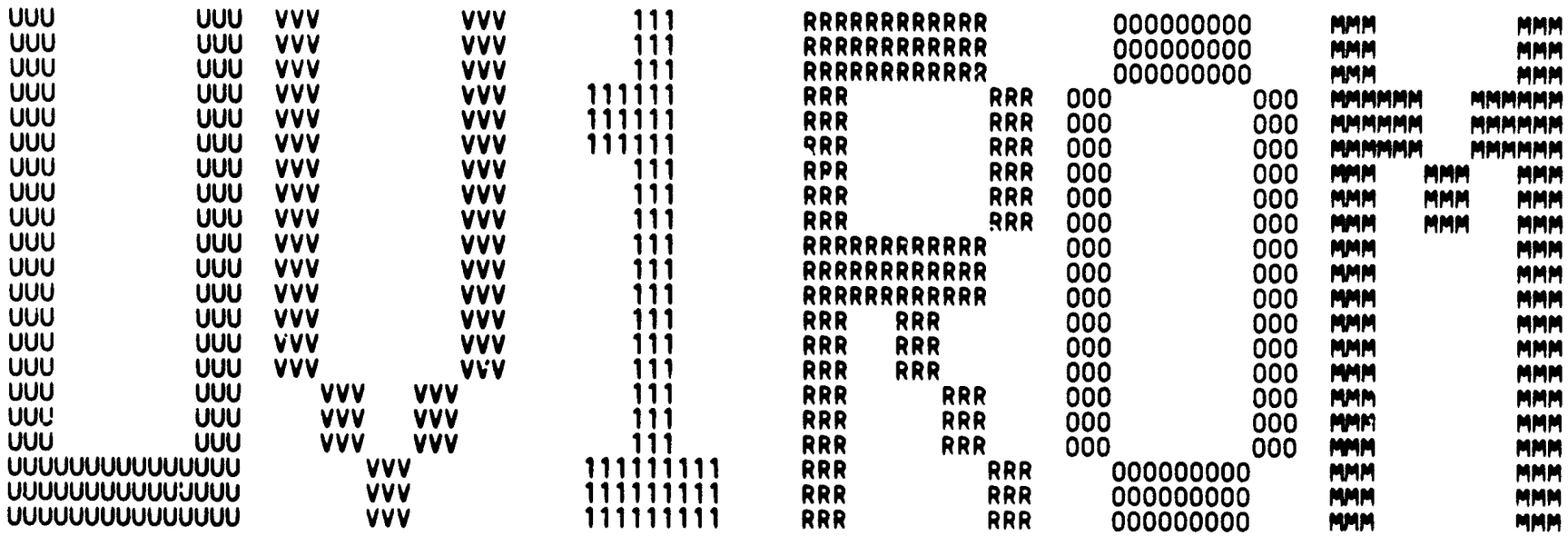
UUU UUU VVV VVV 111 RRRRRRRRRRR 000000000 MMM MMM
UUU UUU VVV VVV 111 RRRRRRRRRRR 000000000 MMM MMM
UUU UUU VVV VVV 111 RRRRRRRRRRR 000000000 MMM MMM
UUU UUU VVV VVV 111111 RRR RRR 000 000 MMMMMM MMMMMM
UUU UUU VVV VVV 111111 RRR RRR 000 000 MMMMMM MMMMMM
UUU UUU VVV VVV 111111 RRR RRR 000 000 MMMMMM MMMMMM
UUU UUU VVV VVV 111 RRR RRR 000 000 MMM MMM MMM
UUU UUU VVV VVV 111 RRR RRR 000 000 MMM MMM MMM
UUU UUU VVV VVV 111 RRR RRR 000 000 MMM MMM MMM
UUU UUU VVV VVV 111 RRRRRRRRRRR 000 000 MMM MMM
UUU UUU VVV VVV 111 RRRRRRRRRRR 000 000 MMM MMM
UUU UUU VVV VVV 111 RRRRRRRRRRR 000 000 MMM MMM
UUU UUU VVV VVV 111 RRR RRR 000 000 MMM MMM
UUU UUU VVV VVV 111 RRR RRR 000 000 MMM MMM
UUU UUU VVV VVV 111 RRR RRR 000 000 MMM MMM
UUU UUU VVV VVV 111 RRR RRR 000 000 MMM MMM
UUU UUU VVV VVV 111 RRR RRR 000 000 MMM MMM
UUU UUU VVV VVV 111 RRR RRR 000 000 MMM MMM
UUU UUU VVV VVV 111 RRR RRR 000 000 MMM MMM
UUUUUUUUUUUUUUU VVV 111111111 RRR RRR 000000000 MMM MMM
UUUUUUUUUUUUUUU VVV 111111111 RRR RRR 000000000 MMM MMM
UUUUUUUUUUUUUUU VVV 111111111 RRR RRR 000000000 MMM MMM

```
VV      VV  MM      MM  BBBBBBBB    UU      UU  VV      VV    AAAAAA    XX      XX      11
VV      VV  MM      MM  BBBBBBBB    UU      UU  VV      VV    AAAAAA    XX      XX      11
VV      VV  MMMM  MMMM  BB      BB  UU      UU  VV      VV  AA      AA  XX      XX    1111
VV      VV  MMMM  MMMM  BB      BB  UU      UU  VV      VV  AA      AA  XX      XX    1111
VV      VV  MM  MM  MM  BB      BB  UU      UU  VV      VV  AA      AA    XX  XX        11
VV      VV  MM  MM  MM  BB      BB  UU      UU  VV      VV  AA      AA    XX  XX        11
VV      VV  MM      MM  BBBBBBBB    UU      UU  VV      VV  AA      AA      XX          11
VV      VV  MM      MM  BBBBBBBB    UU      UU  VV      VV  AA      AA      XX          11
VV      VV  MM      MM  BB      BB  UU      UU  VV      VV  AAAAAAAAAA    XX  XX        11
