.REM _

IDENTIFICATION

PRODUCT CODE: AC-E676G-MC

PRODUCT NAME: CXRKAG0 DEC/X11 RK11 MODULE

DATE: SEPTEMBER 1978

MAINTAINER: DECX11 SUPPORT GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973,1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

RKA IS AN IOMODX THAT EXERCISES RK02, RK03, RK04, RK05 DISK DRIVES ON AN RK11 CONTROLLER. IT EXERCISES THE DRIVES BY DOING WRITES, WRITE-CHECKS, READS, AND IN-CORE COMPARISONS. ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE TTY.

2. REQUIREMENTS

HARDWARE: 1 TO 8 RK DISK DRIVES WITH AN RK11 CONTROLLER

STORAGE:: RKA REQUIRES:
1. DECIMAL WORDS: 1057
2. OCTAL WORDS: 02041
3. OCTAL BYTES: 4102

3. PASS DEFINITION

ONE PASS OF THE RKA MODULE CONSISTS OF 512 CYCLES OF THE BASIC TEST SEQUENCE (WRITE, WRITE-CHECK, READ, DATA-CHECK). THE TEST SEQUENCE WRITES 1024 WORDS, WRITE-CHECKS SAME, READS THE FIRST 256 WORDS, AND DATA-CHECKS SAME.

4. EXECUTION TIME

ONE PASS OF RKA RUNNING ALONE ON A PDP-11/40 TAKES APPROXIMATELY 1 MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 177400, VECTOR: 220, BR1: 5, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP

MAKE CERTAIN THAT ALL DRIVES ARE POWERED UP, WRITE ENABLED, AND READY

7. MODULE OPERATION

TEST SEQUENCE:

A. SETUP DEVICE REGISTER ADDRESSES AND MODULE VARIABLES
B. RESET ALL DRIVES ON-LINE AND DROP ALL THAT ARE NOT
C. GET A DISK ADDRESS AND A FRESH BLOCK OF DATA
D. GET A DRIVE ADDRESS
E. DO A WRITE -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
F. DO A WRITE-CHECK -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
G. DO A READ -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
H. DO A DATA-CHECK -- IF ERRORS, REPORT AND CONTINUE
I. IF END OF PASS, REPORT AND GO TO C
J. IF END OF DRIVES, GO TO C ELSE GO TO D

8. OPERATION OPTIONS

SR1 BIT 0 SET(1):
IF THE RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, A HARD ERROR IS ASSUMED AND THE DRIVE IS DROPPED

SR1 BIT 0 CLEAR(0):
IF THE RETRY LIMIT IS EXCEEDED, THE FUNCTION IS ABORTED AND THE TESTING CONTINUES

SR1 BIT 2 SET(1):
WILL NOT TYPE OUT DATA LATE ERRORS BUT WILL KEEP TRACK OF THE NUMBER OF DATA LATE ERRORS

SR1 BIT 2 CLEAR(0):
TYPE OUT DATA LATE ERRORS AND KEEP TRACK OF THE NUMBER OF DATA LATE ERRORS IN "DLTCNT"

