```
   .INSERT,11                    V$SGEN1   V75        02/01/75  MCMURRAY  V75
*A*V75     SET      1                                                     V75
   .INSERT,274                   V$SGEN1   V75        02/01/75  MCMURRAY  V75
*A*        TFT      V75                                                   V75
*A*SGMRY5  DATA     0            V75 FLAG                                 V75
   .REPLACE,932                  V$SGEN1   V75        02/01/75  MCMURRAY  V75
*D*        TFF      VORTEX-1                                             V2
*A*        TFT      VORTEX-1                                              V75
*A*        GOTO     1                                                     V75
*A*        TFT      V75                                                   V75
*A*        TZA                                                            V75
*A*        TFF      V75                                                   V75
   .INSERT,933                   V$SGEN1   V75        02/01/75  MCMURRAY  V75
*A*1       CONT                                                           V75
   .INSERT,941                   V$SGEN1   V75        02/01/75  MCMURRAY  V75
*A*        TFT      V75                                                   V75
*A*        TZA                                                            V75
*A*        TFF      V75                                                   V75
   .INSERT,948                   V$SGEN1   V75        02/01/75  MCMURRAY  V75
*A*        TFF      V75                                                   V75
*A*        GOTO     1                                                     V75
*A*        LDAE     SG$VCH       GET TERMINATOR                           V75
*A*        SUB      =0215                                                 V75
*A*        JAZ      SG3DTR       EXIT ON END OF LINE                      V75
*A*        SUB      =037                                                  V75
*A*        JANZ     SG3MRE       ERROR IF TERMINATOR NOT COMMA            V75
*A*        JSR      SGGNT,Y      YES, GET NEXT PARAMETER                  V75
*A*        DATA     TBUF2                                                 V75
*A*        DATA     3                                                     V75
*A*        SUB      THREE                                                 V75
*A*        JANZ     SG3MRE       ERROR IF NOT 3 CHARS                     V75
*A*        LDA      TBUF2                                                 V75
*A*        FRAI     'V7'                                                  V75
*A*        JANZ     SG3MRE       ERROR IF NOT 'V7'                        V75
*A*        LDA      TBUF2+1                                               V75
*A*        FRAI     '5 '                                                  V75
*A*        ANA      LHW                                                   V75
*A*        JANZ     SG3MRE       ERROR IF NOT '5 '                        V75
*A*        INR      SGMRY5       SET V75 FLAG                             V75
*A*1       CONT                                                           V75
   .REPLACE,1030,1062            V$SGEN1   SMR-767   12/13/74 NEWDORF    F
*D*        LDA      TEN                                                   V2
*D*        STA      EQPT+1       INITIALIZE DIVISOR                       V2
```

```
*D*          LDA     FOUR                                                V2
*D*          STA     FOPT+2          INITIALIZE LOOP COUNT               V2
*D*EOPDL     CALL    SGGNC           GET NEXT CHAR                       V2
*D*          LDB     SGSVCH                                              V2
*D*          TBA                                                         V2
*D*          SUB     =0215                                               V2
*D*          JAZ     FOP9            EXIT AT END OF STATEMENT            V2
*D*          SUB     FOPCM                                               V2
*D*          JAZ     FOP8            EXIT ON ','                         V2
*D*          LDA     FOPT+2                                              V2
*D*          JAZ     FOP2            ERROR IF MORE THAN 4 DIGITS         V2
*D*          TBA                                                         V2
*D*          SUB     =0260                                               V2
*D*          JAN     FOP2            ERROR IF NOT DIGIT                  V2
*D*          TAB                                                         V2
*D*          SUB     TEN                                                 V2
*D*          JAP     FOP2            ERROR IF NOT DIGIT                  V2
*D*          TBA                                                         V2
*D*          TZB                                                         V2
*D*          DIV     FOPT+1          CONVERT TO BINARY FRACTION          V2
*D*          TBA                                                         V2
*D*          ADD     TTFM+5          UPDATE I/O VALUE                    V2
*D*          STA     TTFM+5                                              V2
*D*          LDA     FOPT+2                                              V2
*D*          DAB                     DROP LOOP COUNT                     V2
*D*          STA     FOPT+2                                              V2
*D*          LDB     FOPT+1                                              V2
*D*          TZA                                                         V2
*D*          MUL     TEN                                                 V2
*D*          STB     FOPT+1          UPDATE DIVISOR                      V2
*D*          JMP     FOPDL                                               V2
*A*          LDAI    0130260         TWO ASCII ZEROS.                    F
*A*          STA     FOPT+1          ZERO OUT BUFFER.                    F
*A*          ISR     SGGNT,X         INPUT TWO ASCII DIGITS.             F
*A*          DATA    FOPT+1                                              F
*A*          DATA    2                                                   F
*A*          STB     FOPT            STORE TERMINATOR.                   F
*A*          TBA                     LOAD A WITH TERMINATOR.             F
*A*          SUB     =0215           IS IT EOL?                          F
*A*          JAZ     FOP10A          JUMP IF YES.                        F
*A*          SUB     FOPCM           IS IT COMMA?                        F
*A*          JANZ    FOP2            IF NO, ERROR.                       F
*A*EOP10A    LDA     FOPT+1          IS THE BUFFER                       F
```

```
*A*        SUBI    0130260      LESS THAN TWO ZEROS?                    F
*A*        JAN     EQP2         YES -- PARAMETER ERROR.                 F
*A*        SUBI    8            NO -- CONVERT .08 AND .09               F
*A*        JAN     EQP10B       TO .10 (BECAUSE LEADING ZEROS           F
*A*        SUBI    2
*A*        JAP     EQP10B
*A*        LDAI    0130660      MAKES THEM LOOK LIKE OCTAL)             F
*A*        STA     EQPT+1                                               F
*A*EQP10B  LDXI    EQPT+1       BUFFER ADDRESS                          F
*A*        LDAI    2            NUMBER OF DIGITS.                       F
*A*        CALL    SGCNV        CONVERT ASCII TO BINARY.                F
*A*        JOF     EQP2         PARAMETER ERROR.                        F
*A*        TZA                  LOAD MS IN A. LS IS IN B.               F
*A*        LRLB    6            SHIFT LEFT 2 OCTAL PLACES               F
*A*        DIVI    100          CONVERT TO BINARY FRACTION.             F
*A*        LRLB    9            LEFT JUSTIFIED.                         F
*A*        STB     ITEM+5       STORE I/O FRACTION IN STACK.            F
*A*        LDA     EQPT         TERMINATOR.                             F
*A*        SUB     =0215        IS IT EOL?                              F
*A*        JAZ     EQP9         YES--THEN EXIT.                         F
   .INSERT,1063                 VSSGEN1   SMR-767   12/13/74 NEWDORF    F
*A*        TZA                  DISABLE TERMINATOR TEST                 F
```

```
   .INSERT,11                    V$SGEN2   V75        02/01/75  MCMURRAY  V75
*A*V75     SET     1                                                      V75
   .INSERT,282                   V$SGEN2   V75        02/01/75  MCMURRAY  V75
*A*        TFT     V75                                                    V75
*A*SGMRY5  BSS     1             V75 FLAG                                 V75
   .REPLACE,748                  V$SGEN2   ALT SRCH   04/09/75  KERNS     F.2
*D*        DATA    2,SGTBUF
*A*        DATA    10,SGTBUF                                              F.2
   .REPLACE,764                  V$SGEN2   ALT SRCH   04/03/75  KERNS     F.2
*D*ICR15   CALL    SGCLR         CLEAR BUFFER                             C
*A*ICR15   LDA     SGTBUF+1                                               F.2
*A*        SUB     ='T,'                                                  F.2
*A*        TANZ    TCR19         IF NO NAME SPECIFIED                     F.2
*A*        CALL    SGMOV         MOVE TARGET NAME TO STORAGE              F.2
*A*        DATA    4,SGTBUF+2,ICR20                                       F.2
*A*        TS1N    TCR19         IF SSW1 NOT SET DO NOT REWIND ALT        F.2
*A*        CALL    TOCS          REWIND ALT UNIT                          F.2
*A*        DATA    REW+ALT                                                F.2
*A*        CALL    TOCS                                                   F.2
*A*        DATA    ALT                                                    F.2
*A*        DATA    TCR4,ICR4,TCR19,*-6                                    F.2
*A*ICR19   CALL    SGCLR         CLEAR BUFFER                             F.2
   .INSERT,772                   V$SGEN2   ALT SRCH   04/03/75  KERNS     F.2
*A*        STX     TCR31                                                  F.2
*A*        LDA     TCR20                                                  F.2
*A*        TAZ     TCR30         IF NAME SEARCH NOT SPECIFIED             F.2
*A*ICR19A  LDA     SGTBUF        FETCH FIRST WORD                         F.2
*A*        LSRA    13                                                     F.2
*A*        ANA     THREE                                                  F.2
*A*        SUB     THREE                                                  F.2
*A*        TAZ     TCR25         IF OBJECT MODULE                         F.2
*A*        LDA     SGTBUF                                                 F.2
*A*        SUBI    'TD'                                                   F.2
*A*        TAZ     TCR21         IF FOUND                                 F.2
*A*        JMP     TCR6          REPORT CONTROL REC ERROR                 F.2
*A*ICR21   EQU     *                                                      F.2
*A*        LDXI    SGTBUF+2                                               F.2
*A*        LDBI    TCR20                                                  F.2
*A*        CALL    SGCPR         COMPARE TO TARGET NAME                   F.2
*A*        TAZ     TCR30         IF MATCH, CONTINUE ALT PROCESSING        F.2
*A*ICR22   CALL    SGCLR         CLEAR BUFFER                             F.2
*A*        CALL    TOCS          GET NEXT RECORD FROM ALT                 F.2
*A*        DATA    RALF+ALT                                               F.2
```

```
*A*        DATA    60,SGIBUF                                          F.2
*A*        CALL    TOCS          IF EOF, POST MODULE  NOT FOUND       F.2
*A*        DATA    ALT                                                F.2
*A*        DATA    TCR4,ICR35,ICR4,*-6                                F.2
*A*        JMP     TCR19A        CHECK THIS RECORD                    F.2
*A*ICR25   EQU     *                                                  F.2
*A*        LDA     SGIBUF                                             F.2
*A*        ANA     =0173400                                           F.2
*A*        ERA     =060400                                            F.2
*A*        JANZ    TCR22         IGNORE IF NOT FIRST RECORD           F.2
*A*        LDXI    SGIBUF+3      MODULE NAME WORDS                    F.2
*A*        LDBI    TCR20                                              F.2
*A*        CALL    SGCPR         COMPARE TO TARGET NAME               F.2
*A*        JAZ     TCR30         IF MATCH, CONTINUE ALT PROCESSING    F.2
*A*        JMP     TCR22         CONTINUE SEARCH                      F.2
*A*ICR30   EQU     *                                                  F.2
*A*        TZA                                                        F.2
*A*        STA     TCR20                                              F.2
*A*        LDXI    0             RESTORE Y                            F.2
*A*ICR31   BES     0                                                  F.2
   .INSERT.Z91                   VSSGEN2   ALT SRCH  04/03/75  KERNS  F.2
*A*ICR35   LDA     FROB                                               F.2
*A*        JMP     TCR4+1        RECOVER AND INPUT RECORD             F.2
*A*ICR20   DATA    0,0,0,0       TARGET NAME BUFFER                   F.2
*A*        EJEC                                                       F.2
*A**                                                                  F.2
*A**       SUBROUTINE-SGCPR   (COMPARE TO BYTE ARRAYS FOR MATCH)      F.2
*A**                          (TERMINATES ON MISMATCH OF OCCURENCE    F.2
*A**                           OF BLANK IN TARGET ARRAY)              F.2
*A**                                                                  F.2
*A**       CALLING SEQUENCE                                           F.2
*A**       LDX     INPUT NAME ADDR                                    F.2
*A**       LDB     TARGET NAME ADDR                                   F.2
*A**               (ADDR IS COMPLEMENTED FOR RIGHT BYTE)              F.2
*A**       CALL    SGCPR                                              F.2
*A**                                                                  F.2
*A**       RETURNS                                                    F.2
*A**       (A)=0 FOR GOOD COMPARE                                     F.2
*A**       (B) AND (X) POINTING TO NEXT BYTES AFTER COMPARE STOP      F.2
*A**                                                                  F.2
*A*SGCPR   ENTR                                                       F.2
*A*        STX     CPRX                                               F.2
*A*        STB     CPRB                                               F.2
```

```
*A*CPR10  LDB   CPRB                                                F.2
*A*       CALL  CPFTCH    FETCH TARGET BYTE                         F.2
*A*       STB   CPRB                                                F.2
*A*       STA   CPRT      SAVE BYTE                                 F.2
*A*       SUBI  0240                                                F.2
*A*       IAZ   CPRTN     END OF COMPARE IF BLANK FOUND             F.2
*A*       LDB   CPRX                                                F.2
*A*       CALL  CPFTCH    FETCH INPUT BYTE                          F.2
*A*       STB   CPRX                                                F.2
*A*       ERA   CPRT      TARGET BYTE                               F.2
*A*       IAZ   CPR10     IF MATCH FOUND                            F.2
*A*CPRTN  EQU   *                                                   F.2
*A*       LDB   CPRB                                                F.2
*A*       LDY   CPRX                                                F.2
*A*       RETU* SGCPR                                               F.2
*A*CPFTCH ENTR                                                      F.2
*A*       TBA                                                       F.2
*A*       IAN   CPF10     IF RIGHT BYTE FETCH                       F.2
*A*       LDA   0,B                                                 F.2
*A*       LSRA  4                                                   F.2
*A*       CPB             SET FOR RIGHT BYTE                        F.2
*A*       RETU* CPFTCH                                              F.2
*A*CPF10  CPB                                                       F.2
*A*       LDA   0,B                                                 F.2
*A*       ANA   SHW                                                 F.2
*A*       TBR             SET FOR NEXT LEFT BYTE                    F.2
*A*       RETU* CPFTCH                                              F.2
*A**                                                                F.2
*A*CPRB   DATA  0         TARGET BYTE POINTER                       F.2
*A*CPRX   DATA  0         INPUT BYTE POINTER                        F.2
*A*CPRT   DATA  0         TARGET BYTE                               F.2
   .REPLACE,1100,1103     VSSGEN2  V75      02/01/75  MCMURRAY    V75
*D*       TFF   VORTEX-2                                            V2
*D*       LDX   #20                                                 V2
*D*       TFF   VORTEX-1                                            V2
*D*       LDX   #28                                                 V2
*A*       TFT   VORTEX-1                                            V75
*A*       GOTO  1                                                   V75
*A*       TFT   V75                                                 V75
*A*       LDX   #38                                                 V75
*A*       TFF   V75                                                 V75
*A*       LDX   #28                                                 V75
*A*1      CONT                                                      V75
```

```
*A*        TFT     VORTEX-2                                                  V75
*A*        GOTO    1                                                         V75
*A*        TFT     V75                                                       V75
*A*        LDY     =39                                                       V75
*A*        TFF     V75                                                       V75
*A*        LDY     =29                                                       V75
*A*1       CONT                                                              V75
   .REPLACE,1124,1129          V$SGEN2   V75        02/01/75  MCMURRAY       V75
*D*        LDA     MA
*D*        TFF     VORTEX-2                                                  V2
*D*        SUB     =28                                                       V2
*D*        TFF     VORTEX-1                                                  V2
*D*        SUB     =25                                                       V2
*D*        STA     TBUF1+3     SAVE TIDB ENTRY LOC. FOR CHAINING
   .REPLACE,1168               V$SGEN2   V75        02/01/75  MCMURRAY       V75
