Table of contents

```
                        .MCALL  .MODULE
000000                  .MODULE OHANDL,RELEASE=V05,VERSION=01,COMMENT<Overlay Handler>,IDENT=NO,LIB=YES

                        ;                       COPYRIGHT (c) 1984 BY
                        ;             DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
                        ;                       ALL RIGHTS RESERVED.
                        ;
                        ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
                        ; ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
                        ; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER
                        ; COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
                        ; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
                        ; TRANSFERRED.
                        ;
                        ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
                        ; AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
                        ; CORPORATION.
                        ;
                        ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
                        ; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
```

```
                        ; MAS,SHD
                        
                        .SBTTL  THE RUN-TIME OVERLAY HANDLER
                        .ENABL  GBL
                        
                        ;+
                        ; THE FOLLOWING CODE IS INCLUDED IN THE USER'S PROGRAM BY THE
                        ; LINKER WHENEVER LOW MEMORY OVERLAYS ARE REQUESTED BY THE USER.
                        ; THE RUN-TIME LOW MEMORY OVERLAY HANDLER IS CALLED BY A DUMMY
                        ; SUBROUTINE OF THE FOLLOWING FORM:
                        ;
                        ;       JSR     R5,$OVRH        ;CALL TO COMMON CODE FOR LOW MEMORY OVERLAYS
                        ;       .WORD   <OVERLAY # *6>  ;# OF DESIRED SEGMENT
                        ;       .WORD   <ENTRY ADDRESS> ;ACTUAL CORE ADDRESS (VIRTUAL ADDRESS)
                        ;
                        ; ONE DUMMY ROUTINE OF THE ABOVE FORM IS STORED IN THE RESIDENT PORTION
                        ; OF THE USER'S PROGRAM FOR EACH ENTRY POINT TO A LOW MEMORY OVERLAY SEGMENT.
                        ; ALL REFERENCES TO THE ENTRY POINT ARE MODIFIED BY THE LINKER TO BE
                        ; REFERENCES TO THE APPROPRIATE DUMMY ROUTINE.  EACH OVERLAY SEGMENT
                        ; IS CALLED INTO CORE AS A UNIT AND MUST BE CONTIGUOUS IN CORE.  AN
                        ; OVERLAY SEGMENT MAY HAVE ANY NUMBER OF ENTRY POINTS, TO THE LIMITS
                        ; OF CORE MEMORY.  ONLY ONE SEGMENT AT A TIME MAY OCCUPY AN OVERLAY REGION.
                        ;
                        ; THERE IS ONE WORD PREFIXED TO EVERY OVERLAY REGION THAT IDENTIFIES THE
                        ; SEGMENT CURRENTLY RESIDENT IN THAT OVERLAY REGION.  THIS WORD IS AN INDEX
                        ; INTO THE OVERLAY TABLE AND POINTS AT THE OVERLAY SEGMENT INFORMATION.
                        ;
                        ; UNDEFINED GLOBALS IN THE OVERLAY HANDLER MUST BE NAMED "$OVDF1" TO
                        ; "$OVDFn" SUCH THAT A RANGE CHECK MAY BE DONE BY LINK TO DETERMINE IF
                        ; THE UNDEFINED GLOBAL NAME IS FROM THE OVERLAY HANDLER.  A CHECK IS
                        ; DONE ON THE .RAD50 CHARACTERS "$OV", AND THEN A RANGE CHECK IS DONE ON
                        ; THE .RAD50 CHARATERS "DF1" TO "DFn".  THESE GLOBAL SYMBOLS DO NOT APPEAR
                        ; ON LINK MAPS, SINCE THEIR VALUE IS NOT KNOWN UNTIL AFTER THE MAP HAS BEEN
                        ; PRINTED.  CURRENTLY $OVDF1 TO $OVDF5 ARE IN USE.
                        ;
                        ; GLOBAL SYMBOLS O$READ AND O$DONE ARE USEFUL WHEN DEBUGGING OVERLAID
                        ; PROGRAMS.
                        ;
                        ;       O$READ:: WILL APPEAR IN THE LINK MAP AND LOCATES THE .READW
                        ;       STATEMENT IN THE OVERLAY HANDLER.
                        ;
                        ;       O$DONE:: WILL APPEAR IN THE LINK MAP AND LOCATES THE FIRST
                        ;       INSTRUCTION AFTER THE .READW IN THE OVERLAY HANDLER.
                        ;-
```