9. NON-STANDARD PRINTOUTS

A. MOST PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN THE DEC/X11 DOCUMENT

B. ERROR MESSAGES DUMP THE CONTENTS OF THE 8 RK11 REGISTERS IN THE FOLLOWING ORDER:

RKDS RKER RKCS RKWC RKBA RKDA RKMR RKDB

```
000000'  -       IOMODX  <RKAG  >,177400,220,5,0,0,512.,5,BUFIN,256.,1024.
000000'          MODULE  150000,RKAG  ,177400,220,5,0,0,512.,5,BUFIN,256.,1024.
                 .TITLE  RKAG DEC/X11 SYSTEM EXERCISER MODULE
         ;       DDXCOM  VERSION  6       23-MAY-78
                                 .LIST   BIN
                                ;*****************************************************************************
000000'                         BEGIN:
000000' 045522  043501     040  MODNAM: .ASCII  /RKAG / ;MODULE NAME.
000005'     000                 XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 177400                  ADDR:   177400+0                ;1ST DEVICE ADDR.
000010' 000220                  VECTOR: 220+0                   ;1ST DEVICE VECTOR.
000012'     240                 BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013'     000                 BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014' 000001                  DVID1:  0+1                     ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN                    ;SWITCH REGISTER 4
                                ;*****************************************************************************
000026' 150000                  STAT:   150000                  ;STATUS WORD.
000030' 000252'                 INIT:   START                   ;MODULE START ADDR.
000032' 000252'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                       ;PASS COUNTER.
000036' 001000                  ICONT:  512.                    ;# OF ITERATIONS PER PASS=512.
000040' 000000                  ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042' 000000                  SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                  HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                  SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                  HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                  SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054' 000000                  RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:                         ;RESERVED FOR MONITOR USE
000056' 000000                  RES1:   0                       ;RESERVED FOR MONITOR USE
000060' 000000                  RES2:   0                       ;RESERVED FOR MONITOR USE
000062' 000000                  SVR0:   OPEN                    ;LOC TO SAVE R0.
000064' 000000                  SVR1:   OPEN                    ;LOC TO SAVE R1.
000066' 000000                  SVR2:   OPEN                    ;LOC TO SAVE R2.
000070' 000000                  SVR3:   OPEN                    ;LOC TO SAVE R3.
000072' 000000                  SVR4:   OPEN                    ;LOC TO SAVE R4.
000074' 000000                  SVR5:   OPEN                    ;LOC TO SAVE R5.
000076' 000000                  SVR6:   OPEN                    ;LOC TO SAVE R6.
000100' 000000                  CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102'                         SBADR:                          ;ADDR OF GOOD DATA, OR
000102' 000000                  ACSR:   OPEN                    ;CONTENTS OF CSR.
000104'                         WABADR:                         ;ADDR OF BAD DATA, OR
000104' 000000                  ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106'                         ERRTYP:                         ;TYPE OF ERROR
000106' 000000                  ASR:    OPEN                    ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN                    ;ACTUAL DATA.
000112' 000456'                 RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122' 000005                  IDNUM:  5                       ;MODULE IDENTIFICATION NUMBER=5
```



```
000124' 002614'                 RBUFVA: BUFIN                   ;READ BUFFER VIRTUAL ADDRESS
000126' 000000                  RBUFPA: OPEN                    ;READ BUFFER PHYSICAL ADDRESS
000130' 000000                  RBUFEA: OPEN                    ;READ BUFFER EA BITS
000132' 000400                  RBUFSZ: 256.                    ;SIZE OF THE READ BUFFER
000134' 000000                  WBUFPA: OPEN                    ;WRITE BUFFER PHYSICAL ADDRESS
000136' 000000                  WBUFEA: OPEN                    ;WRITE BUFFER EA BITS
000140' 002000                  WBUFRQ: 1024.                   ;WRITE BUFFER SIZE REQUESTED
000142' 000000                  WBUFSZ: OPEN                    ;WRITE BUFFER SIZE AVAILABLE
000144' 000000                  CDERCT: OPEN                    ;CDATA/DATCK ERROR COUNT
000146' 000000                  CDWDCT: OPEN                    ;CDATA/DATCK WORD COUNT
000150' 000000                  FREE:   OPEN                    ;RESERVED FOR FUTURE USE
        000040                          .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                        .NLIST
                                        .WORD   0
                                        .LIST
                                        .ENDR
000252'                         MODSP:
                                ;*****************************************************************************
```

```
                    222
000252' 005067  002300          START:  CLR     CNT             ; ZERO END OF PASS TESTER
000256' 012767  000400  177630          MOV     #256.,WDTO      ; WORDS TO MEM
000264' 012767  002000  177624          MOV     #1024.,WDFR     ; WORDS FROM MEM
000272' 012767  000003  177620          MOV     #3,INTR         ; # OF INTERRUPTS/ITERATION
000300' 005067  003570                  CLR     SIDE            ; CLEAR FLAGS AND SIDE INDICATOR
000304' 005067  002244                  CLR     DLTCNT          ; CLEAR DATA LATE ERROR COUNTER
000310' 016767  177500  002250          MOV     DVIDI,DVICE     ; GET DRIVE INDICATOR
000316' 016767  002244  002244          MOV     DVICE,DRIVE     ; ALSO SAVE IT IN DRIVE
000324' 012767  177775  002244          MOV     #-3,BLK1        ; INITIALIZE BLOCK COUNTER
000332' 005067  002234                  CLR     DRVVE           ; ZERO UNIT NUMBER
000336' 012767  160000  002230          MOV     #160000,DRVSFT  ; INITIALIZE THE SHIFTED DRIVE #
000344' 122737  000002  000041          CMPB    #BIT1,@#41      ; IS RK UNIT 0 THE LOAD MEDIUM ?
000352' 001026                          BNE     3$              ; NO, CONTINUE
000354' 012702  000000                  MOV     #0,R2           ; INITIALIZE DRIVE COUNT