*D*        LDA     =0215
*A*        TZA                                                               V75
   .INSERT,1179                V$SGEN2   V75        02/01/75  MCMURRAY       V75
*A*        TFF     V75                                                       V75
*A*        GOTO    1                                                         V75
*A*        LDA     SGMRY5                                                    V75
*A*        JAZ     TDF8C       V75 SYSTEM ?                                  V75
*A*        LDAE    SGSVCH      YES                                           V75
*A*        SUB     =0254       GET TERMINATOR                                V75
*A*        TFF     VORTEX-2                                                  V75
*A*        JANZ    TDF8C       TEST FOR COMMA                                V75
*A*        TFT     VORTEX-1                                                  V75
*A*        GOTO    2                                                         V75
*A*        JAZ     TDF8A       TEST FOR COMMA                                V75
*A*        LDA     SGWBUF+1                                                  V75
*A*        BT      RAO+11,TDF8B   RESIDENT TIDB ?                            V75
*A*        JMP     TDF8C       YES                                           V75
*A*2       CONT                                                              V75
*A*TDF8A   JSR     SGGNT,Y     YES. GET NEXT 3 CHARS                         V75
*A*        DATA    TBUF1+5                                                   V75
*A*        DATA    3                                                         V75
*A*        SUB     THREE                                                     V75
*A*        JANZ    TDF6                                                      V75
*A*        LDA     TBUF1+5                                                   V75
*A*        FRAI    !V7!                                                      V75
*A*        JANZ    TDF6        ERROR IF NOT !V75!                            V75
*A*        LDA     TBUF1+6                                                   V75
*A*        FRAI    !5 !                                                      V75
```

```
*A*        ANA     LHW                                                    V75
*A*        JANZ    TDF6          LONG TIDB ?                              V75
*A*        LDA     BS14          YES                                      V75
*A*        ORA     SGWBUF+2      SET V75 FLAG                             V75
*A*        STA     SGWBUF+2                                               V75
*A*TDFBB   LDAI    -10                                                    V75
*A*        DATA    01006                                                  V75
*A*1       CONT                                                           V75
*A*TDFBC   TZA                                                            V75
*A*        TFT     V75                                                    V75
*A*        STA     TBUF1+5       SAVE V75 FLAG                            V75
*A*        ADD     MA                                                     V75
*A*        TFF     VORTEX-1                                               V75
*A*        SUB     =25                                                    V75
*A*        TFF     VORTEX-2                                               V75
*A*        SUB     =28                                                    V75
*A*        STA     TBUF1+3       STORE TIDB BASE ADDRESS                  V75
   .REPLACE,1212,1215           VSSGEN2   V75        02/01/75  MCMURRAY  V75
*D*        TFF     VORTEX-2                                               V2
*D*        LDB     =29                                                    V2
*D*        TFF     VORTEX-1                                               V2
*D*        LDB     =26           STORE TIDB IN VIRTUAL MEMORY             V2
*A*        TFF     VORTEX-2                                               V75
*A*        LDA     =29                                                    V75
*A*        TFF     VORTEX-1                                               V75
*A*        LDA     =26                                                    V75
*A*        TFT     V75                                                    V75
*A*        SUB     TBUF1+5                                                V75
*A*        TAB                                                            V75
   .INSERT,1599                 VSSGEN2   BICTABLE  03/12/75  NEWDORF    F.2
*A*        STA     TBUF3                                                  F.2
   .INSERT,1615                 VSSGEN2   BICTABLE  03/12/75  NEWDORF    F.2
*A*        JMP     FTG3                                                   F.2
   .INSERT,1616                 VSSGEN2   BICTABLE  03/12/75  NEWDORF    F.2
*A*FTG9    LDA     TBUF3         CONVERT DEVICE ADDRESS                   F.2
*A*        ANAI    070           TO TWO SIX BIT                           F.2
*A*        ADDI    03200         LOADER CODE WORDS                        F.2
*A*        LRLA    03                                                     F.2
*A*        TAB                                                            F.2
*A*        LDA     TBUF3                                                  F.2
*A*        ANAI    07                                                     F.2
*A*        ADDI    020                                                    F.2
*A*        DATA    05031         INCLUSIVE OR B INTO A                    F.2
```

```
*A*          STA     FTG16          STORE LOADER WORD IN IBICXX BUF          F.2
*A*          LDAI    -1             SEARCH ONLY                              F.2
*A*          LDRI    FTG16-2        FOR IBICXX                               F.2
*A*          LDXI    $G$CB6         IN LOADER TABLE                          F.2
*A*          CALL    $G$AF                                                   F.2
*A*          DATA    0              SIZE OF KEY                              F.2
*A*          JAP     FTG14          JUMP IF ENTRY FOUND                      F.2
*A*          LDXI    15             LINEAR SEARCH THRU DEFAULT TABL          F.2
*A*FTG11     LDAE    FTGTBL,X                                                F.2
*A*          ANA     RHW            DEV ADDR IS IN RHW                       F.2
*A*          SUB     TBUF3          CURRENT BIC DA                           F.2
*A*          JAZ     FTG13          JUMP IF MATCH                            F.2
*A*          JAP     FTG15          ERROR - NOT IN DEFAULT TABLE             F.2
*A*          TXA                                                             F.2
*A*          DAR            DECREMENT INDEX                                  F.2
*A*          JAN     FTG15          ERROR--NOT IN DEFAULT TBL                F.2
*A*          TAX                                                             F.2
*A*          JMP     FTG11          LOOP                                     F.2
*A*FTG13     LDAE    FTGTBL,X       BIC DA FOUND IN DEFAULT TABLE            F.2
*A*          ISRA    8              TRAP LOC IN LHW                          F.2
*A*FTG14     STA     VDATA                                                   F.2
*A*          ISR     VMWCT,X        STORE TRAP LOC IN BIC FLAG TABL          F.2
*A*          LDA     MA                                                      F.2
*A*          DAR                                                             F.2
*A*          STA     MA                                                      F.2
    .REPLACE,1617                   V$$GEN2   BICTABLE  03/12/75  NEWDORF   F.2
*D*FTG3      LDA     0,X            MOVE CONTROLLER NAME
*A*FTG3      LDX     FTG7           PTR TO EQUIP STACK ENTRY                 F.2
*A*          LDA     0,X            MOVE CONTROLLER NAME                     F.2
    .REPLACE,1642,1643              V$$GEN2   BICTABLE  03/12/75  NEWDORF   F.2
*D*          LDX     FTG7           RESTORE (XR)
*D*          JAP     FTG3           IF ENTRY MADE
*A*          JAP     FTG9           IF ENTRY MADE                            F.2
    .INSERT,1648                    V$$GEN2   BICTABLE  03/12/75  NEWDORF   F.2
*A*FTG15     LDA     ER25           POST "INCOMPLETE DEFINITION"             F.2
*A*          DATA    01006                                                   F.2
    .INSERT,1653                    V$$GEN2   BICTABLE  03/12/75  NEWDORF   F.2
*A*FTG16     DATA    030000         LOADER CODE FOR                          F.2
*A*          DATA    061246         IBIC XX                                  F.2
*A*          DATA    0                                                       F.2
*A**                                                                         F.2
*A**                 DEFAULT TABLE                                           F.2
*A*FTGTBL    DATA    057076         TRAP 136  BIC 76                         F.2
```

```
*A*        DATA    046074          114      74                          F.2
*A*        DATA    045072          112      72                          F.2
*A*        DATA    044070          110      70                          F.2
*A*        DATA    056066          134      66                          F.2
*A*        DATA    055064          132      64                          F.2
*A*        DATA    054062          130      62                          F.2
*A*        DATA    053060          126      60                          F.2
*A*        DATA    052056          124      56                          F.2
*A*        DATA    051054          122      54                          F.2
*A*        DATA    050052          120      52                          F.2
*A*        DATA    047050          116      50                          F.2
*A*        DATA    043026          106      26                          F.2
*A*        DATA    042024          104      24                          F.2
*A*        DATA    041022          102      22                          F.2
*A*        DATA    040020          100      20                          F.2
   .REPLACE,2958                 VSSGEN2   V75       02/01/75  MCMURRAY  V75
*D*        LDA     =25                                                   V2
*A*        TEE     V75                                                   V75
*A*        GOTO    2                                                     V75
*A*        LDA     SGMRY5                                                V75
*A*        TAZ     USA4A         V75 SYSTEM ?                            V75
*A*        LDA     TEN           YES                                     V75
*A*USA4A   EQU     *                                                     V75
*A*2       CONT                                                          V75
*A*        ADD     =25                                                   V75
   .INSERT,3338                  VSSGEN2   PATCH     11/12/74  KERNS     E.1
*A*        LDY     =SGSCB6                                               E.1
*A*        LDB     LCF25                                                 E.1
*A*        DECR    1             SET FOR SEARCH ONLY                     E.1
*A*        CALL    SGSAF         SEARCH FOR VSPTS7                       E.1
*A*        DATA    0                                                     E.1
*A*        TAN     LCF21A        IF NOT FOUND                            E.1
*A*        STA     LCTMP         SAVE PATCH AREA SIZE                    E.1
*A*        LDA     LLC           CURRENT VSBVN                           E.1
*A*        STA     LCF26+3                                               E.1
*A*        LDY     =SGSCB6                                               E.1
*A*        CALL    SG4ENT        ADD ENTRY FOR VSPEND                    E.1
*A*        DATA    LCF26                                                 E.1
*A*        LDA     LLC           CURRENT VSBVN                           E.1
*A*        SUB     LCTMP         PATCH AREA SIZE                         E.1
*A*        STA     LLC           UPDATE VSBVN                            E.1
*A*LCF21A  EQU     *                                                     E.1
   .INSERT,3339                  VSSGEN2   PATCH     11/13/74  KERNS     E.1
```

```
*A*          ANAI    0177000     SET TO PAGE BOUNDARY                          E.1
*A*          TAR                 LEAVE ROOM FOR CORE RESIDENT DIRECTORY        F.2
*A*          STA     LCF21B+1                                                  F.2
*A*          SUB     LLC                                                       F.2
*A*          JAN     LCF21B      OK, ORIGINAL VSRVN ABOVE NEW VSRVN            F.2
*A*          ADD     LLC         RESTORE VSRVN, CANNOT SAVE ROOM               F.2
*A*          JMP     LCF21C                                                    F.2
*A*LCF21B    IDAI    0           NEW VSRVN VALUE                               F.2
*A*LCF21C    EQU     *                                                         F.2
*A*          STA     LLC                                                       E.1
    .INSERT,3340                 VSSGEN2   PATCH     11/14/74  KERNS          E.1
*A*          STA     LCF27+3                                                   E.1
*A*          IDX     =SGSCRB                                                   E.1
*A*          CALL    SGAENT      ADD ENTRY FOR VSPSTR                          E.1
*A*          DATA    LCF27                                                     E.1
*A*          LDA     LLC         CURRENT VSRVN                                 E.1
    .INSERT,3376                 VSSGEN2   V75       02/01/75  MCMURRAY       V75
*A*          TEZ     V75                                                       V75
*A*          CNTO    1                                                         V75
*A*          LDA     SGMRV5                                                    V75
*A*          IAZ     LCF2A       V75 SYSTEM ?                                  V75
*A*          LDA     LCF1B+3     YES                                           V75
*A*          ORA     BS1         SET V75 FLAG                                  V75
*A*          STA     LCF1B+3                                                   V75
*A*LCF2A     EQU     *                                                         V75
*A*1         CONT                                                              V75
    .INSERT,3457                 VSSGEN2   PATCH     11/12/74  KERNS          E.1
*A*LCF25     BZE     *-1                                                       E.1
*A*          DATA    048372      VSPTSZ                                        E.1
*A*          DATA    0102303     TN                                            E.1
*A*          DATA    015         LOADER CODE                                   E.1
*A*LCF26     DATA    !VSPEND!,0  ENTRY FOR VSPEND                              E.1
*A*LCF27     DATA    !VSPSTR!,0  ENTRY FOR VSPSTR                              E.1
*A*LCTMP     DATA    0           TEMP STORAGE                                  E.1
    .INSERT,3469                 VSSGEN2   NO SMR     11/20/74  KERNS          E.1
*A*          LDB     =SGSCRB                                                   E.1
*A*          LDA     0,B         CURRENT START OF STACK                        E.1
*A*          STA     SRSTR                                                     E.1
*A*          LDA     1,B                                                       E.1
*A*          STA     SREND       END OF STACK                                  E.1
*A*          LDA     2,B         ITEM SIZE                                     E.1
*A*          STA     SRINC       ITEM SIZE                                     E.1
*A*          CALL    SGSRT       SORT CI DIRECTORY                             E.1
```

```
    .INSERT,3710                    VSSGEN2   NO SMR    11/20/74  KERNS    E,1
*A*         EJEC                                                           E,1
*A**                                                                       E,1
*A**        SGSRT                                                          E,1
*A**                                                                       E,1
*A**                PURPOSE:  SORT CI STACK NUMERICALLY, HIGH TO LOW       E,1
*A**                                                                       E,1
*A**                CALLING SEQUENCE:  CALL SGSRT                          E,1
*A**                                   LOC SRSTR = START OF STACK          E,1
*A**                                   LOC SREND = END OF STACK            E,1
*A**                                   LOC SRINC = STACK ITEM SIZE         E,1
*A**                                                                       E,1
*A**                RETURN PARAMETERS:  NONE                               E,1
*A**                                                                       E,1
*A*SGSRT    ENTR                                                           E,1
*A*SR05     LDA     SRSTR           STACK START                            E,1
*A*         ADD     SRINC           ITEM SIZE                              E,1
*A*         STA     SRITM           SET NEXT ITEM                          E,1
*A*SR10     LDX     SRSTR                                                  E,1
*A*         LDB     SRITM           NEXT ITEM                              E,1
*A*         LDA     0,X                                                    E,1
*A*         SUB     0,B                                                    E,1
*A*         TAP     SR20            IF CURRENT ITEM LE TOP ITEM            E,1
*A*         LDA     SRINC                                                  E,1
*A*         STA     SRCNT           SET EXCHANGE COUNT                     E,1
*A*SR15     LDA     0,X                                                    E,1
*A*         STA     SRTMP           EXCHANGE                               E,1
*A*         LDA     0,B                                                    E,1
*A*         STA     0,X             ONE                                    E,1
*A*         LDA     SRTMP                                                  E,1
*A*         STA     0,B             WORD                                   E,1
*A*         LDA     SRCNT           LOOP COUNT                             E,1
*A*         DAR                                                            E,1
*A*         TAZ     SR20            IF DONE                                E,1
*A*         STA     SRCNT                                                  E,1
*A*         TXB                     UPDATE ITEM POINTERS                   E,1
*A*         TBR                                                            E,1
*A*         JMP     SR15            CONTINUE EXCHANGE                      E,1