VV      VV  MM      MM  BB      BB  UU      UU  VV      VV  AAAAAAAAAA    XX  XX        11
  VV  VV    MM      MM  BB      BB  UU      UU    VV  VV    AA      AA  XX      XX      11          ....
  VV  VV    MM      MM  BB      BB  UU      UU    VV  VV    AA      AA  XX      XX      11          ....
    VV      MM      MM  BBBBBBBB    UUUUUUUUUU      VV      AA      AA  XX      XX  111111          ....
    VV      MM      MM  BBBBBBBB    UUUUUUUUUU      VV      AA      AA  XX      XX  111111          ....
```

```
  CCCCCCCC    000000    MM      MM
  CCCCCCCC    000000    MM      MM
CC          OO      OO  MMMM  MMMM
CC          OO      OO  MMMM  MMMM
CC          OO      OO  MM  MM  MM
CC          OO      OO  MM  MM  MM
CC          OO      OO  MM      MM
CC          OO      OO  MM      MM
CC          OO      OO  MM      MM
CC          OO      OO  MM      MM
CC          OO      OO  MM      MM
CC          OO      OO  MM      MM
  CCCCCCCC    000000    MM      MM
  CCCCCCCC    000000    MM      MM
```

```
$! This command procedure builds a floppy disk which can be used
$! on MicroVAX-I to boot from disks with floating CSR's.
$!      P1      system disk to be booted
$!      P2      target device, intermediary RX50
$!
$ on error then goto cleanup_error
$ on controly then goto cleanup_ctrl
$ say :== write sys$output
$ intermediate_device = ""
$!
$! which device to boot?
$!
$ return := get_device
$ device = P1
$ if P1 .nes. "" then goto chkdev
$get_device:
$ say ""
$ inquir/nopunc device "What system disk do you want to be able to boot from?  "
$chkdev:
$ if f$length(device) .ne. 4 then goto bad_device
$!
$! copy VMBUVAX1P.EXE to directory ard rename to SCRATCHVMB.EXE
$!
$ copy sys$system:vmbuvax1p.exe scratchvmb.exe
$!
$! create file to patch vmbuvax1p.exe, overuriting location
$! patch_device_name with the name of the system disk.