000360' 113700  000040                  MOVB    @#40,R0         ; GET LOAD MEDIUM COUNT
000364' 012701  000001                  MOV     #1,R1           ; LOAD UP R1 TO POINT TO DRIVE #0
000370' 105700          1$:     TSTB    R0              ;IF R0 EQUAL TO 0 THEN
000372' 001404                  BEQ     2$              ;GO TO 2$
000374' 006301                  ASL     R1              ;ELSE UPDATE DRIVE POINTER
000376' 105300                  DECB    R0              ;DECREMENT COUNT
000400' 005202                  INC     R2              ; UPDATE DRIVE NUMBER
000402' 000772                  BR      1$              ;TRY AGAIN
000404' 130167  002156          2$:     BITB    R1,DVICE        ;IF DRIVE NOT SELECTED TO BE TESTED THEN
000410' 001407                  BEQ     3$
000412' 010267  002154                  MOV     R2,DRVVE        ;ELSE LOAD DRIVE ADDRESS TO BE DROPPED
000416' 004767  000724                  JSR     PC,DROP         ;GO DROP IT
000422' 104403  000000' 004050'         MSGNS,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
                    250
000430' 012767  177777  002134  3$:     MOV     #-1,DRVVE       ; INITIALIZE DRIVE COUNTER
000436' 004767  001572                  JSR     PC,SETUP        ; GENERATE REGISTER ADDRESSES
000442' 004767  001770                  JSR     PC,REZET        ; INITIALIZE RK REGS. AND ALL DRIVES
000446' 005767  002114                  TST     DVICE           ; DROP THE MODULE ?
000452' 001536                  BEQ     FINI            ; YES
000454' 000404                  BR      RSTRT1          ;+     / THIS IS
000456' 005767  002074          RESTRT: TST     CNT             ;+    /  SUPPORT
000462' 001001                  BNE     RSTRT1          ;+   /    FOR
000464' 000672                  BR      START           ;+ /      DT03
000466'                 RSTRT1:                         ;+/       BUS SWITCH
000466' 104415  000000' 000124'         GETPAS,BEGIN, RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
000474' 016767  177432  002106          MOV     RBUFSZ,WCNT2    ; SAVE READ BUFFER SIZE
000502' 005467  002102                  NEG     WCNT2           ; GET THE 2'S COMPLEMENT
                    264
000506' 004767  000672          STRT:   JSR     PC,BLOCK        ; GET NEXT BLOCK NUMBER
000512' 104414  000000'                 GWBUFS, BEGIN           ;GET WRITE BUFFER INFORMATION
000516' 016767  177420  002062          MOV     WBUFSZ,WCNT1    ; SAVE WRITE BUFFER SIZE
000524' 005467  002056                  NEG     WCNT1           ; GET THE 2'S COMPLEMENT
000530' 016700  002042                  MOV     BLK1,R0         ; LOAD BLOCK # FOR CONVRT
000534' 004767  001574                  JSR     PC,CONVRT       ; GENERATE DISK ADR. FROM BLOCK #
                    271
000540' 004767  001016          NEXT:   JSR     PC,DRVADR       ; GET A DRIVE ADDRESS
000544' 005767  002016                  TST     DVICE           ; ANY DRIVES LEFT ?
000550' 001477                  BEQ     FINI            ; NO, GO DROP THE MODULE
000552' 132767  000010  003315          BITB    #BIT3,FLAG      ; ALL DRIVES DONE ?
000560' 001352                  BNE     STRT            ; YES, GO GET ANOTHER BLOCK
000562' 042767  160000  001774          BIC     #160000,DSKADR  ; CLEAR DRIVE ADDRESS
```



```
000570' 056767  002000  001766          BIS     DRVSFT,DSKADR   ; LOAD DRIVE ADDRESS
000576' 016777  001762  003024          MOV     DSKADR,@RKDA    ; LOAD DISK ADDRESS
000604' 032777  000040  003014          BIT     #BIT5,@RKDS     ; WRITE PROTECTED ?
000612' 001406                          BEQ     1$              ; NO, CONTINUE
000614' 004767  000526                  JSR     PC,DROP         ; YES, DROP THE DRIVE
000620' 104403  000000' 004050'         MSGNS,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
000626' 000744                  BR      NEXT            ; GO ON TO NEXT DRIVE
000630' 032777  000100  002760  1$:     BIT     #BIT6,@RKDS     ; DRIVE READY ?
000636' 001003                  BNE     2$              ; YES, CONTINUE
000640' 004767  001010                  JSR     PC,NOTRDY       ; NO, WAIT FOR READY
000644' 000720                  BR      STRT            ; TRY AGAIN
000646' 005067  003224          2$:     CLR     TRY1            ; ZERO RETRY COUNTERS
000652' 105067  003222                  CLRB    TRY3            ;
```

```
                                ;
000656' 004567  000212          GO:     JSR     R5,WRITE        ; WRITE SOME DATA
000662' 000434                          BR      RETRY1          ; IF ERRORS, TRY IT AGAIN
000664' 132767  000004  003203          BITB    #BIT2,FLAG      ; DID THE DISK OVERFLOW ?
000672' 001407                          BEQ     GOA             ; NO, CONTINUE
000674' 142767  000004  003173          BICB    #BIT2,FLAG      ; YES, CLEAR THE OVERFLOW FLAG
000702' 012767  177775  001666          MOV     #-3,BLK1        ; RESET THE BLOCK NUMBER
000710' 000676                          BR      STRT            ; START OVER AT BEGINNING OF DISK
000712' 004567  000210          GOA:    JSR     R5,WRITCK       ; WRITE-CHECK THE DATA
000716' 000430                          BR      RETRY2          ; IF ERRORS, TRY AGAIN
000720' 004567  000234          GOB:    JSR     R5,READ         ; READ THE DATA WRITTEN
000724' 000437                          BR      RETRY3          ; IF ERRORS, TRY AGAIN
000726' 104412  000000' 000126'         CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
000734' 000736'                         .+2                     ; IF ERROR, CONTINUE
                                ;
                                ;
000736' 005267  001514          PASS:   INC     CNT             ; COUNT A CYCLE
000742'                                 ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
000742' 104413  000000'                                         ;MONITOR SHALL TEST END OF PASS
000746' 000674                          BR      NEXT
                                ;
000750'                         FINI:
000750' 104410  000000'                 ENDS,BEGIN              ; DROP THE MODULE
                                ;       ------------------------
                                ;
                                ;
                                ;
000754' 105267  003116          RETRY1: INCB    TRY1            ; COUNT THE RETRYS
000760' 122767  000003  003110          CMPB    #3,TRY1         ; LIMIT EXCEEDED ?
000766' 001333                          BNE     GO              ; NO, GO TRY IT AGAIN
000770' 104403  000000' 004026'         MSGNS,BEGIN,EXCED1      ;ASCII MESSAGE CALL WITH COMMON HEADER
000776' 000423                          BR      NEXTA           ; GO ON TO NEXT DRIVE