*A*SR20     LDA     SRITM           CURRENT ITEM                           E,1
*A*         ADD     SRINC           ITEM SIZE                              E,1
*A*         STA     SRITM           SET FOR NEXT ITEM                      E,1
*A*         SUB     SREND           END OF STACK                           E,1
*A*         TAP     SR30            IF END OF STACK                        E,1
```

```
*A*        IMP     SR10        CONTINUE BUBBLE SORT                    E,1
*A*SR30    LDA     SRSTR       CURRENT TOP OF SORT                     E,1
*A*        ADD     SRINC       ITME SIZE                               E,1
*A*        STA     SRSTR       SET NEW TOP                             E,1
*A*        SUB     SREND       END OF STACK                            E,1
*A*        IAP*    SGSRT       RETURN IF SORT DONE                     E,1
*A*        IMP     SR05                                                E,1
*A**                                                                   E,1
*A*SRSTR   DATA    0           START OF STACK                          E,1
*A*SREND   DATA    0           END OF STACK                            E,1
*A*SRITM   DATA    0           CURRENT ITEM                            E,1
*A*SRCNT   DATA    0           EXCHANGE COUNT                          E,1
*A*SRINC   DATA    0           ITM+      SIZE                          E,1
*A*SRTMP   DATA    0           TEMP STORAGE                            E,1
   .INSERT.5220                V*SGEN2   V75       02/01/75  MCMURRAY  V75
*A*        LDA     GACD2                                               V75
*A*        IAZ     GACD1-1     CHECK IF TEST DISABLED                  V75
```

```
   .INSERT,405                    VSSGEN4   SMR-      04/29/75  KERNS     E.2
*A*        LDA     3,X            X=DST ADDR, WORD 3 = DEVICE NAME         E.2
*A*        ANA     RHW            ISOLATE DEVICE TYPE                      E.2
*A*        SUBI    0302                                                    E.2
*A*        JAP     RTCO           IF MODEL B,C, OR D                       E.2
*A*        TZA                                                             E.2
*A*RTCO    STA     SGMOD+1        SET DEVICE TYPE  (0=A/B, 1=C, 2=D)       E.2
```

```
   .REPLACE,98                    ISDA      SMR-      04/29/75  KERNS     E.2
*D*        LDA      VSBVN         SGEN SETS/USES VIRTUAL ADDR.             V2
*A*VSCRDR  EQU      0341                                                   E.2
*A*        LDA      VSCRDR        BOTTOM OF CORE RESIDENT DIRECTORY        E.2
```

```
   .REPLACE,136                    ISDB      SMR-      04/29/75  KERNS      F.2
*D*        LDA      VSBVN          SGEN SETS/USES VIRTUAL ADDR.              V2
*A*VSCRDR  EQU      0341                                                     F.2
*A*        LDA      VSCRDR         BOTTOM OF CORE RESIDENT DIRECTORY         F.2
```

```
   .REPLACE,142                    I$DC      SMR-      04/29/75  KERNS     F.2
*D*        LDA      V$BVN          SGEN SETS/USES VIRTUAL ADDR.            V2
*A*V$CRDR  EQU      0341                                                   F.2
*A*        LDA      V$CRDR         BOTTOM OF CORE RESIDENT DIRECTORY       F.2
```

```
   .REPLACE,133                    ISDD      SMR-      04/29/75  KERNS      F.2
*D*        LDA      VSBVN          SGEN SETS/USES VIRTUAL ADDR.             5S
*A*VSCRDR  EQU      0341                                                    F.2
*A*        LDA      VSCRDR         BOTTOM OF CORE RESIDENT DIRECTORY        F.2
```

```
   .INSERT,2                          V$SYTASK  V75       04/08/75  SAKAMOTO 8V75
*A*V75    SET     1                                                          V75
   .INSERT,49                         V$SYTASK  V75       04/08/75  SAKAMOTO 8V75
*A*       IFF     V75-1                                                      V75
*A*TBINC  EQU     10                 TIDB SIZE INCREMENT FOR LONG TIDB       V75
   .INSERT,524                        V$SYTASK  V75       04/08/75  SAKAMOTO 8V75
*A*       IFT     V75-1                                                      V75
*A*       GOTO    2                                                          V75
*A*       EXT     LOC7,LOC8,LOC9,LOC10,LOC11                                 V75
*A*       LDR     ERTIDB             ADDRESS OF ERROR TASK TIDB              V75
*A*       LDA     TBPL,B             FETCH LONG TIDB INDICATOR               V75
*A*       BT      RA0+R14,ERV1         JUMP ON SHORT TIDB                    V75
*A*       LDXI    MPMGR3             POINTER TO DESTINATION FIELD            V75
*A*       LDRE    LOC7               VALUE OF R3                             V75
*A*       JMPM    ASCONV             CONVERT TO ASCII AND STORE              V75
*A*       LDXI    MPMGR4                                                     V75
*A*       LDRE    LOC8                                                       V75
*A*       JMPM    ASCONV                                                     V75
*A*       LDXI    MPMGR5                                                     V75
*A*       LDBE    LOC9                                                       V75
*A*       JMPM    ASCONV                                                     V75
*A*       LDXI    MPMGR6                                                     V75
*A*       LDBE    LOC10                                                      V75
*A*       JMPM    ASCONV                                                     V75
*A*       LDXI    MPMGR7                                                     V75
*A*       LDBE    LOC11                                                      V75
*A*       JMPM    ASCONV                                                     V75
*A*       WRTTF   MPDCBV,1           OUTPUT V75 REGISTER CONTENTS            V75
*A*ERV1   EQU     *                                                          V75
*A*2      CONT                                                               V75
   .INSERT,610                        V$SYTASK  V75       04/08/75  SAKAMOTO 8V75
*A*ER2A   CONT                                                               V75
   .REPLACE,632                       V$SYTASK  V75       04/08/75  SAKAMOTO 8V75
*D*ER2A   CONT
*A*       IFT     V75-1                                                      V75
*A*       GOTO    2                                                          V75
*A*MPDCBV DCB     FV75MG-MPMSGV,MPMSGV                                       V75
*A*MPMSGV DATA    '   R3='                                                   V75
*A*MPMGR3 DATA    '012345,R4='                                               V75
*A*MPMGR4 DATA    '012345,R5='                                               V75
*A*MPMGR5 DATA    '012345,R6='                                               V75
*A*MPMGR6 DATA    '012345,R7='                                               V75
*A*MPMGR7 DATA    '012345'                                                   V75
```

```
*A*EV75MG EQU     *                                                    V75
*A*2      CONT                                                         V75
   .INSERT,1555                 VSSYTASK  SMR-      10/24/74  KERNS   E.1
*A*       STX     SAL2C        SAVE TIOB ADDR                          E.1
   .INSERT,1991                 VSSYTASK  V75       04/08/75  SAKAMOTO 8V75
*A*       TFF     V75-1                                                V75
*A*       LDB     TBPL,X       FETCH LONG TIOB INDICATOR               V75
   .INSERT,1999                 VSSYTASK  V75       04/08/75  SAKAMOTO 8V75
*A*       TFT     V75-1                                                V75
*A*       GOTO    2                                                    V75
*A*       BT      RB0+B14,*+4   JUMP ON SHORT TIOB                     V75
*A*       ADDI    TBINC        LENGTH OF LONG TIOB                     V75
*A*2      CONT                                                         V75
   .INSERT,2287                 VSSYTASK  SMR-      10/24/74  KERNS   E.1
*A*       BT      RA1+B15,SAL10F  JUMP IF RTC INTERRUPTED              E.1
*A*       BT      RA0+B0,SAL10F   JUMP IF NOT LOADED                   E.1
*A*       BT      RA1+B7,SAL10F   JUMP IF TASK IN TIME DELAY TYPE 1    E.1
   .INSERT,2398                 VSSYTASK  SMR-      03/14/75  KERNS   E.2
*A*       SUB     TWO          CHECK IF ONLY ONE WORD TOO MANY         E.2
```

```
    .INSERT,2                          VSFUNC    V75       04/08/75  SAKAMOTO 8V75
*A*V75      SET     1                                                        V75
*A*         TFT     V75-1                                                    V75
*A*         GOTO    2                                                        V75
*A*SAVER    MAC                                                              V75
*A*         ST,3    P(2),P(1)                                                V75
*A*         ST,4    P(2)+1,P(1)                                              V75
*A*         ST,5    P(2)+2,P(1)                                              V75
*A*         ST,6    P(2)+3,P(1)                                              V75
*A*         ST,7    P(2)+4,P(1)                                              V75
*A*         EMAC                                                             V75
*A*RESTOR   MAC                                                              V75
*A*         LD,3    P(2),P(1)                                                V75
*A*         LD,4    P(2)+1,P(1)                                              V75
*A*         LD,5    P(2)+2,P(1)                                              V75
*A*         LD,6    P(2)+3,P(1)                                              V75
*A*         LD,7    P(2)+4,P(1)                                              V75
*A*         EMAC                                                             V75
*A*MOVER    MAC                                                              V75
*A*         LDA     P(1),P(2)                                                V75
*A*         STA     P(3),P(4)                                                V75
*A*         LDA     P(1)+1,P(2)                                              V75
*A*         STA     P(3)+1,P(4)                                              V75
*A*         LDA     P(1)+2,P(2)                                              V75
*A*         STA     P(3)+2,P(4)                                              V75
*A*         LDA     P(1)+3,P(2)                                              V75
*A*         STA     P(3)+3,P(4)                                              V75
*A*         LDA     P(1)+4,P(2)                                              V75
*A*         STA     P(3)+4,P(4)                                              V75
*A*         EMAC                                                             V75
*A*2        CONT                                                             V75
    .INSERT,67                         VSFUNC    V75       04/08/75  SAKAMOTO 8V75
*A*         TFT     V75-1                                                    V75
*A*         GOTO    2                                                        V75
*A*         TFT     VORTEX-2                                                 V75
*A*TBRSR3   EQU     26                                                       V75
*A*         TFF     VORTEX-2                                                 V75
*A*TBRSR3   EQU     29          V75 REGISTER SAVE AREA IN                    V75
*A**                            REENTRANT STACK                             V75
*A*TBRSR4   EQU     TBRSR3+1                                                 V75
*A*TBRSR5   EQU     TBRSR4+1                                                 V75
*A*TBRSR6   EQU     TBRSR5+1                                                 V75
*A*TBRSR7   EQU     TBRSR6+1                                                 V75
```

```
*A*TBISR3 EQU     T8RSR7+1      V75 REGISTER SAVE AREA IN                  V75
*A**                            INVERRUPT STACK                           V75
*A*TBISR4 EQU     TBTSR3+1                                                V75
*A*TBISR5 EQU     TBTSR4+1                                                V75
*A*TBISR6 EQU     TBTSR5+1                                                V75
*A*TBISR7 EQU     T6ISR6+1                                                V75
*A*2      CONT                                                            V75
   ,REPLACE,153                   VSFUNC    V75       04/08/75  SAKAMOTO 8V75
*D*VSCRDM EQU     LCJP+22       CARD KEYPUNCH TYPE, 0=026, 1=029
*A*VSCRDM EQU     LCJP-2                                                  V75
   ,INSERT,552                    VSFUNC    V75       04/08/75  SAKAMOTO 8V75
*A*       IFT     V75-1                                                   V75
*A*       GOTO    2                                                       V75
*A*DTSP5  LDA     TBPL,X        CHECK FOR LONG TIDB                       V75
*A*       BT      RA0+B14,DISP5A    JUMP ON SHORT TIDB                    V75
*A*       RESTOR  X,TBTSR3      RESTORE V75 REG.S FROM INTERRUPT STK      V75
*A*DTSP5A EQU     *                                                       V75
*A*       LDA     TBTSA,1                                                 V75
*A*2      CONT                                                            V75
*A*       IFT     V75-1                                                   V75
   ,REPLACE,559,560               VSFUNC    NO SMR    06/01/75  KERNS     E.2
*D*       IFT     VORTEX-2      ASSEMBLE IF NOT VORTEX II                 V2
*D*       ANA     BM1777         AND I/O THREADED.                        V2
   ,REPLACE,575,576               VSFUNC    SMR=      12/10/74  KERNS     E,1
*D*       LDB     TBRSTS,1          JUMP IF RESIDENT TASK                 D,1
*D*       JBZ     DISAL2            JUMP IF RESIDENT TASK                 D,1
*A*       LDA     TBRSTS,1      JUMP IF RESIDENT TASK                     E,1
*A*       ANA     BM377         ISOLATE LOGICAL UNIT                      E,1
*A*       JAZ     DISAL2        JUMP IF RESIDENT TASK                     E,1
   ,REPLACE,660,664               VSFUNC    F3592     02/21/75  NEWDORF   E.2
*D*       LDA     TBKN2,1       CK IF TTY DRIVER                          V2
*D*       FRAI    'TY'                                                    V2
*D*       TAB                    IGNOR IF TT IS                           V2
*D*       LDA     TBST,1                                                  V2
*D*       JBZ     DIS18A        JMP IF TTY                                V2
*A*       LDA     TBKN2,1       CHECK DRIVER NAME                         E.2
*A*       SURI    'CT'          IS IT A CT?                               E.2
*A*       JAZ     DIS18C        JUMP IF YES                               E.2
*A*       SURI    'TY'-'CT'     IS IT A TY?                               E.2
*A*DTS18C TAB                                                             E.2
*A*       LDA     TBST,1                                                  E.2
*A*       JBZ     DIS18A        FOR TY & CT DONT RESET INTRP EXPECTED     E.2
   ,INSERT,675                    VSFUNC    V75       04/08/75  SAKAMOTO 8V75
```

```
*A*          TFT     V75-1                                                  V75
*A*          GOTO    2                                                      V75
*A*DTSP19    LDA     TBPL,X          CHECK FOR LONG TIOB                    V75
*A*          BT      RA0+B14,DSP19A     JUMP ON SHORT TIOB                  V75
*A*          RESTOR  X,TBRSR3        RESTORE V75 REGISTERS                  V75
*A*DSP19A    EQU     *                                                      V75
*A*          LDA     TBRSA,1                                                V75
*A*2         CONT                                                           V75
*A*          TFT     V75-1                                                  V75
   .INSERT,759                                                              E.2
*A*          ANAI    0177037         CLEAR STATE FLAGS                      E.2
*A*          ORA     BS5             SET STATE 0                            E.2
*A*          STA     TBTST,X                                                E.2
   .INSERT,798                      VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*          TFT     V75-1                                                  V75
*A*          GOTO    2                                                      V75
*A*          JSR     SV75RG,B        SAVE V75 REGISTERS                     V75
*A*          DATA    SR3             POINTER TO SAVE AREA                   V75
*A*2         CONT                                                           V75
   .INSERT,840                      VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*          TFF     V75-1                                                  V75
*A*          LDAF    FCLK                                                   V75
*A*          TFT     V75-1                                                  V75
   .INSERT,867                      VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*          TFT     V75-1                                                  V75
*A*          GOTO    2                                                      V75
*A*          LDA     TBPL,B          CHECK FOR LONG TIOB                    V75
*A*          BT      RA0+B14,DRT1                                           V75
*A*          JMPM    MOV75           MOVE V75 REGISTERS                     V75
*A*          DATA    SR3             POINTER TO SOURCE FIELD                V75
*A*          DATA    TBISR3          OFFSET TO DESTINATION AREA             V75
*A*DRT1      EQU     *                                                      V75
*A*2         CONT                                                           V75
   .INSERT,909                      VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*          TFT     V75-1                                                  V75
*A*          GOTO    2                                                      V75
*A*          JSR     SV75RG,B        SAVE V75 REGISTERS                     V75
*A*          DATA    SR3             POINTER TO SAVE AREA                   V75
*A*2         CONT                                                           V75
   .INSERT,977                      VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*          TFT     V75-1                                                  V75
*A*          GOTO    2                                                      V75
*A*          LDA     TBPL,B          CHECK FOR LONG TIOB                    V75
```

```
*A*          BT      RA0+B14,IHD1A1    JUMP ON SHORT TIDB                    V75