THE RUN-TIME OVERLAY HANDLER

```
                        .MCALL  .READW,..V1..
000000                          ..V1..          ;V1 FORMAT

                        .SBTTL  $OVTAB  OVERLAY TABLE

000000                  .PSECT  $OTABL,D,GBL,OVR

                        ;+
                        ; OVERLAY TABLE STRUCTURE:
                        ;
                        ; LOC 64 ->    $OVTAB:
                        ;               .WORD   <CORE ADDR>,<RELATIVE BLK>,<WORD COUNT>   /0 OVERLAYS
                        ;               DUMMY SUBROUTINES FOR ALL OVERLAY SEGMENTS
                        ;-

000000                  $OVTAB:
```

```
                                .SBTTL  OVERLAY HANDLER CODE

000000                          .PSECT  $OHAND,GBL
000000                          .PSECT  ZOHAND,GBL,OVR

                                .ENABL  LSB

                                ; $OVRH IS THE ENTRY POINT TO THE OVERLAY HANDLER

000000  060502                          .RAD50  /OVR/           ;THIS KEEPS HANDLER THE SAME SIZE AS V03
000002  010046                  $OVRH:: MOV     R0,-(SP)        ;/O OVERLAY ENTRY POINT
000004  010146                          MOV     R1,-(SP)        ;SAVE REGISTERS
000006  010246                          MOV     R2,-(SP)

000010                          1$:
000010  000422                          BR      5$              ;FIRST CALL ONLY * * *
                                ;       MOV     @R5,R1          ;PICK UP OVERLAY NUMBER
000012  062701  177772'                  ADD     #$OVTAB-6,R1    ;CALC TABLE ADDR
000016  012102                          MOV     (R1)+,R2        ;GET FIRST ARG. OF OVERLAY SEG. ENTRY
000020  022512                  2$:     CMP     (R5)+,@R2       ;IS OVERLAY ALREADY RESIDENT?
000022  001406                          BEQ     3$              ;YES, BRANCH TO IT

                                ;+
                                ; THE .READW ARGUMENTS ARE AS FOLLOWS:
                                ; CHANNEL NUMBER, CORE ADDRESS, LENGTH TO READ, RELATIVE BLOCK ON DISK.
                                ; THESE ARE USED IN REVERSE ORDER FROM THAT SPECIFIED IN THE CALL.
                                ;-

000024                          O$READ:: .READW 17,R2,@R1,(R1)+ ;READ FROM OVERLAY FILE
000036  103405                  O$DONE:: BCS    4$
000040  012602                  3$:     MOV     (SP)+,R2        ;RESTORE USERS REGISTERS
000042  012601                          MOV     (SP)+,R1
000044  012600                          MOV     (SP)+,R0
000046  011505                          MOV     @R5,R5          ;GET ENTRY ADDRESS
000050  000205                          RTS     R5              ;ENTER OVERLAY ROUTINE AND RESTORE USER'S R5

000052  104376                  4$:     EMT     376             ;SYSTEM ERROR 10 (OVERLAY I/O)
000054     000      373                 .BYTE   0,373

000056  012767  011501  177724  5$:     MOV     #11501,1$       ;RESTORE SWITCH INSTR (MOV @R5,R1)
000064  016701  000012                  MOV     $ODF1,R1        ;START ADDR FOR CLEAR OPERATION
000070  005021                  6$:     CLR     (R1)+           ;CLEAR ALL OVERLAY REGIONS
000072  020167  000006                  CMP     R1,$ODF2        ;DONE?
000076  103774                          BLO     6$              ;LO -> NO, REPEAT
000100  000743                          BR      1$              ;AND RETURN TO CALL IN PROGRESS

000102  000000G                 $ODF1:: .WORD   $OVDF1           ;HIGH ADDR OF ROOT SEGMENT + 2 (NXT AVAIL)
000104  000000G                 $ODF2:: .WORD   $OVDF2           ;HIGH ADDRESS OF /O OVERLAYS +2 (NXT AVAIL)

                                .DSABL  LSB

        000001                  .END
Errors detected:  0

*** Assembler statistics
```

OVERLAY HANDLER CODE


```
Work  file  reads:  0
Work  file writes:  0
Size of work file:  10442 Words  ( 41 Pages)
Size of core pool:  18176 Words  ( 71 Pages)
Operating  system:  RT-11

Elapsed time:  00:00:07.25
,LP:OHANDL=DK:OHANDL/C/N:SYM
```

Cross reference table (CREF V05.05)

```
$ODF1     4-42      4-48#
$ODF2     4-44      4-49#
$OVDF1    4-48
$OVDF2    4-49
$OVRH     4-12#
$OVTAB    3-17#     4-19
...V1     3-2#      4-30      4-30
...V2     1-2       1-2       1-2       1-2       1-2       1-2       1-2       1-2       1-2       1-2       1-2       1-2
          1-2       1-2       1-2       1-2       1-2       1-2       1-2       1-2#      1-2#      1-2#      1-2#      1-2#
          1-2#      1-2#
...V27    1-2       1-2
...V28    1-2
...V5     1-2#
...V6     1-2       1-2       1-2       1-2       1-2       1-2       1-2       1-2       1-2#      1-2#      1-2#      1-2#
          1-2#      1-2#      1-2#      1-2#
.OHAND    1-2#
O$DONE    4-31#
O$READ    4-30#
```

Cross reference table (CREF V05.05)

```
...CM0    3-2#      4-30      4-30      4-30
...CM1    3-2#
...CM2    3-2#
...CM3    3-2#
...CM4    3-2#
...CM5    3-2#      4-30
...CM6    3-2#
...CM7    3-2#      4-30
..V1..    3-1#      3-2
.AUDIT    1-2#
.MODUL    1-1#      1-2
.NLCSI    1-2#
.READW    3-1#      4-30
.RMODU    1-2#
```