                                ;       ------------------------
001000' 105267  003073          RETRY2: INCB    TRY2            ; COUNT RETRYS
001004' 122767  000003  003065          CMPB    #3,TRY2         ; LIMIT EXCEEDED ?
001012' 001337                          BNE     GOA             ; NO, TRY AGAIN
001014' 104403  000000' 004034'         MSGNS,BEGIN,EXCED2      ;ASCII MESSAGE CALL WITH COMMON HEADER
001022' 000411                          BR      NEXTA           ; GO ON TO NEXT DRIVE
                                ;       ------------------------
001024' 105267  003050          RETRY3: INCB    TRY3            ; COUNT RETRYS
001030' 122767  000003  003042          CMPB    #3,TRY3         ; LIMIT EXCEEDED ?
001036' 001330                          BNE     GOB             ; NO, GO TRY AGAIN
001040' 104403  000000' 004042'         MSGNS,BEGIN,EXCED3      ;ASCII MESSAGE CALL WITH COMMON HEADER
                                ;       ------------------------
001046' 032767  000001  176742  NEXTA:  BIT     #BIT0,SR1       ; DROP THE DRIVE ?
001054' 001405                          BEQ     1$              ; NO, SKIP TO NEXT DRIVE
001056' 004767  000264                  JSR     PC,DROP         ; YES, DROP OFFENDING DRIVE
001062' 104403  000000' 004050'         MSGNS,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
001070' 000167  177444          1$:     JMP     NEXT            ; GO ON TO NEXT DRIVE
                                ;       ------------------------
```



```
                                ;               -----  RK11 DISK DRIVERS  -----
001074' 012767  000503  001456  WRITE:  MOV     #503,FUNC       ; LOAD WRITE FUNCTION
001102' 016777  001500  002514          MOV     WCNT1,@RKWC     ; LOAD WORD COUNT
001110' 016777  177020  002510          MOV     WBUFPA,@RKBA    ; LOAD BUFFER ADDRESS
001116' 016767  177014  001436          MOV     WBUFEA,XMEM     ; LOAD EXTENDED MEMORY BITS
001124' 000462                          BR      GOGO            ; CONTINUE
001126' 012767  000507  001424  WRITCK: MOV     #507,FUNC       ; LOAD WRITE-CHECK FUNCTION
001134' 016777  001446  002462          MOV     WCNT1,@RKWC     ; LOAD WORD COUNT
001142' 016777  176766  002456          MOV     WBUFPA,@RKBA    ; LOAD BUFFER ADDRESS
001150' 016767  176762  001404          MOV     WBUFEA,XMEM     ; LOAD EXTENDED MEMORY BITS
001156' 000445                          BR      GOGO            ; CONTINUE
001160' 012767  000505  001372  READ:   MOV     #505,FUNC       ; LOAD READ FUNCTION
001166' 016777  001416  002430          MOV     WCNT2,@RKWC     ; LOAD WORD COUNT
001174' 016777  176726  002424          MOV     RBUFPA,@RKBA    ; LOAD BUFFER ADDRESS
001202' 016767  176722  001352          MOV     RBUFEA,XMEM     ; LOAD EXTENDED MEMORY BITS
001210' 000430                          BR      GOGO            ; CONTINUE
                                ;
001212' 012777  000001  002402  CLEAR:  MOV     #1,@RKCS        ; ISSUE A CONTROL RESET
001220' 004767  001254                  JSR     PC,WAIT1        ; GO WAIT FOR CONTROLLER READY
001224' 016777  001344  002376          MOV     DRVSET,@RKDA    ; RELOAD THE DRIVE ADDRESS
001232' 032777  000100  002356          BIT     #BIT6,@RKDS     ; DRIVE READY ?
001240' 001001                          BNE     2$              ; YES, CONTINUE
001242' 000205                          RTS     R5              ; NO, ABORT DRIVE RESET
001244' 012777  000015  002350  2$:     MOV     #15,@RKCS       ; ISSUE A DRIVE RESET
001252' 004767  000524                  JSR     PC,WAIT         ; GIVE IT TIME TO COMPLETE
001256' 012777  000001  002336          MOV     #1,@RKCS        ; ISSUE ANOTHER CONTROLLER RESET
001264' 004767  001210                  JSR     PC,WAIT1        ; WAIT FOR CONTROLLER READY
001270' 000205                          RTS     R5              ; RETURN
                                ;
001272' 012777  001326' 176510  GOGO:   MOV     #NTRUPT,@VECTOR ; SET INTERUPT ENTRY POINTER
001300' 016777  001260  002322          MOV     DSKADR,@RKDA    ; LOAD THE DISK ADDRESS
001306' 056767  001250  001244          BIS     XMEM,FUNC       ; LOAD EXTENDED MEMORY BITS
001314' 016777  001240  002300          MOV     FUNC,@RKCS      ; EXECUTE THE FUNCTION
001322' 104400  000000'                 EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
                                ;
001326'                         NTRUPT:
                                ;-------------------------------------------------------------------
001326' 000004  000000' 001334'         PIRQS,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                ;-------------------------------------------------------------------
                                ;
001334' 004567  000512  1$:     JSR     R5,ERRORS       ; GO CHECK FOR ERRORS
001340' 000205                  RTS     R5              ; ERRORS DETECTED, RETURN
001342' 005725                  TST     (R5)+           ; NO ERRORS, SKIP RETRY
001344' 000205                  RTS     R5              ; RETURN OK
                                ;       ------------------------
```

```
                                ;*****************************************************************
                                ;CONVERT DRVVE TO ASCII AND
                                ;STORE AT ADR1
001372' 104420  000000' 002572'         OTOAS,BEGIN,DRVVE,ADR1
001400' 004464'
                                ;*****************************************************************
001402' 000207                  RTS     PC              ; RETURN
                                ;       ------------------------

001404' 062767  000003  001164  BLOCK:  ADD     #3,BLK1         ; STEP TO NEXT BLOCK
001412' 022767  011277  001156          CMP     #4799.,BLK1     ; BLOCK LIMIT REACHED ?
001420' 100002                          BPL     1$              ; NO, CONTINUE
001422' 005067  001150                  CLR     BLK1            ; YES, RESET BLOCK #
001426' 016767  001144  001144  1$:     MOV     BLK1,BLK2       ; READ WHERE WRITE
001434' 000207                  RTS     PC              ; RETURN
                                ;       ------------------------


001436' 016700  001134  ROOM:   MOV     BLK1,R0         ; SAVE THE CURRENT BLOCK NUMBER
001442' 012701  004537          MOV     #2399.,R1       ; LOAD MAX. NUMBER OF BLOCK PER SIDE
001446' 005002                  CLR     R2              ; ZERO REG. 2
001450' 022706  004537          CMP     #2399.,R0       ; IS SIDE 0 DONE ?
001454' 002002                  BGE     1$              ; NO, CONTINUE
001456' 162700  004540          SUB     #2400.,R0       ; YES, NORMALIZE BLOCK # FOR SIDE 1
001462' 012767  004400  001112  1$:     MOV     #256.,BSIZ      ; HI DENSITY BLOCK SIZE