*A*          JMPM    MOV75             MOVE V75 REGISTERS                    V75
*A*          DATA    SR3               POINTER TO SOURCE AREA                V75
*A*          DATA    TBTSR3            OFFSET TO DESTINATION AREA            V75
*A*IHD1A1    EQU     *                                                       V75
*A*2         CONT                                                            V75
   .INSERT,1018                        VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*          IFF     V75-1                                                   V75
*A*IHD3A     STA     THD3A1                                                  V75
*A*          IFT     V75-1                                                   V75
   .INSERT,1036                        VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*          IFT     V75-1                                                   V75
*A*          GOTO    2                                                       V75
*A*IHD3A1    RES     0                                                       V75
*A*2         CONT                                                            V75
   .INSERT,1042                        VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*          IFF     V75-1                                                   V75
*A*SR3       DATA    0,0,0,0,0         V75 REGISTER SAVE AREA                V75
   .INSERT,1087                        VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*          IFT     V75-1                                                   V75
*A*          GOTO    2                                                       V75
*A*          LDA     TBPL,B            CHECK FOR LONG TIDB                   V75
*A*          BT      RA0+B14,CLK5A     JUMP ON SHORT TIDB                    V75
*A*          SAVER   B,TBTSR3          SAVE V75 REGISTERS                    V75
*A*CLK5A     LDAE    CLOK5+1                                                 V75
*A*2         CONT                                                            V75
   .INSERT,1243                        VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*          IFT     V75-1                                                   V75
*A*          GOTO    2                                                       V75
*A*          JSR     SV75RG,B          SAVE V75 REGISTERS                    V75
*A*          DATA    REGR3             POINTER TO SAVE AREA                  V75
*A*          LDX     REGX              RESTORE [X]                           V75
*A*2         CONT                                                            V75
   .INSERT,1301                        VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*          IFT     V75-1                                                   V75
*A*          GOTO    2                                                       V75
*A*          STA     MP2AX                                                   V75
*A*          JSR     SV75RG,B          SAVE V75 REGISTERS                    V75
*A*          DATA    REGR3             POINTER TO SAVE AREA                  V75
*A*          LDAI    *-*               RESTORE A                             V75
*A*MP2AX     RES     0                                                       V75
*A*2         CONT                                                            V75
   .INSERT,1308                        VSFUNC   V75      04/08/75  SAKAMOTO 8V75
```

```
*A*         TFF     V75-1                                                   V75
*A*         INRE    IFLAG                                                   V75
*A*         TFT     V75-1                                                   V75
   .INSERT,1312                 V$FUNC    V75      04/08/75  SAKAMOTO 8V75
*A*         TFT     V75-1                                                   V75
*A*         GOTO    2                                                       V75
*A*         JSR     SV75RG,B     SAVE V75 REGISTERS                         V75
*A*         DATA    REGR3        POINTER TO SAVE AREA                       V75
*A*         LDX     REGX         RESTORE (X)                                V75
*A*2        CONT                                                            V75
   .INSERT,1338                 V$FUNC    V75      04/08/75  SAKAMOTO 8V75
*A*         TFT     V75-1                                                   V75
*A*         GOTO    2                                                       V75
*A*ENPCK    JSR     PV75RG,B     RESTORE V75 REGISTERS                      V75
*A*         DATA    REGR3        POINTER TO SAVE AREA                       V75
*A*         LDA     REGA                                                    V75
*A*2        CONT                                                            V75
*A*         TFT     V75-1                                                   V75
   .INSERT,1346                 V$FUNC    V75      04/08/75  SAKAMOTO 8V75
*A*         TFT     V75-1                                                   V75
*A*         GOTO    2                                                       V75
*A*DTSPM    JSR     RV75RG,B     RESTORE V75 REGISTERS                      V75
*A*         DATA    REGR3        POINTER TO REGISTER SAVE AREA              V75
*A*         LDA     REGA                                                    V75
*A*2        CONT                                                            V75
*A*         TFT     V75-1                                                   V75
   .INSERT,1381                 V$FUNC    V75      04/08/75  SAKAMOTO 8V75
*A*         TFT     V75-1                                                   V75
*A*         GOTO    2                                                       V75
*A*         JSR     SV75RG,B     SAVE V75 REGISTERS                         V75
*A*         DATA    REGR3        POINTER TO REGISTER SAVE AREA              V75
*A*2        CONT                                                            V75
   .INSERT,1416                 V$FUNC    V75      04/08/75  SAKAMOTO 8V75
*A*         TFT     V75-1                                                   V75
*A*         GOTO    2                                                       V75
*A*         JSR     SV75RG,B     SAVE V75 REGISTERS                         V75
*A*         DATA    LOC7         POINTER TO SAVE AREA FOR VSERDR            V75
*A*         LDX     VSCTI        RESTORE (X)                                V75
*A*2        CONT                                                            V75
   .INSERT,1450                 V$FUNC    V75      04/08/75  SAKAMOTO 8V75
*A*         TFT     V75-1                                                   V75
*A*         GOTO    2                                                       V75
*A*         JMPM    MOV75        MOVE V75 REGISTERS                         V75
```

```
*A*          DATA    REGR3          POINTER TO SOORCE FIELD                    V75
*A*          DATA    TBISR3         OFFSET TO DESTINATION AREA                 V75
*A*2         CONT                                                              V75
   .INSERT,1477                     V$FUNC    V75      04/08/75  SAKAMOTO 8V75
*A*          IFT     V75-1                                                     V75
*A*          GOTO    2                                                         V75
*A*          NAME    LOC7,LOC8,LOC9,LOC10,LOC11                                V75
*A*LOC7      DATA    0              R3 SAVE LOCATION                           V75
*A*LOC8      DATA    0              R4 SAVE LOCATION                           V75
*A*LOC9      DATA    0              R5 SAVE LOCATION                           V75
*A*LOC10     DATA    0              R6 SAVE LOCATION                           V75
*A*LOC11     DATA    0              R7 SAVE LOCATION                           V75
*A*2         CONT                                                              V75
   .REPLACE,1519                    V$FUNC    RQBLK-V2  02/27/75  KERNS       E.2
*D*          JANZ    MPER2
*A*          JAZ     MPJP4                                                     E.2
*A*          LDAI    0406                                                      E.2
*A*          SUB     1,X                                                       E.2
*A*          JAZ     MPJP4          IF EXEC CALL                               E.2
*A*          SUB     TWO                                                       E.2
*A*          JAZ     MPJP2          IF I/O CALL                                E.2
*A*          LDAI    0373                                                      E.2
*A*          SUB     1,X                                                       E.2
*A*          JAZ     MPJP3          IF STAT CALL                               E.2
*A*          JMP     MPER2          ILLEGAL REQUEST                            E.2
*A*MPJP4     EQU     *                                                         E.2
   .INSERT,1551                     V$FUNC    SMR-      03/03/75  KERNS       E.2
*A*          LDRE    V$MPER                                                    E.2
*A*          LDA     0,B            INST CAUSING ERROR                         E.2
*A*          LSRA    8                                                         E.2
*A*          SUBI    0105                                                      E.2
*A*          JANZ    MPER1          IF NOT A BCS INST                          E.2
*A*          STB     MPJP1                                                     E.2
*A*          LDRI    011126         ERROR CODE, FX26-CONTINUE TASK             E.2
*A*          XEC     MPER2-1        ENABLE PTM                                 E.2
*A*          JMP     V$MP5A                                                    E.2
*A*MPER1     EQU     *                                                         E.2
   .INSERT,1603                     V$FUNC    V75      04/08/75  SAKAMOTO 8V75
*A*          IFT     V75-1                                                     V75
*A*          GOTO    2                                                         V75
*A*          JSR     SV75RG,B       SAVE V75 REGISTERS                         V75
*A*          DATA    REGR3          POINTER TO REGISTER SAVE AREA              V75
*A*2         CONT                                                              V75
```

```
    .INSERT,1636                    VSFUNC   V75      07/21/75  MCMURRAY  V75
*A*         LDX     REGX            RESTORE X-REG
    .INSERT,1672                    VSFUNC   V75      07/21/75  MCMURRAY  V75
*A*         LDX     REGX            RESTORE X-REG
    .INSERT,1685                    VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*         IFF     V75-1                                                 V75
*A*REGR3    DATA    0,0,0,0,0       V75 REGISTER SAVE AREA                V75
    .INSERT,1711                    VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*         IFT     V75-1                                                 V75
*A*         GOTO    2                                                     V75
*A*         JSR     SV75RG,B        SAVE V75 REGISTERS                    V75
*A*         DATA    PFONR3          POINTER TO REGISTER SAVE AREA         V75
*A*2        CONT                                                          V75
    .INSERT,1862                    VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*         IFT     V75-1                                                 V75
*A*         GOTO    2                                                     V75
*A*         JSR     RV75RG,B        RESTORE V75 REGISTERS                 V75
*A*         DATA    PFONR3          POINTER TO REGISTER SAVE AREA         V75
*A*2        CONT                                                          V75
    .INSERT,1900                    VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*         IFT     V75-1                                                 V75
*A*         GOTO    2                                                     V75
*A*         JMPM    MOV75           MOVE V75 REGISTERS                    V75
*A*         DATA    PFONR3          POINTER TO SOURCE AREA                V75
*A*         DATA    TBISP3          OFFSET TO DESTINATION AREA            V75
*A*2        CONT                                                          V75
    .INSERT,1940                    VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*         IFF     V75-1                                                 V75
*A*PFONR3   DATA    0,0,0,0,0       V75 REGISTER SAVE AREA                V75
    .INSERT,1951                    VSFUNC   V75      04/08/75  SAKAMOTO 8V75
*A*         IFT     V75-1                                                 V75
*A*         GOTO    2                                                     V75
*A*         EJEC                                                          V75
*A**        SUBROUTINE:             SV75RG                                V75
*A**                                                                      V75
*A**        PURPOSE:                SAVE V75 REGISTERS                    V75
*A**                                                                      V75
*A**        CALLING SEQUENCE:                                             V75
*A**                JSR     SV75RG,B                                      V75
*A**                DATA    ADDRESS OF SAVE AREA                          V75
*A**                                                                      V75
*A*SV75RG   EQU     *               (B) > RETURN ADDRESS                  V75
*A*         LDA     V$CROM          CHECK FOR V75                         V75
```

```
*A*         BT      RA0+B1,SV75EX    JUMP IF NOT V75                    V75
*A*         LDX     0,B              FETCH POINTER TO SAVE AREA         V75
*A*         SAVER   X,0              SAVE V75 REGISTERS                 V75
*A*SV75EX   IJMP    1,B              RETURN                             V75
*A*         FJEC                                                        V75
*A**        SUBROUTINE:              RV75RG                             V75
*A**                                                                    V75
*A**        PURPOSE:                 RESTORE V75 REGISTERS              V75
*A**                                                                    V75
*A**        CALLING SEQUENCE:                                           V75
*A**                 JSR    RV75RG,B                                    V75
*A**                 DATA   ADDRESS OF REGISTER SAVE AREA               V75
*A**                                                                    V75
*A*RV75RG   EQU     *                [B] > RETURN ADDRESS               V75
*A*         LDA     VSCROM           CHECK FOR V75                      V75
*A*         BT      RA0+B1,RV75EX    JUMP IF NOT V75                    V75
*A*         LDX     0,B              FETCH POINTER TO SAVE AREA         V75
*A*         RESTOR  X,0              RESTORE V75 REGISTERS              V75
*A*RV75EX   IJMP    1,B              RETURN                             V75
*A*         FJEC                                                        V75
*A**        SUBROUTINE:              MOV75                              V75
*A**                                                                    V75
*A**        PURPOSE:                 MOVE V75 REGISTERS FROM SOURCE TO  V75
*A**                                 DESTINATION AREA                   V75
*A**                                                                    V75
*A**        CALLING SEQUENCE:                                           V75
*A**                 JMPM   MOV75                                       V75
*A**                 DATA   ADDRESS OF SOURCE FIELD                     V75
*A**                 DATA   OFFSET TO DESTINATION AREA                  V75
*A**                 [B] > BASE ADDRESS OF DESTINATION AREA             V75
*A**                                                                    V75
*A*MOV75    ENTR                                                        V75
*A*         LDA     VSCROM           CHECK FOR V75                      V75
*A*         BT      RA0+B1,MOV75A    JUMP IF NOT V75                    V75
*A*         STB     SVB+1            SAVE [B]                           V75
*A*         STX     SVX+1            SAVE [X]                           V75
*A*         LDXE    MOV75            POINTER TO PARAMETER LIST          V75
*A*         TBA                                                         V75
*A*         ADD     1,X              FORM ADDRESS OF DESTINATION AREA   V75
*A*         TAB                                                         V75
*A*         LDX     0,X              FETCH POINTER TO SOURCE AREA       V75
*A*         MOVER   0,X,0,B          MOVE V75 REGISTERS                 V75
*A*SVB      LDBI    *                RESTORE [B]                        V75
```

```
*A*SVX      LDXI     *                RESTORE [X] REGISTER                    V75
*A*MOV75A   INRE     MOV75                                                    V75
*A*         INRE     MOV75                                                    V75
*A*         RETU*    MOV75            RETURN                                  V75
*A*2        CONT                                                              V75
```

```
   ,REPLACE,12,13                    VSIOC     V75       02/01/75  MCMURRAY  V75
*D*        TFF      VORTEX-2                                                    V2
*D*        EXT      VSMALC,VSMDAL  ALLOCATE/DEALLOCATE DYNAMIC MEMORY          V2
*A*        SPAC
*A*************************************************************************
*A**                                                                         *
*A**        V O R T E X / V O R T E X  I I  I N P U T / O U T P U T          *
*A**                                                                         *
*A**                        C O N T R O L  S Y S T E M                       *
*A**                                                                         *
*A** FUNCTION: TO PROCESS ALL I/O IN VORTEX/VORTEX II SYSTEMS.               *
*A**                                                                         *
*A*************************************************************************
*A*        SPAC
*A************
*A** ENTRIES *
*A************
*A*        SPAC
*A*        NAME     VSBIC          SET UP BIC OPERATION
*A*        NAME     VSERR          ERROR SERVICE ROUTINE
*A*        NAME     VSFNR          FINISH I/O REQUEST
*A*        NAME     VSFNRM         START UP I/O REQUEST
*A*        NAME     VSTOC          LINK I/O REQUEST
*A*        NAME     VSTOST         TEST I/O REQUEST STATUS
*A*        TFF      VORTEX-1
*A*        NAME     IOFOOD         ?