$!
$ say ""
$ say ""
$ say ""
$ say ""
$ say "       Apply patch to create new SYSBOOT.EXE"
$ say ""
$ open/write dataf patchfile.com                        ! open patch file
$ temp[0,16] = %x2727                                   ! temp = ''
$ device = """" + device + """"                         ! device = 'device'
$ write dataf "$ patch/absolute scratchvmb.exe"         ! create patch
$ write dataf "replace/ascii"                           ! do a replace
$ write dataf "8"                                       ! address of patch_device_name
$ write dataf temp                                      ! old = ''
$ write dataf "exit"
$ write dataf device                                    ! new = device
$ write dataf "exit"
$ write dataf "update"                                  ! update VMB.EXE
$ write dataf "exit"
$ close dataf
$ @patchfile                                            ! invoke patch
$!
$! where is the intermediate boot media mounted
$!
$ return := get_intermediate
$ intermediate_device = P2
$ if P2 .nes. "" then goto chkintermediate
$get_intermediate:
$ say ""
$ inquire/nopunc intermediate_device -
```

```
    'Where will the intermediate RX50 media be mounted? "
$ask:
$ inquire/nopunc ok "Are you ready? "
$ if .not. ok then goto ask
$chkintermediate:
$ if f$length(intermediate_device) .ne. 4 then goto bad_device
$ say ""
$!
$! ready intermediate device
$!
$ if f$getdvi(intermediate_device,"MNT") then dismount/nounload 'intermediate_device
$ initialize 'intermediate_device' INTERMEDIATE
$ mount/over=ident 'intermediate_device
$!
$! Place 2 files on the intermediate media.  The first is SCRATCHVMB.EXE,
$! which we call [SYS0.SYSEXE]SYSBOOT.EXE.  When the primary bootstrap
$! VMB executes, it loads the file SYSBOOT.EXE into memory and branches
$! to it.  In the case of the intermediate VMB, SYSBOOT.EXE is actually
$! a modified copy of VMB which allows you to boot from devices with
$! floating CSR's.  It in turn finds SYSBOOT.EXE on the real system
$! disk and branches to it.  The 2nd file we place on the intermediate
$! VMB is an unpatched version of VMBUVAX1P.EXE which prompts the console
$  at boot time for the name of the system disk.  To use this version,
$! enter B/100 ddcu and enter [SYS0.SYSEXE]SYSBOOTP.EXE as the name
$! of the boot file.
$!
$ create/directory 'intermediate_device':[sys0.sysexe]
$ copy scratchvmb.exe 'intermediate_device':[sys0.sysexe]sysboot.exe
$ copy sys$system:vmbuvax1p.exe 'intermediate_device':[sys0.sysexe]sysbootp.exe
$ device = device - """ - """
$ say ""
$ say "The media on ",intermediate_device," may now be used to boot device ",device
$ say ""
$ goto cleanup
$!
$! bad device name, inquire again
$!
$bad_device:
$ say ""
$ say "** You entered an invalid device name **"
$ goto 'return
$!
$! error handlers
$!
$cleanup_error:
$ say ""
$ say "** Unexpected error - creation of intermediate RX50 media unsuccessful **"
$ goto cleanup
$cleanup_ctrl:
$ say ""
$ say "** Unexpected halt due to control_y - creation of intermediate RX50 media unsuccessful **"
$!
$! cleanup directory, dismount intermediate device
$!
$cleanup:
$ if intermediate_device .eqs. "" then goto delete_files
```

```
$ if f$getdvi(intermediate_device,"MNT") then dismount/unload 'intermediate_device'
$delete_files:
$ temp= f$search("patchfile.com")
$ if temp .nes. "" then delete patchfile.com;*
$ temp = f$search("scratchvmb.exe")
$ if temp .nes. "" then delete scratchvmb.exe;*
$ temp = f$search("scratchvmb.jnl")
$ if temp .nes. "" then delete scratchvmb.jnl;*
$ exit
```

0452 AH-EF71A-SE
VAX/VMS V4.1 SRC LST MCRF UPD