001470' 032777  004000  002120          BIT     #BIT11,@RKDS    ; HI DENSITY DRIVE ?
001476' 001002                          BNE     2$              ; YES, CONTINUE
001500' 006267  001076                  ASR     BSIZ            ; NO, SET TO 128 -- LO DENSITY
001504' 160001          2$:     SUB     R0,R1           ; GET # OF BLOCKS LEFT ON DISK
001506' 066702  001070          3$:     ADD     BSIZ,R2         ; GET TOTAL NUMBER OF WORDS LEFT
001512' 005301                          DEC     R1              ; ALL BLOCKS ADDED IN ?
001514' 003374                          BGT     3$              ; NO, KEEP ADDING
001516' 005702                          TST     R2              ; IS # OF WORDS LEFT ON DISK NEG. ?
001520' 100404                          BMI     4$              ; YES
001522' 005767  176414                  TST     WBUFSZ          ; IS TRANSFER SIZE NEG. ?
001526' 100411                          BMI     7$              ; YES
001530' 000403                          BR      5$              ; NO, GO COMPARE
001532' 005767  176404  4$:      TST     WBUFSZ          ; IS TRANSFER SIZE POS. ?
001536' 100003                          BPL     6$              ; YES
001540' 020267  176376  5$:      CMP     R2,WBUFSZ       ; WAS THERE ENOUGH ROOM FOR THE TRANSFER ?
001544' 002402                          BLT     7$              ; NO, RETURN OK
001546' 005725          6$:     TST     (R5)+           ; YES, MUST BE A REAL ERROR
001550' 000205                  RTS     R5              ; RETURN, ERROR
001552' 152767  000004  002315  7$:     BISB    #BIT2,FLAG      ; SET OVERFLOW FLAG
001560' 000205                  RTS     R5              ; RETURN OK
                                ;       ------------------------
```



```
001562' 005267  001004  DRVADR: INC     DRVVE           ; COUNT A DRIVE
001566' 062767  020000  001000          ADD     #BIT13,DRVSFT   ; DRIVE COUNT LINED UP WITH RKDA
001574' 142767  000010  002273          BICB    #BIT3,FLAG      ; CLEAR END OF DRIVES FLAG
001602' 022767  000010  000762          CMP     #8.,DRVVE       ; ALL DRIVES CHECKED ?
001610' 001404                          BEQ     1$              ; YES, GO FLAG END OF DRIVES
001612' 006267  000752                  ASR     DRIVE           ; NO, IS NEXT DRIVE CHOSEN ?
001616' 103361                          BCC     DRVADR          ; NO, GO TRY ANOTHER DRIVE
001620' 000207                  RTS     PC              ; RETURN

001622' 152767  000010  002245  1$:     BISB    #BIT3,FLAG      ; SET END OF DRIVES FLAG
001630' 012767  177777  000734          MOV     #-1,DRVVE       ; RESET DRIVE COUNTER
001636' 012767  160000  000730          MOV     #160000,DRVSFT  ; ZERO THE SHIFTED DRIVE #
001644' 016767  000716  000716          MOV     DVICE,DRIVE     ; RESTORE CHOSEN DRIVES
001652' 000207                  RTS     PC              ; RETURN
                                ;       ------------------------


001654' 012767  177777  000710  NOTRDY: MOV     #-1,DRVVE       ; START WITH FIRST DRIVE
001662' 012767  160000  000704          MOV     #160000,DRVSFT  ;
001670' 016767  000672  000672          MOV     DVICE,DRIVE     ; RESET DRIVE SELECT
001676' 004767  177660          1$:     JSR     PC,DRVADR       ; GET A DRIVE ADDRESS
001702' 132767  000010  002165          BITB    #BIT3,FLAG      ; ALL DRIVES CHECKED ?
001710' 001412                          BNE     2$              ; YES, RETURN
001712' 016777  000656  001710          MOV     DRVSFT,@RKDA    ; NO, LOAD NEXT DRIVE ADDRESS
001720' 032777  000100  001670          BIT     #BIT6,@RKDS     ; IS THIS DRIVE READY ?
001726' 001363                          BNE     1$              ; YES, CONTINUE
001730' 004767  000046                  JSR     PC,WAIT         ; NO, WAIT FOR IT
001734' 000760                          BR      1$              ; GO CHECK REST OF DRIVES
001736' 000207                  RTS     PC              ; RETURN
                                ;       ------------------------


001740' 014167  176142  ERSUB2: MOV     -(R1),ASB       ; LOAD THE DATA
001744' 010167  176132          MOV     R1,SBADR        ; LOAD ADDRESS OF DATA WRITTEN
001750' 014267  176134          MOV     -(R2),AWAS      ; LOAD THE DATA
001754' 010267  176124          MOV     R2,WASADR       ; LOAD ADDRESS OF DATA READ
001760' 005721                  TST     (R1)+           ; RESET REG. 1
001762' 005722                  TST     (R2)+           ; RESET REG. 2

001764' 016767  001632  176106  ERSUB1: MOV     RKCS,CSRA       ; LOAD ADR. OF CURRENT CSR
001772' 017767  001624  176102          MOV     @RKCS,ACSR      ; LOAD CONTENTS OF CURRENT CSR
002000' 000207                  RTS     PC              ; RETURN
                                ;       ------------------------
```

```
                                ;
                                ;
                                ;
002002' 012767  077777  001604  WAIT:   MOV     #77777,CLK      ; SET THE TIMER
002010'                         1$:
002010' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002014' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002020' 032777  000100  001570          BIT     #BIT6,@RKDS     ; DRIVE READY ?
002026' 001210                          BNE     2$              ; YES, RETURN
002030' 005367  001560                  DEC     CLK             ; NO, WAIT SOME MORE ?
002034' 001365                          BNE     1$              ; YES, WAIT
002036' 004767  177304                  JSR     PC,DROP         ; TIME-OUT, DROP THE DRIVE
002042' 104403  000000' 004050'         MSGN$,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
002050' 000207                  2$:     RTS     PC              ; RETURN
                                ;       ----------------------------
002052' 004767  177706          ERRORS: JSR     PC,ERSUB1       ; LOAD ERROR INFORMATION
002056' 032777  040000  001536          BIT     #BIT14,@RKCS    ; HARD ERROR ?
002064' 001006                          BNE     1$              ; YES, GO REPORT
002066' 032777  000003  001524          BIT     #3,@RKER        ; SOFT ERROR ?
002074' 001041                          BNE     3$              ; YES, GO REPORT
002076' 005725                          TST     (R5)+           ; NO, SKIP RETRY
002100' 000205                          RTS     R5              ; RETURN OK
002102' 032777  040000  001510  1$:     BIT     #BIT14,@RKER    ; DISK OVERFLOW ?
002110' 001403                          BEQ     7$              ; NO, CONTINUE
002112' 004567  177320                  JSR     R5,ROOM         ; YES, IS IT A REAL ERROR ?
002116' 000442                          BR      5$              ; NO, CONTINUE