*A*        SPAC
*A**************
*A** EXTERNALS *
*A**************
*A*        SPAC
*A*        EXT      VSDISP         VORTEX DISPATCHER
*A*        EXT      VSEXEC         VORTEX EXEC SERVICE REQUESTS
*A*        TFT      VORTEX-2
*A*        GOTO     1
*A*        EXT      VSMALC         ALLOCATE DYNAMIC MEMORY
*A*        EXT      VSMDAL         DEALLOCATE DYNAMIC MEMORY
*A*1       CONT
*A*        EJEC
*A*************************************************************************
*A**                                                                         *
*A**              L I N K   I / O   R E Q U E S T S ( V S I O C )            *
*A**                                                                         *
```

```
*A** FUNCTION: TO ACCEPT I/O REQUESTS FROM TASKS, VERIFY THEM, AND       *
*A**            LINK THEM, IN ORDER OF PRIORITY OF CALLING TASK, TO A    *
*A**            CHAIN WHOSE HEAD IS IN THE CONTROLLER TABLE OF REQUESTED  *
*A**            I/O DEVICE.                                              *
*A**                                                                     *
*A** ENTRY: DIRECT TO VSIOC                                              *
*A**        VSCTL = USER TTDB ADDRESS                                    *
*A**        X = ADDRESS OF REQUEST BLOCK                                 *
*A**        VSST0,VSST1,VSST2,VSST3 SET TO USER MAP(VORTEX IT)           *
*A**                                                                     *
*A**        VORTEX   : BACKGROUND: THRU MEMORY PROTECT PROCESSOR VSMPJP  *
*A**                   FOREGROUND: DIRECT TO EXTERNAL LABEL VSIOC        *
*A**                                                                     *
*A**        VORTEX II: ALL TASKS: THRU LOW-MEM CELLS 0404,0405,          *
*A**                              WHICH CONTAIN 'JMP VSIOC'              *
*A**                                                                     *
*A**                   MAP 0:   DIRECT TO LABEL VSTOC.                   *
*A**                                                                     *
*A**                   LEVEL 1: THRU MAP ERROR(I/O INSTRUCTION)          *
*A**                            PROCESSOR VSMP3 WHEN 1ST INSTRUCTION     *
*A**                            OF VSIOC(EXC) IS EXECUTED.               *
*A**                                                                     *
*A**                   OTHERS ' THRU MAP ERROR(JUMP) PROCESSOR VSMAP2    *
*A**                            WHEN JSR 404 TO PROTECTED PAGE 0 IS      *
*A**                            EXECUTED'.                               *
*A**                                                                     *
*A** CALLING SEQUENCE: VORTEX   : JSR  VSIOC,X                           *
*A**                              5-WORD REQUEST BLOCK                   *
*A**                                                                     *
*A**                   VORTEX II: JSR  0404,X                            *
*A**                              5-WORD REQUEST BLOCK                   *
*A**                                                                     *
*A** EXIT : VSENRM: NORMAL EXIT                                          *
*A**        VSDISP: ERROR OR IF LUN=DUM                                  *
*A**        REGS SAVED IN USER TTDB                                      *
*A**        VSKEY = 0(VORTEX IT)                                         *
*A*       EJEC
*A** ERRORS: ABORT CALLING TASK IF BACKGROUND LEVEL 0                    *
*A**                                                                     *
*A**          I001: DEVICE DOWN                                          *
*A**          I002: INVALID LUN                                          *
*A**          I003: DCB/FCB PARAMETER ERROR                              *
*A**          I004: PROTECT KEY ERROR                                    *
```

```
*A**          IO05: BACKGROUND LEVEL 0 READ INTO PROTECTED MEMORY          *
*A**          IO06: REQUEST BLOCK COMPLETE BIT C RESET                     *
*A**          IO07: READ FROM WRITE-ONLY DEVICE AND VICE-VERSA             *
*A**                                                                       *
*A*************************************************************************
*A*        SPAC
*A*        EJEC
*A*************************************************************************
*A**                                                                       *
*A**       F O R M A T   O F   U S E R   R E Q U E S T   B L O C K        *
*A**                                                                       *
*A**                 O N   E N T R Y   T O   V S I O C                     *
*A**                                                                       *
*A**                                                                       *
*A**          15 14    12 11     9  8  7     5  4           0             *
*A**         ***************************************************           *
*A**         *  *                 *  *         *             *            *
*A**         *C * --------------- *--* ------- * ----------- *   WORD 0   *
*A**         *  *                 *  *         *             *            *
*A**         ***************************************************           *
*A**         *  *        *           *                       *            *
*A**         *W *  MODE  *    OP     *        L U N          *   WORD 1   *
*A**         *  *        *           *                       *            *
*A**         ***************************************************           *
*A**         *                                               *            *
*A**         *        D C B / F C B   A D D R E S S          *   WORD 2   *
*A**         *                                               *            *
*A**         ***************************************************           *
*A**         *                                               *            *
*A**         * ------------------------------------------- *   WORD 3   *
*A**         *                                               *            *
*A**         ***************************************************           *
*A**         *                                               *            *
*A**         * ------------------------------------------- *   WORD 4   *
*A**         *                                               *            *
*A**         ***************************************************           *
*A**                                                                       *
*A**       C    : COMPLETE BIT: MUST BE 1                                  *
*A**       W    : WATT FLAG: =1: DONT SUSPEND CALLING TASK                 *
*A**                          0: SUSPEND CALLING TASK                      *
*A*        EJEC
*A**       MODE: USED BY DRIVER. STANDARD VORTEX MODES ARE:                *
*A**                                                                       *
```

```
*A**                0: SYSTEM BINARY                                        *
*A**                1: ASCII                                                *
*A**                2: BCD                                                  *
*A**                3: UNFORMATTED(NOT FORTRAN UNFORMATTED)                 *
*A**                                                                        *
*A**        OP   : OPERATION SPECIFIER. STANDARD VORTEX OPS ARE:            *
*A**                                                                        *
*A**                0: READ                                                 *
*A**                1: WRITE                                                *
*A**                2: WRITE EOF                                            *
*A**                3: REWIND                                               *
*A**                4: SKIP RECORD                                          *
*A**                5: FUNCTION                                             *
*A**                6: OPEN                                                 *
*A**                7: CLOSE                                                *
*A**                                                                        *
*A**        LUN  : LOGICAL UNIT NUMBER                                      *
*A**                                                                        *
*A**                                                                        *
*A*************************************************************************
*A*         EJEC
*A*************************************************************************
*A**                                                                        *
*A**        F O R M A T   O F   U S E R   R E Q U E S T   B L O C K         *
*A**                                                                        *
*A**        O N   E X I T   F R O M   V $ I O C   T O   V $ E N R M         *
*A**                                                                        *
*A**                                                                        *
*A**          15 14    12 11     9  8  7     5  4           0              *
*A**         ***************************************************            *
*A**         *  *                 *  *         *  USER       *              *
*A**         *0 *        0        *0 *    0    *  TASK       *   WORD 0     *
*A**         *  *                 *  *         *  PRIORITY   *              *
*A**         ***************************************************            *
*A**         *  *        *           *                        *              *
*A**         *W *  MODE  *    OP     *        L U N           *   WORD 1     *
*A**         *  *        *           *                        *              *
*A**         ***************************************************            *
*A**         *                                                *              *
*A**         *        D C B / F C B   A D D R E S S           *   WORD 2     *
*A**         *                                                *              *
*A**         ***************************************************            *
*A**         *                                                *              *
```

```
*A**        *      U S E R   T I O B   A D D R E S S        *   WORD 3   *
*A**        *                                               *            *
*A**        ***************************************************            *
*A**        *                                               *            *
*A**        *  L I N K   T O   N E X T   R E Q   B L O C K  *   WORD 4   *
*A**        *                                               *            *
*A**        ***************************************************            *
*A**        *                                               *            *
*A**        *R E Q B L K   A D D R   I N   U S E R   T A S K*   WORD 5   *
*A**        *                                               *            *
*A**        ***************************************************            *
*A**                                                                     *
*A**      NOTE: WORD 5 EXISTS ONLY IN TEMP AREA, WHEN RQBLK IS MOVED      *
*A**            BY V$IOC FROM USER TASK INTO TEMP AREA. IN VORTEX, TEMP   *
*A**            AREA IS IN V$IOC MODULE. IN VORTEX II, TEMP AREA IS       *
*A**            ALLOCATED DYNAMICALLY AND MAPPED INTO MAP 0. VORTEX       *
*A**            MOVES ONLY BACKGROUND REQUEST BLOCKS INTO TEMP AREA.      *
*A**            VORTEX II MOVES ALL EXCEPT MAP 0 REQUEST BLOCKS INTO      *
*A**            THE TEMP AREA.                                            *
*A**                                                                     *
*A*********************************************************************
*A*       NLIS
*A*       SPAC
*A*V75    SET     1                                                      V75
   .REPLACE,115                  V$IOC     BICTABLE  03/12/75  NEWDORF   E.2
*D*V$BIC1 EQU     LCJP+6        BIC INTERRUPT ADDRESS TABLE  (10 WORDS)
   .REPLACE,119,121              V$IOC     V75       02/01/75  MCMURRAY  V75
*D*V$CRDM EQU     LCJP+22       CARD KEYPUNCH TYPE, 0=026, 1=029
*D**                            BIT 0 = SYSTEM NOMINAL KEYPUNCH MODE.
*D**                            BIT 9 = CURRENT JOB KEYPUNCH MODE.
*A*V$CRDM EQU     LCJP-2        KP MODE(0=026,1=029)                     V75
*A**                            BIT 0 = PERMANENT KP MODE                V75
*A**                            BIT 8 = CURRENT KP MODE                  V75
*A**                            BIT 1 = 1 IF V75 SYSTEM                  V75
   .REPLACE,401                  V$IOC     BICTABLE  03/12/75  NEWDORF   E.2
*D**   1 ENTRY FOR EACH BIC ,EACH CONTROLLER TABLE,CTBL, WHICH
*A**   TWO WORD ENTRY FOR EACH BIC.                                      E.2
*A**    EACH CONTROLLER TABLE, CTBL, WHICH                               E.2
   .INSERT,403                                                           E.2
*A**  CTBICB-1  CONTAINS THE TRAP LOCATION FOR THAT BIC                  E.2
   .INSERT,496                   V$IOC     V75       02/01/75  MCMURRAY  V75
*A*       SPAC                                                           V75
*A*ARST   SET     040           A-REGISTER BIT RESET                     V75
```

```
*A*TB75Z  EQU   1                                                V75
*A*TB75B  EQU   14                                               V75
*A*TB75   EQU   2                                                V75
*A*       IFF   VORTEX-1                                         V75
*A*TBRSRO EQU   TBRSF+2                                          V75
*A*       IFF   VORTEX-2                                         V75
*A*TBRSRO EQU   TBTST+1                                          V75
*A*       SPAC                                                   V75
*A***********                                                    V75
*A** MACROS *                                                    V75
*A***********                                                    V75
*A*       SPAC                                                   V75
*A*********************************                             V75
*A** TEST BIT/RESET/IN A/B-REF *                                 V75
*A*********************************                             V75
*A*TRAP    MAC                                                  V75
*A*       LDA   P(1),B                                           V75
*A*       RT    ARST+P(2),P(3)                                   V75
*A*       EMAC                                                   V75
*A*       SPAC                                                   V75
*A*********************                                         V75
*A** STORE V75 REGS *                                           V75
*A*********************                                         V75
*A*       SPAC                                                   V75
*A*ST75    MAC                                                  V75
*A*       TFT   P(0)-1                                           V75
*A*       GOTO  STMAC1                                           V75
*A*       ST,3  P(1)                                             V75
*A*       ST,4  P(1)+1                                           V75
*A*       ST,5  P(1)+2                                           V75
*A*       ST,6  P(1)+3                                           V75
*A*       ST,7  P(1)+4                                           V75
*A*       GOTO  STMACZ                                           V75
*A*STMAC1 CONT                                                   V75
*A*       TFT   P(0)-2                                           V75
*A*       GOTO  STMAC2                                           V75
*A*       ST,3  P(1),P(2)                                        V75
*A*       ST,4  P(1)+1,P(2)                                      V75
*A*       ST,5  P(1)+2,P(2)                                      V75
*A*       ST,6  P(1)+3,P(2)                                      V75
*A*       ST,7  P(1)+4,P(2)                                      V75
*A*       GOTO  STMACZ                                           V75
*A*STMAC2 CONT                                                   V75
```

```
*A*         ST*,3    P(1),P(2)                                                  V75
*A*         ST*,4    P(1)+1,P(2)                                                V75
*A*         ST*,5    P(1)+2,P(2)                                                V75
*A*         ST*,6    P(1)+3,P(2)                                                V75
*A*         ST*,7    P(1)+4,P(2)                                                V75
*A*STMACZ   CONT                                                                V75
*A*         FMAC                                                                V75
*A*         SPAC                                                                V75
   .REPLACE,507,508              VSTOC    V75     02/01/75  MCMURRAY  V75
*D*         NAME     VSTOC
*D*         EXT      VSDISP
   .INSERT,512                   VSTOC    V75     02/01/75  MCMURRAY  V75
*A*         LIST
*A*         SPAC
   .REPLACE,519                                                             E.2
*D*         EXC2     0500+MAP                                              V2
   .INSERT,520                                                              E.2
*A*         JMPM     VSSMS        SAVE MAP STATUS                         E.2
   .INSERT,521                                                              E.2
*A*         TFT      VORTEX-1                                              E.2
*A*         GOTO     1                                                     E.2
   .INSERT,527                                                              E.2
*A*         TFF      VORTEX-1                                              E.2
*A*1        CONT                                                           E.2
*A*         TFF      V75                                                   V75
*A*         GOTO     1                                                     V75
*A*         TRAB     T575,TB75B,IOC00      LONG TIDB ?                     V75
*A*         ST75     TBRSR0,B      YES. STORE EXTRA REGS                   V75
*A*IOC00    EQU      *                                                     V75
*A*1        CONT                                                           V75
   .REPLACE,878,879              VSTOC    V75     02/01/75  MCMURRAY  V75
*D*         NAME     VSFNRM
*D*         NAME     VSFNR
   .REPLACE,1043,1044            VSTOC    V75     02/01/75  MCMURRAY  V75
*D*         TFF      VORTEX-1                                              V2
*D*         NAME     TOFOOD
   .REPLACE,1288                 VSTOC    SMR-808 03/24/75  KERNS     E.2
*D*         STA      RADNR+1,B    CLEAR FSRQBK SLOT
   .REPLACE,1303                 VSTOC    SMR-619 09/26/74  KERNS     E.1
*D*         LDA      RQPWD,X      SAVE LUN OF CLOSE/UPDATE REQUEST
*A*         LDA      RQPWD,B      SAVE LUN OF CLOSE/UPDATE REQUEST         E.1
   .REPLACE,1311                 VSTOC    SMR-619 09/26/74  KERNS     E.1
*D*         JAZ      TOF42        SAME. CHECK NEXT REQUEST.
