEK  WAND                          *
C       *                        VERSION 3.1                           *
C       *                   WHIZZARD 3300 PARALLEL                     *
C       *                                                              *
C       *      NAME:    WINIT.FOR                                      *
C       *                                                              *
C       *      REVISION HISTORY:                                       *
C       *      REV     DATE            DESCRIPTION                     *
C       *      00      03-APR-85       ORIGINAL VERSION                *
C       *                                                              *
C       *         US GOVERNMENT RESTRICTED RIGHTS LEGEND               *
C       *                                                              *
C       *      USE, DUPLICATION, OR DISCLOSURE  IS  SUBJECT  TO        *
C       *      RESTRICTIONS STATED IN CONTRACT NO. GS-00C-02493        *
C       *      WITH MEGATEK CORPORATION.                               *
C       *                                                              *
C       *      THIS PROGRAM IS PROPRIETARY TO MEGATEK CORPORATION      *
C       *      AND IS  DISTRIBUTED  UNDER  A  LICENSE  AGREEMENT.      *
C       *      REPRODUCTION, DISCLOSURE, OR USE, IN  WHOLE  OR IN      *
C       *      PART, OTHER THAN AS  SPECIFIED  IN THE LICENSE ARE      *
C       *      NOT TO BE  UNDERTAKEN  EXCEPT  WITH  PRIOR WRITTEN      *
C       *      AUTHORIZATION OF MEGATEK CORPORATION.                   *
C       *                                                              *
C       *              COPYRIGHT (C) MEGATEK CORP.,1984                *
C       *                      ALL RIGHTS RESERVED                     *
C       *                                                              *
C       ****************************************************************
C
        PROGRAM WINIT
```

New Module WINIT.FOR

```
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C               SYSTEM SERVICE FUNCTIONS
C               ------------------------
C
        INTEGER*4       SYS$ASSIGN
        INTEGER*4       SYS$QIOW
        INTEGER*4       SYS$GETMSG
C
C               SYSTEM SERVICE SYMBOLS
C               ----------------------
C
        INCLUDE '($IODEF)/NOLIST'
        INCLUDE '($SSDEF)/NOLIST'
C
C               DEC IEU11 SPECIFIC PARAMETERS
C               -----------------------------
C
C **  I/O FUNCTION CODES
        PARAMETER (IO$_AUXILIARY    = '0000006A'X)
C **  I/O FUNCTION MODIFIERS
        PARAMETER (IO$M_TIMOUT      = '00004000'X)
C
C               LOCAL VARIABLES
C               ---------------
C
C **  CHANNEL NUMBER ASSIGNED TO IEEE 488 DEVICE
        INTEGER*2       CHAN
C **  OPTIONS MASK
        INTEGER*4       IMASK
C **  SYSTEM SERVICE STATUS RETURN
        INTEGER*4       ISTATUS
C **  QUEUE I/O SERVICE STATUS BLOCK
        INTEGER*2       IOSB(4)
C **  SYSTEM SERVICE STATUS MESSAGE NUMBER
        INTEGER*2       MSGNUM
C **  SYSTEM SERVICE STATUS MESSAGE LENGTH
        INTEGER*2       MSGLEN
C **  SYSTEM SERVICE STATUS MESSAGE STRING
        CHARACTER*256   MESSAGE
C **  ERROR SOURCE MESSAGE FORMAT NUMBER
        INTEGER         SRCMSG
C
C               LOCAL PARAMETERS
C               ----------------
C
C **  GRAPHICS DISPLAY PHYSICAL DEVICE NAME
        CHARACTER*(*)   GXNAM
        PARAMETER (GXNAM = '_IXA00:')
C **  TIMEOUT, IN SECONDS
        INTEGER         TIMEOUT
        PARAMETER (TIMEOUT =     120)
```

New Module WINIT.FOR

```
C
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C       ASSIGN, INITIALIZE AND ISSUE INTERFACE CLEAR
C       --------------------------------------------
C
C **  ASSIGN CHANNEL TO DEVICE
        ISTATUS = SYS$ASSIGN (GXNAM,CHAN,,)
        IF (ISTATUS .NE. SS$_NORMAL) THEN
C **  ASSIGN FAILURE
           ASSIGN 6200 TO SRCMSG
           GOTO 8900
        END IF
        IOSB(1) = SS$_NORMAL
C **  INITIALIZE DEVICE PRIMARY ADDRESS 0
C     (MASTER CLEAR | SOFTWARE RESET | SICS | CACS | PRIMARY ADDRESS)
        IMASK = '0037'X
        ISTATUS = SYS$QIOW (,%VAL(CHAN),%VAL(IO$_INITIALIZE),
     1                    IOSB,,,%VAL(IMASK),%VAL(0),,,,)
        IF ((ISTATUS .NE. SS$_NORMAL) .OR.
     1      (IOSB(1) .NE. SS$_NORMAL)) THEN
C **  INITIALIZATION FAILURE
           ASSIGN 6100 TO SRCMSG
           GOTO 8900
        END IF
C **  ASSERT INTERFACE CLEAR (IFC)
        IMASK = '000F'X
        ISTATUS = SYS$QIOW (,%VAL(CHAN),%VAL(IO$_AUXILIARY),
     1                    IOSB,,,%VAL(IMASK),,,,,)
        IF ((ISTATUS .NE. SS$_NORMAL) .OR.
     1      (IOSB(1) .NE. SS$_NORMAL)) THEN
C **  INTERFACE CLEAR FAILURE
           ASSIGN 6000 TO SRCMSG
           GOTO 8900
        END IF
C **  RESET TIMEOUT VALUE
        IMASK = TIMEOUT
        ISTATUS = SYS$QIOW (,%VAL(CHAN),%VAL(IO$_SETMODE + IO$M_TIMOUT),
     1                    IOSB,,,%VAL(IMASK),,,,,)
        GOTO 9000
C
C       REPORT FAILURES
C       ---------------
C
 8900   TYPE 6400
 6400   FORMAT (' **** WAND VAX/VMS ERROR MESSAGE ****')
        TYPE SRCMSG, GXNAM
 6000   FORMAT (' INTERFACE CLEAR FAILURE ON DEVICE: ',A)
 6100   FORMAT (' INITIALIZE FAILURE ON DEVICE: ',A)
 6200   FORMAT (' ASSIGN FAILURE ON DEVICE: ',A)
        IF (ISTATUS .EQ. SS$_NORMAL) THEN
           MSGNUM = IOSB(1)
        ELSE
```

```
            MSGNUM = ISTATUS
         END IF
         CALL SYS$GETMSG (%VAL(MSGNUM),MSGLEN,MESSAGE,%VAL(15),)
         TYPE 6600, MESSAGE(1:MSGLEN)
 6600    FORMAT (' STATUS RETURN: ',A)
C
 9000    STOP
         END
```

C.2 DCL Commands

The following DCL commands may be used to compile and link WINIT.

```
$ Fortran /List WINIT
$ Link    /Map  WINIT
```