002120' 032777  001000  001472  7$:     BIT     #BIT9,@RKER     ;DATA LATE ERROR?
002126' 001411                          BEQ     2$              ;NO
002130' 005267  000420                  INC     DLTCNT          ;INCREMENT ERROR COUNTER
002134' 032767  000004  175654          BIT     #BIT2,SR1       ;TYPE OUT ERROR?
002142' 001013                          BNE     6$              ;NO
002144' 104403  000000' 004060'         MSGN$,BEGIN,DLTERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
002152'                         2$:
002152' 104403  000000' 004016'         MSGN$,BEGIN,HARD        ;ASCII MESSAGE CALL WITH COMMON HEADER
002160' 005067  175722                  CLR     ERRTYP          ;*****************************************************************
                                        ;
002164' 104405  000000' 003616'         HRDER$,BEGIN,TABLE      ;*****************************************
                                        ;
002172' 004567  177014          6$:     JSR     R5,CLEAR        ; GO CLEAR OUT ERRORS
002176' 000411                          BR      4$              ; RETURN
002200'                         3$:
002200' 104403  000000' 004022'         MSGN$,BEGIN,SOFT        ;ASCII MESSAGE CALL WITH COMMON HEADER
002206' 012767  000001  175672          MOV     #1,ERRTYP       ; DATA ERROR
                                        ;*****************************************************************
                                        ;
002214' 104406  000000' 003616'         SOFER$,BEGIN,TABLE      ;*****************************************
                                        ;
002222' 000205                  4$:     RTS     R5              ; RETURN, ERRORS
002224' 004567  176762          5$:     JSR     R5,CLEAR        ; CLEAR OUT ERRORS
002230' 005725                          TST     (R5)+           ; SKIP RETRY
002232' 000205                          RTS     R5              ; RETURN OK
                                ;       ----------------------------
```



```
                                ;
                                ;
002234' 016700  175546          SETUP:  MOV     ADDR,R0         ; GET DEVICE ADDRESS
002240' 010067  001352                  MOV     R0,RKDS         ; GENERATE CONTPOLLER REGS. ADDRESSES
002244' 005720                          TST     (R0)+
002246' 010067  001346                  MOV     R0,RKER
002252' 005720                          TST     (R0)+
002254' 010067  001342                  MOV     R0,RKCS
002260' 005720                          TST     (R0)+
002262' 010067  001336                  MOV     R0,RKWC
002266' 005720                          TST     (R0)+
002270' 010067  001332                  MOV     R0,RKBA
002274' 005720                          TST     (R0)+
002276' 010067  001326                  MOV     R0,RKDA
002302' 005720                          TST     (R0)+
002304' 010067  001322                  MOV     R0,RKMR
002310' 005720                          TST     (R0)+
002312' 010067  001316                  MOV     R0,RKDB
                                ;
002316' 016700  175466                  MOV     VECTOR,R0       ; GET THE VECTOR ADDRESS
002322' 012720  000506'                 MOV     #STRT,(R0)+     ; SET POINTER JUST IN CASE
002326' 116710  175460                  MOVB    BR1,(R0)        ; SET PRIORITY
                                ;
002332' 000207                  2$:     RTS     PC              ; RETURN
                                ;       ----------------------------
                                ;
                                ;
002334' 005001                  CONVRT: CLR     R1              ; ZERO REG. 1
002336' 105067  001532                  CLRB    SIDE            ; ZERO THE SIDE INDICATOR
002342' 012703  177764                  MOV     #-12.,R3        ; LOAD REG. 3
002346' 012704  000013                  MOV     #11.,R4         ; LOAD REG. 4
002352' 022700  004537                  CMP     #2399.,R0       ; IS BLOCK ON SIDE 0 ?
002356' 002005                          BGE     1$              ; YES, CONTINUE
002360' 152767  000020  001506          BISB    #BIT4,SIDE      ; NO, FLIP TO SIDE 1
002366' 062700  173240                  ADD     #-2400.,R0      ; NORMALIZE BLOCK # FOR SIDE 1
002372' 020400                  1$:     CMP     R4,R0           ; FIND THE RIGHT CYLINDER ?
002374' 002003                          BGE     2$              ; YES, CONTINUE
002376' 060300                          ADD     R3,R0           ; NO, SUBTRACT 12 SECTORS (1 CYLINDER)
002400' 005201                          INC     R1              ; KEEP TRACK OF CYLINDER ADDRESS
002402' 000773                          BR      1$              ; GO TRY AGAIN
002404' 010067  000154          2$:     MOV     R0,DSKADR       ; LOAD THE SECTOR ADDRESS
002410' 156767  001160  000146          BISB    SIDE,DSKADR     ; LOAD THE SIDE ADDRESS
002416' 012702  000005                  MOV     #5,R2           ; SET UP FOR SHIFT
002422' 006301                  3$:     ASL     R1              ; LINE UP CYL. ADR. WITH DSKADR
002424' 005302                          DEC     R2              ; DONE ?
002426' 003375                          BGT     3$              ; NO, GO SHIFT AGAIN
002430' 050167  000130                  BIS     R1,DSKADR       ; YES, LOAD THE CYLINDER ADDRESS
002434' 000207                          RTS     PC              ; RETURN
                                ;       ----------------------------
```

```
002436' 012777  000001  001156  REZET:  MOV     #1,@RKCS                ; EXECUTE CONTROLLER RESET
002444' 004767  000030                  JSR     PC,WAIT1                ; GO WAIT FOR CONTROLLER READY
002450' 004767  177200                  JSR     PC,NOTRDY               ; MAKE SURE ALL CHOSEN DRIVES ARE READY
002454' 004767  177102          1$:     JSR     PC,DRVADR               ; GET A DRIVE ADDRESS
002460' 132767  000010  001407          BITB    #BIT3,FLAG              ; ALL DRIVES DONE ?
002466' 001003                          BNE     2$                      ; YES, RETURN
002470' 004567  176516                  JSR     R5,CLEAR                ; ISSUE DRIVE RESET AND CONTROLLER CLEAR
002474' 000767                          BR      1$                      ; KEEP GOING
002476' 000207                  2$:     RTS     PC                      ; RETURN
                                ;       ------------------------

002500' 012767  077777  001106  WAIT1:  MOV     #77777,CLK              ; SET THE TIMER
002506' 105777  001110          1$:     TSTB    @RKCS                   ; CONTROLLER READY ?
002512' 100417                          BMI     2$                      ; YES, CONTINUE