```

```
*A*         JANZ    TOF45           IF NOT SAME, REQUEUE                    E.1
*A*         LDA     RSTPR,B                                                 E.1
*A*         ANA     BM37            ISOLATE PRIORITY                        E.1
*A*         OAR                                                             E.1
*A*         JANZ    TOF42           IF NOT SYSTEM BL (FMAIN, LMGEN, SMAIN)  E.1
*A*IOF45    EQU     *                                                       E.1
   .REPLACE,1362                    VSIOC     V75      02/01/75  MCMURRAY  V75
*D*         NAME    VSERR
   .INSERT,1413                                                             F.2
*A*         SUBI    020             IS THIS A                               E.2
*A*         JAN     IOERR3          BIC ERROR                               E.2
*A*         SUBI    010             20 - 27 ?                               E.2
*A*         JAP     IOERR3          JUMP IF NOT.                            E.2
*A*         LDA     RSTPR,B         ORIGINAL STATUS WORD                    E.2
*A*         ANAI    0100777         PRESERVE PRIORITY + COMPLETION.         E.2
*A*         ORAI    020000          CREATE IO20                             E.2
*A*         STA     RSTPR,B         STORE NEW STATUS WORD                   E.2
*A*IOERR3   LDA     RSTPR,B                                                 E.2
*A*         ISRA    9                                                       E.2
*A*         ANA     BM77            ERROR CODE                              E.2
   .REPLACE,1552                    VSIOC     V75      02/01/75  MCMURRAY  V75
*D*         NAME    VSIOST
   .REPLACE,1640                    VSIOC     SMR-     03/24/75  NEWDORF   E.2
*D**        REGISTER EQUAL TO BIC NO. COMPLEMENTED.
*A**        REGISTER > BIC DA COMPLEMENTED                                  E.2
   .REPLACE,1652                    VSIOC     SMR-     03/24/75  NEWDORF   E.2
*D**                EXIT - UNSUCCESSFUL,A= BIC NUMBER COMPLEMENTED
*A**                EXIT UNSUCCESSFUL, A > BIC DA COMPLEMENTED              E.2
   .REPLACE,1660                    VSIOC     V75      02/01/75  MCMURRAY  V75
*D*         NAME    VSBIC
   .REPLACE,1673,1674               VSIOC     BICTABLE 03/12/75  NEWDORF   E.2
*D*         ISRA    1
*D*         ANA     SEVEN                                                   C
   .REPLACE,1676,1677               VSIOC     BICTABLE 03/12/75  NEWDORF   E.2
*D*         TAY                     NOW BIC NO. 0-3.
*D*         LDX     VSBIC1,X        BIC INTERRUPT TRAP ADDR.
*A*         LDXE    CTBICB,X        ADDR OF ENTRY IN BIC FLAG TABLE         E.2
*A*         DXR                                                             E.2
*A*         LDY     0,Y             BIC INTERRUPT TRAP LOC                  E.2
   .INSERT,1728                                                             E.2
*A*         IFT     VORTEX-2                                                E.2
*A*         GOTO    2                                                       E.2
*A*         EJEC                                                            E.2
```

```
*A** * * *
```

```
   .INSERT,1                          V$SERV    V75        02/18/75  MEIR      V75
*A*V75      SET      1                   V75 FLAG                                V75
   .INSERT,65                         V$SERV    V75        02/14/75  MEIR      V75
*A*TBEND    SET      TBRSE+2                                                     V75
*A*         IFF      VORTEX-2                                                    V75
*A*TBEND    SET      TBIST+1                                                     V75
*A*TBRSR3   EQU      TBEND+0 R3        REENTRANT AND SUSPEND STACK               V75
*A*TBRSR4   EQU      TBEND+1 R4        REENTRANT AND SUSPEND STACK               V75
*A*TBRSR5   EQU      TBEND+2 R5        REENTRANT AND SUSPEND STACK               V75
*A*TBRSR6   EQU      TBEND+3 R6        REENTRANT AND SUSPEND STACK               V75
*A*TBRSR7   EQU      TBEND+4 R7        REENTRANT AND SUSPEND STACK               V75
*A*TBISR3   EQU      TBEND+5 R3        INTERRUPT STACK                           V75
*A*TBISR4   EQU      TBEND+6 R4        INTERRUPT STACK                           V75
*A*TBISR5   EQU      TBEND+7 R5        INTERRUPT STACK                           V75
*A*TBISR6   EQU      TBEND+8 R6        INTERRUPT STACK                           V75
*A*TBISR7   EQU      TBEND+9 R7        INTERRUPT STACK                           V75
   .REPLACE,103                 V$SERV  V75      2/14/75 MEIR                    V75
*D**        BIT 14   -  UNUSED
*A**                 BIT 14 IF ON A LONG TIDB(FOR V75) IS USED                   V75
*A*TB75     EQU      2         WORD 2                                            V75
*A*TB75B    EQU      14        BIT 14                                            V75
*A*TB75Z    EQU      1         1 BIT LONG                                        V75
   .REPLACE,145                       V$SERV    V75        02/18/75  MEIR      V75
*D*V$CRDM EQU      LCJP+22           CARD KEYPUNCH TYPE, 0=026, 1=029
*A*V$CRDM   EQU      LCJP-2    CARD KEYPUNCH TYPE, 0= 026, 1= 029                V75
*A*LC$75    EQU      V$CRDM    V75 FLAG (BIT 1) SET BY SYSGEN                    V75
*A*LC$75B   EQU      1         BIT 1                                V75          V75
*A*LC$75Z   EQU      1         1 BIT LONG                                        V75
   .INSERT,411                        V$SERV    V75        02/14/75  MEIR      V75
*A**        SAVE AND RESTORE V75 REGISTERS                                       V75
*A*SAVER    MAC                                                                  V75
*A*         ST,3     P(2),P(1)                                                   V75
*A*         ST,4     P(2)+1,P(1)                                                 V75
*A*         ST,5     P(2)+2,P(1)                                                 V75
*A*         ST,6     P(2)+3,P(1)                                                 V75
*A*         ST,7     P(2)+4,P(1)                                                 V75
*A*         EMAC                                                                 V75
*A*RESTOR   MAC                                                      V75          V75
*A*         LD,3     P(2),P(1)                                       V75          V75
*A*         LD,4     P(2)+1,P(1)                                                 V75
*A*         LD,5     P(2)+2,P(1)                                                 V75
*A*         LD,6     P(2)+3,P(1)                                                 V75
*A*         LD,7     P(2)+4,P(1)                                                 V75
```

```
*A*          EMAC                                                          V75
*A*TESTF     MAC                                    V75                    V75
*A*          LDA     P(2),P(1)        PICK UP WORD            V75          V75
*A*          ANA     BSO+P(3)         CONTAINING FLAG         V75          V75
*A*          EMAC                                             V75          V75
*A*MOVER     MAC                                              V75          V75
*A*          LDA     P(1),P(2)        R3                                   V75
*A*          STA     P(3),P(4)                                             V75
*A*          LDA     P(1)+1,P(2)      R4                                   V75
*A*          STA     P(3)+1,P(4)                                           V75
*A*          LDA     P(1)+2,P(2)      R5                                   V75
*A*          STA     P(3)+2,P(4)                                           V75
*A*          LDA     P(1)+3,P(2)      R6                                   V75
*A*          STA     P(3)+3,P(4)                                           V75
*A*          LDA     P(1)+4,P(2)      R7                                   V75
*A*          STA     P(3)+4,P(4)                                           V75
*A*          EMAC                                                          V75
   .INSERT,469                    V$SERV    V75       02/14/75  MEIR       V75
*A*          IFT     V75-1                                                 V75
*A*          GOTO    1                                                     V75
*A*          TESTF   B,TB75,TB75B,TB75Z      TEST IF A LONG TIDB           V75
*A*          JAZ     EX11             JMP IF NOT                           V75
*A*          SAVFR   B,TBRSR3         SAVE R3 THRU R7                      V75
*A*EX11      EQU     *                                                     V75
*A*1         CONT                                                          V75
   .REPLACE,479,480               V$SERV    SMR-792   04/01/75  NEWDORF   E.2
*D*          EXC     ENAPTM         ENABLE PTMS AND CLK                    V2
*D*          EXC     ENACLK                                                V2
   .INSERT,490                    V$SERV    SMR-792   04/01/75  NEWDORF   E.2
*A*          TXA                      LOAD A WITH REQUEST CODE             E.2
*A*          DAR                                                           E.2
*A*          ADDI    04540            CREATE LLSR & %REQUEST CODE -1< BITS  E.2
*A*          STA     SERV2                                                 E.2
*A*          LDBI    040004           ENABLE MASK FOR CODES 1 THRU 16      E.2
*A*          TZA                      ENABLE MASK FOR CODES 17 THRU 32     E.2
*A*SERV2     LLSR    0                                                     E.2
*A*          BT      RB1,SERV3        JUMP IF B BIT 0 # 1                  E.2
*A*          EXC     ENACLK                                                E.2
*A*          EXC     ENAPTM                                                E.2
*A*SERV3     LDB     V$CTL            RESTORE B REG                        E.2
   .INSERT,555                    V$SERV    V75       02/14/75  MEIR       V75
*A*          LDA     V$KEY                                                 V75
   .INSERT,616                    V$SERV    V75       02/18/75  MEIR       V75
```

```
*A**                          TBPL- BITS 15,13 TO 5 ZERO. 5 TO ZERO CONTAIN
*A**                          PRIORITY LEVEL. BIT 14 (TB75) IS SET IF        V75
*A**                          A LONG TIDB IS REQUESTED.                      V75
   .INSERT,664                     V$SERV    V75        02/18/75  MERI       V75
*A*        IFT      V75-1                                                    V75
*A*        GOTO     1                                                        V75
*A*        LDA      LC$75   V75 FLAG IN LOW CORE SET BY SYSGEN               V75
*A*        BT       RA0+B1,SCH1      JMP IF NOT LONG TIDB                    V75
*A*        LDA      BS14                                                     V75
*A*        ORA      TB75,X  SET TB75 FLAG IN NEW TIDB.LONG TIDB              V75
*A*        STA      TB75,X                                                   V75
*A**                IF ALSO CALLER TIDB IS LONG, PASS PARAMETERS             V75
*A**                SAVED IN REGISTERS R3 THRU R7                            V75
*A*        TESTF    B,TB75,TB75B,TB757                                       V75
*A*        JAZ      SCH1    JMP OUT IF NO                    V75             V75
*A*        MOVER    TBRSR3,B,TBRSR3,X                                        V75
*A*SCH1    EQU      *                                                        V75
*A*1       CONT                                                              V75
   .REPLACE,681,682                V$SERV    SMR-       10/23/74  HERNANDEXE.1
*D*        LDA      1,2            STORE KEY AND LIB LU IN TEMP STORAGE
*D*        STA      TBRSTS,1
*A*        LDA      1,2            GET KEY AND LIB LUN                      E,1
   .REPLACE,752,753                V$SERV    SMR-       10/23/74  HERNANDEZE,1
*D*        STA      SCDA           SAVE A (KEY AND LUN NUMBER)              D,1
*D*        STB      SCDB           SAVE B                                   D,1
   .REPLACE,757                    V$SERV    SMR-       10/23/74  HERNANDEZE.1
*D*        JMP      SCDOUT-1       LUN = 0  OK AS IT IS A CL TASK           E,
*A*        JMP      SCDOUT         LUN = 0 OK AS IT IS A RESIDENT TASK      E,1
   .REPLACE,759                    V$SERV    SMR-       10/23/74  HERNANDEZE,1
*D*        STA      SCDC           SAVE LUN                                 D,1
*A*        STA      TBRSTS,X       SAVE LUN                                 E,1
   .REPLACE,764                    V$SERV    SMR-       10/23/74  HERNANDEZE,1
*D*        SUB      SCDC                                                    D,1
*A*        SUB      TBRSTS,X                                                E,1
   .REPLACE,767                    V$SERV    SMR-       10/23/74  HERNANDEZE,1
*D*        ADD      SCDC           GET DST ADDRESS                          D,1
*A*        ADD      TBRSTS,X       GET DST ADDRESS                          E,1
   .REPLACE,770                    V$SERV    SMR-       10/23/74  HERNANDEZE,1
*D*SCDST1  STA      SCDC           101 LEQ LUN LEQ 179?                     D,1
*A*SCDST1  STA      TBRSTS,X       101 LEQ LUN LEQ 179?                     E,1
   .REPLACE,775                    V$SERV    SMR-       10/23/74  HERNANDEZE,1
*D*        SUB      SCDC                                                    D,1
*A*        SUB      TBRSTS,X                                                E.1
```

```
   .REPLACE,778                    V$SERV    SMR-      10/23/74  HERNANDEZE,1
*D*        ADD      SCDC                                                D,1
*A*        ADD      TBRSTS,X                                            E,1
   .REPLACE,782                    V$SERV    SMR-      10/23/74  HERNANDEZE,1
*D*SCDST2  STA      SCDC           180 LEQ LUN LEQ MAX                  D,1
*A*SCDST2  STA      TBRSTS,X       180 LEQ LUN LEQ MAX                  E,1
   .REPLACE,785                    V$SERV    SMR-      10/23/74  HERNANDEZE,1
*D*        SUB      SCDC                                                D,1
*A*        SUB      TBRSTS,X                                            E,1
   .REPLACE,788                    V$SERV    SMR-      10/23/74  HERNANDEZE,1
*D*        ADD      SCDC                                                D,1
*A*        ADD      TBRSTS,X                                            E,1
   .REPLACE,794                    V$SERV    SMR-      10/23/74  HERNANDEZE,1
*D*        STA      SCDC           CONVERT TO                           D,1
*A*        STA      TBRSTS,X       CONVERT TO                           E,1
   .REPLACE,796                    V$SERV    SMR-      10/23/74  HERNANDEZE,1
*D*        ADD      SCDC           DISPLACEMENT                         D,1
*A*        ADD      TBRSTS,X       DISPLACEMENT                         E,1
   .REPLACE,801                    V.SERV    SMR-      10/23/74  HERNANDEZE,1
*D*        LDB      SCDB           RESTORE B                            D,1
   .REPLACE,803,805                V$SERV    SMR-      10/23/74  HERNANDEZE,1
*D*        TFF      VORTEX-2                                            D,1
*D*        OME      MAP,V$ST2                                           D,1
*D*        JMP      SCED3                                               D,1
*A*        STA      TBRSTS,X       SAVE ERROR FLAG                      E,1
*A*        LDB      V$CTL          GET TIDB LOC. OF CALLER              E,1
*A*        TFT      VORTEX-2                                            E,1
*A*        GOTO     S3                                                  E,1
*A*        ROF                     RESET FLAG FOR NOT MAP 0             E,1
*A*        LDA      TBKEY,B        CHECK IF MAP 0 TASK                  E,1
*A*        LDB      TBRSX,B        PARAMETER LIST LOC.                  E,1
*A*        ANA      BM17                                                E,1
*A*        JAZ      *+5            JUMP IF MAP 0                        E,1
*A*        SOF                     SET FLAG FOR NOT MAP 0 TASK          E,1
*A*        OME      MAP,V$ST2                                           E,1
*A*S3      CONT                                                         E,1
*A*        TFT      VORTEX-2                                            E,1
*A*        LDB      TBRSX,B        PARAMETER LIST LOC.                  E,1
*A*        LDB      1,B            GET KEY AND LIB LUN                  E,1
*A*        TFF      VORTEX-2                                            E,1
*A*        OME      MAP,V$ST0                                           E,1
*A*        LDA      TBRSTS,X       RESTORE ERROR FLAG                   E,1
*A*        STB      TBRSTS,X       SAVE KEY AND LUN IN TEMP LOC.        E.1
```

```
*A*        LDB     V$CTL          CALLER TIDB                                  E,1
*A*        LDB     TBRSX,B        RESTORE B TO PARAMETER LIST LOC.             E,1
*A*        IFT     VORTEX-2                                                    E,1
*A*        GOTO    S4                                                          E,1
*A*        IOFN    SCED3          MAP 0 TASK                                   E,1
*A*        OME     MAP,V$ST2      NOT MAP 0, SWITCH TO EXEC STATE 2            E,1
*A*S4      CONT                                                                E,1
*A*        JMP     SCED3                                                       E,1
   .REPLACE,812,814              V$SERV    SMR-      10/23/74  HERNANDEZE,1
*D*SCDA    DATA    0                                                           D,1
*D*SCDB    DATA    0                                                           D,1
*D*SCDC    DATA    0                                                           D.1
   .REPLACE,907                  V$SERV    SMR-753   02/26/75  KERNS          E.2
*D*        JMP     V$DP1                                                       E
*A*        JMP     SUSP3                                                       E.2
   .INSERT,915                   V$SERV    SMR-753   02/26/75  KERNS          E.2
*A*SUSP3   EQU     *                                                           E.2
*A*        JX7     V$DISP         IF AT END OF TIDB THREAD                     E.2
   .INSERT,1110                  V$SERV    V75       11/20/74  MEIR           V75
*A*        TAB                                                                 V75
*A*        SUB     STX                                                         V75
*A*        JAP     ALOCO    IF VALUE IS GRATER THAN OR EQUAL TO 5              V75
*A*        LDB     FIVE     DEFAULT TO A VALUE OF FIVE                         V75
*A*ALOCO   TBA                                                                 V75
   .INSERT,1112                  V$SERV    V75       02/18/75  MEIR           V75
*A*        IFT     V75-1                                                       V75
*A*        GOTO    1                                                           V75
*A*        LDX     V$CTL    TIDB                                               V75
*A*        IFSTF   X,TB75,TB75B,TB757       V75 FLAG(LONG TIDB)                V75
*A*        JAZ     ALO                                                         V75
*A*        TBA                                                                 V75
*A*        ADD     FIVE     LARGER SIZE                                        V75
*A*        TAB                                                                 V75
*A*ALO     TBA                                                                 V75
*A*1       CONT                                                                V75
   .INSERT,1128                  V$SERV    V75       02/18/75  MEIR           V75
*A**                SAVE V75 REGISTERS R3-R7 IN TOP OF STACK                   V75
*A*        IFT     V75-1                                                       V75
*A*        GOTO    1                                                           V75
*A*        IFSTF   B,TB75,TB75B,TB757                                          V75
*A*        JAZ     ALOC3    NO. JMP                                            V75
*A*        LDA     V$PTVB   1ST FREE ENTRY ABOVE STACK                         V75
*A*        SUB FIVE                                                            V75
```

```
*A*          TAX          STORE HERE R3 THRU R7                              V75
*A*          MOVER   TBRSR3,B,0,X                                            V75
*A*          LDX     V$CRS          RESTORE STACK POINTER                    V75
*A*ALOC3     EQU     *                                                       V75
*A*1         CONT                                                            V75
   .INSERT,1168                    V$SERV   V75      02/19/75  MEIR         V75
*A*          IFT     V75-1                                                   V75
*A*          GOTO    1                                                       V75
*A*          LDB     V$CTL   TIDB ADDRESS                                    V75
*A*          TESTF   B,TB75,TB758,TB757     LONG TIDB?                       V75
*A*          JAZ     OALC2   JMP IF NO                                       V75
*A**                 RESTORE V75 REGISTERS FROM LAST 5 LOC OF STACK          V75
*A*          LDA     V$PTVB  FIRST AVAILABLE ENTRY                           V75
*A*          SUB     FIVE                                                    V75
   .REPLACE,1458                   V$SERV   SMR-753   02/26/75  KERNS       E.2
*D*       JMP     V$DP1                                                      E
*A*       JMP     SUSP3                                                      E,2
*A*          TAX                                                             V75
*A*          RESTOR  X,0                                                     V75
*A*OALC2     EQU     *                                                       V75
*A*1         CONT                                                            V75
*A*1         CONT                                                            V75
   .REPLACE,1579,1580              V$SERV   V75      02/19/75  MEIR         V75
*D*       JSR     V$MALC,X       ALLOCATE MEMORY                            V2
*D*       DATA    TBEND+64       NUMBER OF WDS,29/TIDB 64/MAP               V2
*A**                 ALLOCATE MEMORY FOR BOTH TIDB AND MAP                   V75
*A*          IFT     V75-1                                                   V75
*A*          GOTO    1                                                       V75
*A*          LDA     LC$75   LOW CORE V75 FLAG                               V75
*A*          BT      RA0+81,ALT1      JMP IF NOT V75          V75            V75
*A*          JSR     V$MALC,X         ALLOCATE MEMORY. LONG TIDB             V75
*A*          DATA    TREND+74         39 WORDS FOR TIDB, 64 /MAP             V75
*A*          JMP     ALT2                                                    V75
*A*1         CONT                                                            V75
*A*ALT1      JSR     V$MALC,X         ALLOCATE MEMORY REGULAR TIDB           V75
*A*          DATA    TREND+64         29 WORDS FOR TIDB,64/MAP               V75
*A*ALT2      EQU     *                                                       V75
   .REPLACE,1590,1591              V$SERV   V75      02/19/75  MEIR         V75
*D*       TXA                                                               V2
*D*       ADDI    TBEND            TBEND IS END OF TIDB                     V2
*A**                 SET STARTING ADDRESS FOR MAP IMAGE FOLLOWINGTIDB        V75
*A*          IFT     V75-1                                                   V75
*A*          GOTO    2                                                       V75
```

```
*A*          LDA     LC$75   LOW CORE V75 FLAGE                              V75
*A*          BT      RA0+B1,ALT3     JMP IF NOT V75                          V75
*A*          TXA                                                             V75
*A*          ADDI    TREND+10        END OF LONG TIDB                        V75
*A*          JMP     ALT4                                                    V75
*A*2         CONT                                                            V75
*A*ALT3      TXA                                                             V75
*A*          ADDI    TREND   END OF TIDB                                     V75
*A*ALT4      EQU     *                                                       V75
   .INSERT,1706                 V$SERV  SMR-    11/11/74  KERNS             E.1
*A*          JAZ     TBSR1       IT IS SAL                                   E.1
   .INSERT,1804                 V$SERV  NO SMR  03/05/75  KERNS             E.2
*A*          LDB     VSCTL       TASK TIDB ADDR                              E.2
   .INSERT,2298         V$SERV  SMR-759 01/21/75        NEWDORF F
*A*          JAZ     MPER    PAGE ZERO ILLEGAL                       F
   .INSERT,2364                 V$SERV  MAPIN   05/22/75  KERNS             E.2
*A*          STA     MP25B+1     SAVE MAPIN PAGE                             E.2
   .REPLACE,2367                V$SERV  MAPIN   05/22/75  KERNS             E.2
*D*MP25A     ADDI    0                                                       V2
*A*MP25B     LDAI    0           MAPIN PAGE                                  E.2
*A*MP25A     SUBI    0           CRDR PAGE                                   F.2
```

```
.REPLACE,562                    CTSP0A   SMR-706  03/14/75 KERNS    F.2
*D*RECSIZ EQU     73
*A*RECSIZ EQU     74                                                F.2
```

```
   .REPLACE,21                    CTSP1A    SMR-708   03/14/75  KERNS      F.2
*D*RECSI7 EQU      73
*A*RECSI7 EQU      74                                                      F.2
```

```
   .REPLACE.21                    CTSP2A    SMR-706   03/14/75  KERNS     F.2
*D*RFCSI7 EQU      73
*A*RFCSI7 EQU      74                                                     F.2
```

```
   .REPLACE,21                    CTSP3A    SMR-706   03/14/75  KERNS     F.2
*D*RECSIZ EQU      73
*A*RECSIZ EQU      74                                                     F.2
```

```
   .REPLACE,21                    CTSP4A    SMR-706   03/14/75  KERNS     F.2
*D*RECSI7 EQU      73
*A*RECSI7 EQU      74                                                     F.2
```

```
   .REPLACE,21                    CTSP5A    SMR-706   03/14/75  KERNS     F.2
*0*RFCSI7 EQU      73
*A*RFCSI7 EQU      74                                                     F.2
```

```
   .REPLACE.21                    CTSP6A    SMR-706   03/14/75  KERNS     E.2
*D*RFCSI7 EQU      73
*A*RFCSI7 EQU      74                                                     E.2
```

```
   .REPLACE,21                    CTSP7A    SMR-706   03/14/75  KERNS     E.2
*D*RECSIZ EQU      73
*A*RECSIZ EQU      74                                                     E.2
```

```
    .REPLACE,623                    VSTYA     SMR=     10/23/74  KERNS    E.1
*D*         FRA     RR15            SET UP CHARACTER COUNT                D.1
*A*         TAP     TY04A                                                 E.1
*A*         LDA     THREE                                                 E.1
*A*TY04A    EQU     *                                                     E.1
*A*         FRA     RR15                                                  E.1
    .REPLACE,785                    VSTYA     SMR=     10/23/74  KERNS    E.1
*D*         LDB     0,B                                                   D.1
*A*         TFF     VORTEX=2                                              E.1
*A*         OMF     MPDVAD,VSST3    USER MAP                              E.1
*A*         LDB     0,B                                                   E.1
*A*         TFF     VORTEX=2                                              E.1
*A*         OMF     MPDVAD,VSST0    EXEC MAP                              E.1
    .INSERT,857                     VSTYA     SMR=     07/03/75  KERNS    F.2
*A*         LDB     4,X             RQBLK ADDR                            F.2
*A*         LDA     2,B             FCB ADDR                              F.2
*A*         STA     10,X            CTFCB                                 F.2
    .INSERT,1003                    VSTYA     SMR=903  06/16/75  NEWDORF  F.2
*A*         LDA     CTMODE,X        MODE> 4#                              F.2
*A*         TANZ    TYT16B          IF NOT, JUMP AROUND                   F.2
*A*         LDBI    TYIEXT          LOAD JUMP ADDRESS                     F.2
*A*         JMP     TYT1B                                                 F.2
*A*TYI16B   EQU     *                                                     F.2
    .REPLACE,1011,1014              VSTYA     SMR=903  06/16/75  NEWDORF  F.2
*D*         LDA     CTMODE,X        CHECK MODE                            D.1
*D*         TANZ    TYT16A          MODE NEQ 4                            D.1
*D*         LDBI    TYIEXT          MODE = 4, DO NOT OUTPUT CR/LF         D.1
*D*         JMP     TYT1B                                                 D.1
```

```
   .REPLACE,834,837                VZDD      SMR-788   03/12/75  SAKAMOTO  E.2
*D*        JANZ     D1RB30         NOW CHECK STATUS
*D*        TAP
*D*        STA      CTDTPK,X       SEEK ERROR, RECALIBRATE TO HOME
*D*        JSR      DSEEK,B
*A*        JANZ     D1RB30                                                  E.2
*A*        JSR      DRECAL,B       RECALIBRATE                              E.2
   .REPLACE,851                    VZDD      SMR-788   03/12/75  SAKAMOTO  E.2
*D*        JAZ      D1BGOP         NO. NOW BEGIN I/O OPERATION
*A*        JANZ     D1RB31         JUMP ON BAD STATUS FROM SEEK             E.2
*A*        JSR      DRCA,B         READ CURRENT ADDRESS                     E.2
*A*        LDA      CTCYLN,X       CALCULATED CYLINDER NUMBER               E.2
*A*        SUB      CTOCBF,X       SUBTRACT ACTUAL CYLINDER NUMBER          E.2
*A*        JAZ      D1BGOP         0, ON CYLINDER                           E.2
*A*D1RB31  JSR      DRECAL,B       RECALIBRATE                              E.2
*A*        TXA                     SAVE X                                   E.2
*A*        DELAY    100,0,2        DELAY 500 MS FOR RECALIBRATE             E.2
*A*        TAX                                                              E.2
   .INSERT,1809                    VZDD      SMR-788   03/12/75  SAKAMOTO  E.2
*A*        EJEC                                                             E.2
*A**       SUBROUTINE:             DRCA                                     E.2
*A**                                                                        E.2
*A**       PURPOSE:                READ CURRENT ADDRESS                     E.2
*A**                                                                        E.2
*A**       DESCRIPTION:            READS THE CURRENT ADDRESS FROM THE 5     E.2
*A**                               WORD COUNT FIELD IN THE RMD RECORD       E.2
*A**                               UNDER BIC CONTROL.  THE CURRENT          E.2
*A**                               ADDRESS IS READ INTO THE BUFFER          E.2
*A**                               STARTING AT CTOCBF.                      E.2
*A**                                                                        E.2
*A**       CALLING SEQUENCE:       JSR    DRCA,B                            E.2
*A**                                                                        E.2
*A**       ON ENTRY:               X = CTBL ADDRESS                         E.2
*A**                               B = RETURN ADDRESS                       E.2
*A**                                                                        E.2
*A**       ON EXIT:                X = CTBL ADDRESS                         E.2
*A**                                                                        E.2
*A*DRCA    EQU      *                                                       E.2
*A*        STB      CTRTN,X        SAVE RETURN ADDRESS                      E.2
*A*        LDBI     6000           APPROX. 50 MS DELAY                      E.2
*A*DRCA01  EQU      *                                                       E.2
*A*        LDA      DSEN4          OP-CODE FOR SEN  0400                    E.2
*A*        ORA      CTDVAD,X                                                E.2
```

```
*A*        EXC     DISPIM        DISABLE PIMS                                  E,2
*A*        EXC     DISCLK        DISABLE CLOCK                                 E,2
*A*        STA     *+1                                                         E,2
*A*        SEN     0400,DRCA02   SENSE DCU NOT BUSY                            E,2
*A*        EXC     ENAPIM        ENABLE PIMS                                   E,2
*A*        EXC     ENACLK        ENABLE CLOCK                                  E,2
*A*        DBR                                                                 E,2
*A*        JBNZ    DRCA01        WAIT                                          E,2
*A*        JMP     D1SKER        REPORT ERROR, DCU BUSY TOO LONG               E,2
*A*DRCA02  EQU     *                                                           E,2
*A*        LDA     BS15          OP-CODE FOR EXC                               E,2
*A*        ORA     CTDVAD,X      :EXC    0DD: COMMAND                          E,2
*A*        STA     DRCA03                                                      E,2
*A*        ORA     BS7           :EXC    02DD: COMMAND                         E,2
*A*        STA     DRCA05        STORE :READ ADDRESS: COMMAND                  E,2
*A*        LDA     DOAR          :OAR    0: OP-CODE                            E,2
*A*        ORA     CTDVAD,X                                                    E,2
*A*        STA     DRCA04        STORE :OAR    0DD: COMMAND                    E,2
*A*DRCA03  EXC     0             STOP TRANSFER AND INITIALIZE                  E,2
*A*        LDA     CTUNIT,X                                                    E,2
*A*DRCA04  OAR     0             OUTPUT SUW                                    E,2
*A*        TXA                                                                 E,2
*A*        ADDI    CTOCBF        FORM FWA OF INPUT BUFFER                      E,2
*A*        TAX                                                                 E,2
*A*        TAR                   FORM LWA OF ADDRESS BUFFER                    E,2
*A*        JSR     VSBIC,B       COMMON BIC SETUP ROUTINE                      E,2
*A*        DATA    1                                                           E,2
*A*        IFF     VORTEX-2                                                    E,2
*A*        DATA    0             MAP KEY VALUE                                 E,2
*A*        JAN     D1BERR        ERROR IF BIC NOT READY                        E,2
*A*        LDB     VSCTL         CTBL TIDB ADDRESS                             E,2
*A*        LDA     TBST,B                                                      E,2
*A*        ANAI    0177667       CLEAR TIMEOUT AND INTERRUPT                   E,2
*A**                             EXPECTED BITS                                 E,2
*A*        STA     TBST,B                                                      E,2
*A*DRCA05  EXC     0200          READ ADDRESS                                  E,2
*A*        EXC     ENAPIM                                                      E,2
*A*        EXC     ENACLK                                                      E,2
*A*        JSR     DINST,B       INPUT STATUS                                  E,2
*A*        JANZ    D1RWE2        JUMP ON BAD STATUS                            E,2
*A*        LDA     CTOCBF+1,X    FETCH WORD CONTAINING CYLINDER ADDRESS        E,2
*A*        ANA     BM777         EXTRACT CYLINDER NUMBER                       E,2
*A*        STA     CTOCBF,X      SAVE CURRENT ACTUAL CYLINDER NUMBER           E,2
```

```
*A*        LDB     CTRTN,Y      RESTORE RETURN ADDRESS                          E.2
*A*        TJMP    0,B                                                          E.2
   .REPLACE,1824               VZDD      SMR-788   03/12/75  SAKAMOTO          E.2
*D*DSEEK   STB     CTRTN,Y      SAVE RETURN ADDR.