002514' 104407  000000'                 BREAK$,BEGIN                    ;TEMPORARY RETURN TO MONITOR....
002520' 104407  000000'                 BREAK$,BEGIN                    ;THEN CONTINUE AT NEXT INSTRUCTION.
002524' 005367  001064                  DEC     CLK                     ; WAIT SOME MORE ?
002530' 001366                          BNE     1$                      ; YES
002532' 012767  000003  175346          MOV     #3,ERRTYP               ; CONTROLLER NOT READY
                                        ;*****************************************************************
002540' 104405  000000' 003616'         HRDER$,BEGIN,TABLE              ; CONTROLLER NOT READY
                                        ;*****************************************************************
002546' 000167  176176                  JMP     FINI                    ; GO DROP THE MODULE
002552' 000207                  2$:     RTS     PC                      ; READY, RETURN
                                ;       ------------------------

002554' 000000                  DLTCNT: 0
002556' 000000                  CNT:    0
002560' 000000                  FUNC:   0
002562' 000000                  XMEM:   0
002564' 000000                  DSKADR: 0
002566' 000000                  DVICE:  0
002570' 000000                  DRIVE:  0
002572' 000000                  DRYVE:  0
002574' 000000                  DRVSFT: 0
002576' 000000                  BLK1:   0
002600' 000000                  BLK2:   0
002602' 000000                  BSIZ:   0
002604' 000000                  TBUF:   0
002606' 000000                  WCNT1:  0
002610' 000000                  WCNT2:  0
002612' 000400                  BUFLEN: 256.
002614' 000400                  BUFIN:  .BLKW   256.
003614' 000000                  CLK:    0
003616'                         TABLE:
003616' 000000                  RKDS:   0
003620' 000000                  RKER:   0
003622' 000000                  RKCS:   0
003624' 000000                  RKWC:   0
003626' 000000                  RKBA:   0
003630' 000000                  RKDA:   0
003632' 000000                  RKMR:   0
003634' 000000                  RKDB:   0
003636' 177777                          177777
```



```
003640' 020040  044040  051101  MES1:   .ASCIZ  '   HARD ERROR'
003646' 020104  051105  047522
003654' 000122
003656' 020040  051440  043117  MES2:   .ASCIZ  '   SOFT ERROR'
003664' 020124  051105  047522
003672' 000122
003674' 020040  051104  053111  MES4:   .ASCIZ  '  DRIVE  '
003702' 020105  000040
003706' 020040  051104  050117  MES5:   .ASCIZ  '  DROPPED%'
003714' 042520  022504     000
003721'    040  042522  051124  MES6:   .ASCIZ  ' RETRY EXCEEDED%'
003726' 020131  054105  042503
003734' 042105  042105  000045
003742' 053440  044522  042524  MES7:   .ASCIZ  ' WRITE'
003750'    000
003751'    040  051127  052111  MES8:   .ASCIZ  ' WRITE-CHECK'
003756' 026505  044103  041505
003764' 000113
003766' 051040  040505  000104  MES9:   .ASCIZ  ' READ'
003774' 040504  040524  046040  MES10:  .ASCIZ  'DATA LATE ERROR%'
004002' 052101  020105  051105
004010' 047522  022522     000
        004016'                         .EVEN
004016' 003640'                 HARD:   MES1
004020' 177777                          177777
004022' 003656'                 SOFT:   MES2
004024' 177777                          177777
004026' 003742'                 EXCED1: MES7
004030' 003721'                         MES6
004032' 177777                          177777
004034' 003751'                 EXCED2: MES8
004036' 003721'                         MES6
004040' 177777                          177777
004042' 003766'                 EXCED3: MES9
004044' 003721'                         MES6
004046' 177777                          177777
004050' 003674'                 DRP:    MES4
004052' 004071'                         NUMB
004054' 003706'                         MES5
004056' 177777                          177777
004060' 003774'                 DLTERR: MES10
004062' 177777                          177777
004064' 000005                  ADR1:   .BLKB   5
004071'    000                  NUMB:   .BYTE   0
004072' 000000                          .WORD   0
004074'    000                  SIDE:   .BYTE   0
004075'    000                  FLAG:   .BYTE   0
                                        .EVEN
004076'    000                  TRY1:   .BYTE   0
004077'    000                  TRY2:   .BYTE   0
004100'    000                  TRY3:   .BYTE   0
        004102'                         .EVEN

        000001                          .END
```

```
ACSR      000102R      193#    489*
ADDR      000006R      159#    546
ADDR22=   001000       222#
ADR1      004064R      403     690#
ASB       000106R      197#    481*
ASTAT     000104R      195#
AWAS      000110R      198#    483*
BEGIN     000000R      156#    249     261     266     283     304     310     315     323     329     335     340     379
                       383     403     498     499     505     524     526     529     534     537     608     609     614
BIT0    = 000001       222#    337
BIT1    = 000002       222#    234
BIT10   = 002000       222#
BIT11   = 004000       222#    426
BIT12   = 010000       222#
BIT13   = 020000       222#    450
BIT14   = 040000       222#    509     515
BIT15   = 100000       222#
BIT2    = 000004       222#    295     297     444     522
BIT3    = 000010       222#    275     451     458     470     597
BIT4    = 000020       222#    577
BIT5    = 000040       222#    280
BIT6    = 000100       222#    285     366     473     500
BIT7    = 000200       222#
BIT8    = 000400       222#
BIT9    = 001000       222#    519
BLK1      002576R      231*    269     298*    410*    411     413*    414     419     629#
BLK2      002600R      414*    630#
BLOCK     001404R      265     410#
BREAK$=   104407       222#    498     499     608     609
BR1       000012R      161#    565
BR2       000013R      162#
BSIZ      002602R      425*    428*    430     631#
BTOD$   = 104421       222#
BUFIN     002614R      204     636#
BUFLEN    002612R      635#
CDATA$=   104412       222#    304
CDERCT    000144R      212#
CDWDCT    000146R      213#
CLEAR     001212R      363#    531     540     599
CLK       003614R      496*    502*    605*    610*    637#
CNT       002556R      223*    257     308*    621#
CONFIG    000056R      181#
CONVRT    002334R      270     571#
CSRA      000100R      191#    488*
DATCK$=   104411       222#
DATER$=   104404       222#
DLTCNT    002554R      228*    521*    620#
DLTERR    004060R      524     688#
DRIVE     002570R      230*    454*    461*    468*    626#
DROP      001346R      248     282     339     393#    504
DRP       004050R      249     283     340     505     684#
DRVADR    001562R      272     449#    455     469     596
DRVSFT    002574R      233*    278     365     450*    460*    467*    472     628#
DRYVE     002572R      232*    247*    251*    394     403     449*    452     459*    466*    627#
DSKADR    002564R      277*    278*    279     376     584*    585*    590*    624#
DVICE     002566R      229*    230     245     254     273     399*    461     468     625#
```



```
DVID1     000014R      163#    229
ENDIT$=   104413       222#    310
END$    = 104410       222#    315
ERRORS    002052R      386     508#
ERRTYP    000106R      196#    527*    535*    612*
ERSUB1    001764R      488#    509
ERSUB2    001740R      481#
EXCED1    004026R      323     675#
EXCED2    004034R      329     678#
EXCED3    004042R      335     681#
EXIT$   = 104400       222#    379
FINI      000750R      255     274     314#    616
FLAG      004075R      275     295     297*    444*    451*    458*    470     597     694#
FREE      000150R      214#
FUNC      002560R      347*    352*    357*    377*    378     622#
GETPA$=   104415       222#    261
GO        000656R      293#    322
GOA       000712R      296     300#    328
GOB       000720R      302#    334
GOGO      001272R      351     356     361     375#
GWBUF$=   104414       222#    266
HARD      004016R      526     671#
HRDCNT    000044R      176#
HRDER$=   104405       222#    529     614
HRDPAS    000050R      178#
ICONT     000036R      173#
ICOUNT    000040R      174#
IDNUM     000122R      203#
IMODX.=   000000       215#    267
INIT      000030R      170#
INTR      000120R      202#    226*
MAP22$=   104416       222#
MES1      003640R      648#    671
MES10     003774R      667#    688
MES2      003656R      651#    673
MES4      003674R      654#    684
MES5      003706R      656#    686
MES6      003721R      658#    676     679     682
MES7      003742R      661#    675
MES8      003751R      663#    678
MES9      003766R      666#    681
MODNAM    000000R      157#
MODSP     000252R      171     220#
MSGN$   = 104403       222#    249     283     323     329     335     340     505     524     526     534
MSGS$   = 104402       222#
MSG$    = 104401       222#
NEXT      000540R      272#    284     312     341
NEXTA     001046R      324     330     337#
NOTRDY    001654R      287     466#    595
NTRUPT    001326R      375     381#
NULL    = 000000       222#
NUMB      004071R      685     691#
OPEN    = 000000       158     164     165     166     167     184     185     186     187     188     189     190     191
                       193     195     197     198     200     201     202     205     206     208     209     211     212
                       213     214     222#

OTOA$   = 104420       222#    403
```

```
PASCNT  000034R          172#
PASS    000742R          309#
PIRQS = 000004           222#    383
POPSP   005726           222#
POPSP2= 022626           222#
PRTV  = 000000           222#
PRTV0 = 000000           162     222#
PRTV1 = 000040           222#
PRTV2 = 000100           222#
PRTV3 = 000140           222#
PRTV4 = 000200           222#
PRTV5 = 002240           161     222#
PRTV6 = 000300           222#
PRTV7 = 000340           222#
PS    = 177776           222#
PSW   = 177776           222#
PUSH  = 005746           222#
PUSH2 = 024646           222#
RANDS = 104417           227#
RANNUM  000054R          180#
RBUFEA  000130R          206#    360
RBUFPA  000126R          205#    304     359
RBUFSZ  000132R          207#    262
RBUFVA  000124R          204#    261
READ    001160R          302     357#
RESTRT  000456R          199     257#
RES1    000056R          182#
RES2    000060R          183#
RETRY1  000754R          294     320#
RETRY2  001000R          301     326#
RETRY3  001024R          303     332#
REZET   002436R          253     503#
RKBA    003626R          349*    354*    359*    555*    643#
RKCS    003672R          363*    369*    371*    378*    488     489     509     551*    593*    606     641#
RKDA    003630R          279*    365*    376*    472*    557*    644#
RKDB    003634R          561*    646#
RKDS    003616R          280     285     366     426     473     500     547*    639#
RKER    003620R          511     515     519     549*    640#
RKWR    003632R          559*    645#
RKWC    003624R          348*    353*    358*    553*    642#
ROOM    001436R          419#    517
RSTRT   000112R          199#
RSTRT1  000466R          256     258     260#
SBADR   000102R          192#    482*
SETUP   002234R          252     546#
SIDE    004074R          227*    572*    577*    585     693#
SOFCNT  000042R          175#
SOFERS= 004406           222#    537
SOFPAS  000046R          177#
SOFT    004022R          534     673#
SPDINT  000032R          171#
SPSIZ = 000040             1#    215
SR1     000016R          164#    337     522
SR2     000020R          165#
SR3     000022R          166#
SR4     000024R          167#
```



```
START   000252R          170     223#    259
STAT    000026R          169#
STRT    000506R          265#    276     288     299     564
SVR0    000062R          184#
SVR1    000064R          185#
SVR2    000066R          186#
SVR3    000070R          187#
SVR4    000072R          188#
SVR5    000074R          189#
SVR6    000076R          190#
SYSCNT  000052R          179#
TABLE   003616R          529     537     614     638#
TBUF    002604R          632#
TRPDFD= 000022           222#
TRY1    004076R          289*    320*    321     696#
TRY2    004077R          326*    327     697#
TRY3    004100R          290*    332*    333     698#
VECTOR  000010R          160#    375*    563
WAIT    002002R          370     475     496#
WAIT1   002500R          364     372     594     605#
WASADR  000104R          194#    484*
WBUFEA  000136R          209#    350     355
WBUFPA  000134R          208#    349     354
WBUFRQ  000140R          210#
WBUFSZ  000142R          211#    267     435     438     440
WCNT1   002606R          267*    268*    348     353     633#
WCNT2   002610R          262*    263*    358     634#
WDFB    000116R          201#    225*
WDTO    000114R          200#    224*
WRITCK  001126R          300     352#
WRITE   001074R          293     347#
XFLAG   000003R          158#
XMEM    002562R          350*    355*    360*    377     623#
.     = 004102R          305     636#    670#    690#    699#

. ABS.  000000     000
        004102     001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XRKAG0,XRKAG0/SQL/CRF:SYM=DDXCOM,XRKAG0
 RUN-TIME:  1  2  .3 SECONDS
 RUN-TIME RATIO: 16/4=3.8
 CORE USED:  7K  (13 PAGES)
```