*A*DRECAL  ROF                  RESET RECALIBRATE INDICATOR                     E.2
*A*        DATA    01001        SOF OPCODE                                      E.2
*A*DSEEK   SOF                  SET SEEK INDICATOR                              E.2
*A*        STB     CTRTN,Y      SAVE RETURN ADDRESS                             E.2
   .INSERT,1825                VZDD      SMR-788   03/12/75  SAKAMOTO          E.2
*A*        TZB                                                                  E.2
*A*        JOFN    DSK01        JUMP FOR RECALIBRATE                            E.2
   .INSERT,1829                VZDD      SMR-788   03/12/75  SAKAMOTO          E.2
*A*DSK01   EQU     *                                                            E.2
   .INSERT,1848                VZDD      SMR-788   03/12/75  SAKAMOTO          E.2
*A*        EXC     ENAPIM                                                       E.2
*A*        EXC     ENACLK                                                       E.2
   .INSERT,1860                VZDD      SMR-788   03/12/75  SAKAMOTO          E.2
*A*        YOFN    SETBB        FORM RECALIBRATE COMMAND                        E.2
   .REPLACE,1870               VZDD      SMR-788   03/12/75  SAKAMOTO          E.2
*D*        EXC     0100         SELECT CYLINDER
*A*        EXC     0100         SELECT CYLINDER OR RECALIBRATE                  E.2
   .INSERT,1878                VZDD      SMR-788   03/12/75  SAKAMOTO          E.2
*A*SETB8   ORA     BS8                                                          E.2
```

```
    .REPLACE,431                    V7CRA     NO SMR    02/25/75  KERNS     F.2
*D*         TAN     NREADY
*A*         TAP     V7CR1           TF READY                                F.2
*A*         DELAY   1,0,0           WAIT 5MS IN CASE READER STILL IN MOTION F.2
*A*         LDX     VSCTL           TIOB LOCATION                           F.2
*A*         LDX     TBRSTS,1        GET CONTROLLER TABLE ADDR               F.2
*A*         LDA     SENRDY          SENSE READER READY                      F.2
*A*         ADD     CTRVAT,1        DEVICE ADDR                             F.2
*A*         ISR     SENCRD,2        SENSE READER READY                      F.2
*A*         TAN     NREADY          TROUBLE IF STILL NOT READY              F.2
*A*V7CR1    EQU     *
```

```
    .INSERT,618                     VSCLPS    NO SMR    11/18/74 KERNS   E,1
*A*         ANA     THREE           EXTRACT TWO BITS                     E,1
    .INSERT,626                     VSCLPS    NO SMR    11/18/74 KERNS   E,1
*A*         LDA     CTLWTD,X        GET LINE WIDTH CODE                  E,1
*A*         BT      BA0+B2,STS3A    JUMP IF SLTB SPECIFIED               E,1
*A*         LDA     CTATTR,X        IF NOT, RESET SLTB BIT               E,1
*A*         ANA     BR13            IN ATTRIBUTE WORD                    E,1
*A*         STA     CTATTR,X                                             E,1
    .REPLACE,878                    VSCLPS    SMR=      10/23/74 KERNS   E,1
*D*         STA     CTEMP4,X        SET STEP COUNT                       D,1
*A*         STA     CTFUNC,X        SET STEP COUNT                       E,1
    .REPLACE,880,882                VSCLPS    SMR=      10/23/74 KERNS   E,1
*D*         LDA     CTEMP4,X                                             D,1
*D*         DAR                     DECREMENT STEP COUNT                 D,1
*D*         STA     CTEMP4,X                                             D,1
*A*         LDA     CTFUNC,X                                             E,1
*A*         DAR                                                          E,1
*A*         STA     CTFUNC,X                                             E,1
    .REPLACE,931                    VSCLPS    SMR=      11/21/74 KERNS   E,1
*D*         SUBI    1099                                                 D,1
*A*         SUBI    1120            PASTERS PER PAGE (PLUS SLACK)        E,1
*A*         NAME    VSSTPL                                               E,1
*A*VSSTPL   RES     0                                                    E,1
    .REPLACE,1142,1145              VSCLPS    NO SMR    11/22/74 BRACKETT E,1
*D*         LDA     CTDVAT,X                                             D,1
*D*         ORA     SENSB           BUILD SENSE INSTR                    D,1
*D*         STA     *+1                                                  D,1
*D*         SEN     0,STS4C4        JUMP IF AT BOTTOM OF FORM            D,1
    .REPLACE,1156,1157              VSCLPS    NO SMR    11/22/74 BRACKETT E,1
*D*STS4C4   EQU     *                                                    D,1
*D*         LDA     N300                                                 D,1
    .REPLACE,1185                   VSCLPS    NO SMR    12/12/74 KERNS   E,1
*D*         JMP     STS4C3          GO END OPERATION                     D,1
*A*         JMP     STS4C8          GO END OPERATION                     E,1
    .REPLACE,1240,1241              VSCLPS    NO SMR              BRACKETT E,1
*D*         JSR     STEP,B      STEP STATUS
*D*         JSR     STRD,B      JUMP TO PROCESS WRITE
*A*         LDA     CTATTR,X                                             E,1
*A*         BT      BA0+B14,STPLT4  JUMP IF STATUS 33                    E,1
*A*         JSR     STEP,B      STEP STATUS                              E,1
*A*         JSR     STRD,B      PROCESS WRITE                            E,1
*A*         JMP     STPLT5                                               E,1
*A*STPLT4   EQU     *                                                    E,1
```

```
*A*        JSR     STBD,B          PROCESS WRITE                          E,1
*A*        JSR     STEP,B          STEP STATOS                            E,1
*A*STPLTS EQU      *                                                      E,1
   .REPLACE,1281                   VSCLPS    NO SMR              BRACKETT E,1
*D*        BT      RA0+B14,STS6A3  JUMP IF STATOS 33                      D,1
*A*        LDA     CTATTR,X                                               E,1
*A*        BT      RA0+B14,STS6A3  JUMP IF STATOS 33                      E,1
```

```
    .REPLACE,595                        VZLPDX    SMR=      10/30/74  KERNS    E.1
*D*         SUB       CTLSTZ,X        GREATER THAN LINE WIDTH?                  I
*A*         SUB       CTLWCH,X        GREATER THAN LINE WIDTH ?                 E.1
    .REPLACE,598                        VZLPDX    SMR=      10/30/74  KERNS    E.1
*D*         LDA       CTLSTZ,X                                                  L
*A*         LDA       CTLWCH,X                                                  E.1
```

```
   .REPLACE,627,628                VZCPA     V75        02/01/75  MCMURRAY  V75
*D*        LSRA     1
*D*        JAZ      CPDF
*A*        BT       050,CPDF                                                V75
```

```
    .REPLACE,506,508                VZSPA     SMR-708   03/14/75  KERNS     F.2
*D*         SUB       K103
*D*         JAN       D1WR1         BUFFER LENGTH OK
*D*         LDA       K103          SET BUFFER LUNGTH TO MAXIMUM
*A*         SUB       BFSZ                                                  F.2
*A*         JAN       D1WR1         BUFFER LENGTH OK                        F.2
*A*         LDA       BFSZ          SET BUFFER LENGTH TO MAX                F.2
    .REPLACE,604                    VZSPA     SMR-708   03/14/75  KERNS     F.2
*D*K103     DATA      0103
*A*BFSZ     DATA      0104                                                  F.2
```

```
    .INSERT,11                  VSDASMR   V75       02/01/75  MCMURRAY  V75
*A*V75      SET     1           ENABLE V75 INSTRUCTION SET              V75
    .INSERT,46                  VSDASMR   V75       02/01/75  MCMURRAY  V75
*A*BSO      EQU     MT+1                                                V75
*A*TWO      EQU     MT+2                                                V75
*A*BS3      EQU     MT+4                                                V75
*A*BS4      EQU     MT+5                                                E.2
*A*BS6      EQU     MT+7                                                V70
*A*FIVE     EQU     MT+37                                               V75
*A*KD15     EQU     MT+42                                               V75
    .INSERT,106                 VSDASMR   V75       02/01/75  MCMURRAY  V75
*A*         IFT     V75                                                 V75
*A*V75ED    BSS     1           V75 LIST EDIT SWITCH                    V75
    .INSERT,216                 VSDASMR   V75       02/01/75  MCMURRAY  V75
*A*P1INIT   EQU     *                                                   V75
    .INSERT,317                 VSDASMR   V75       02/01/75  MCMURRAY  V75
*A*         IFT     V75                                                 V75
*A*         STAE    V75ED       DISABLE V75 LIST EDIT                   V75
    .REPLACE,741                VSDASMR   SMR-613   10/18/74  NEWDORF  E.1
*D*         JAZM    TOF         TOP OF FORM ON LO
*A*         JANZ    CONT        DO NOT EJECT LISTING                    E,1
*A*         LDAE*   FRM$        IF WE JUST DID TOF,                     E,1
*A*         SUBI    2           DO NOT DO REDUNDANT TOF.                E,1
*A*         SUBE    LCT                                                 E,1
*A*         JANZM   TOF         OTHERWISE, DO TOF.                      E.1
    .INSERT,1583                VSDASMR   V75       02/01/75  MCMURRAY  V75
*A*         IFF     V75                                                 V75
*A*         GOTO    1                                                   V75
*A*         DECR    1                                                   V75
*A*         STA     MACCHC      CLEAR CHARACTER COUNTER                 V75
*A*         STA     MACCMC      AND COMMA COUNTER                       V75
*A*1        CONT                                                        V75
    .INSERT,1590                VSDASMR   V75       02/01/75  MCMURRAY  V75
*A*         IFT     V75                                                 V75
*A*         JAZ     MACC2       JUMP IF COMMA                           V75
*A*         IFF     V75                                                 V75
    .INSERT,1591                VSDASMR   V75       02/01/75  MCMURRAY  V75
*A*         IFT     V75                                                 V75
*A*MACC1    INR     MACCHC      BUMP CHARACTER COUNT                    V75
    .INSERT,1593                VSDASMR   V75       02/01/75  MCMURRAY  V75
*A*         IFF     V75                                                 V75
*A*         GOTO    1                                                   V75
*A*MACC2    LDA     $PCW                                                V75
```

```
*A*         RT      043,MACD    'E' PARAMETER SET ?                 V75
*A*         LDA     MACCMC                                          V75
*A*         JAP     MACC5       1ST COMMA ?                         V75
*A*         CALL    OCFO        YES, LOOK UP OP                     V75
*A*         DATA    LAB2                                            V75
*A*         JXZ     MACD        FIND ?                              V75
*A*         LDA     3,X         YES                                 V75
*A*         LSRA    10                                              V75
*A*         SUB     EIGHT                                           V75
*A*         JANZ    MACD        V75 OP ?                            V75
*A*         LDA     3,X         YES                                 V75
*A*         ANA     RHW                                             V75
*A*         TAR                                                     V75
*A*         SUB     FIVE                                            V75
*A*         JAZ     MACC3       2-COMMA OP ?                        V75
*A*         LDAI    0325        NO                                  V75
*A*         ANA     BSO,B                                           V75
*A*         JAZ     MACD        1-COMMA OP ?                        V75
*A*         JMP     MACC4       YES                                 V75
*A*MACC3    TNRE    V75ED       SET EOTT FLAGS                      V75
*A*MACC4    TNRE    V75ED                                           V75
*A*         TNR     MACCMC      BUMP COMMA COUNT                    V75
*A*         LDA     MACCHC                                          V75
*A*         JAP     MACC6       1ST CHAR ?                          V75
*A*         LDA     SBTA        YES                                 V75
*A*         DAR                 BACK UP CHARACTER POINTER           V75
*A*         STA     SBTA                                            V75
*A*         JMP     MACC6                                           V75
*A*MACC5    LDAE    V75ED                                           V75
*A*         DAR                                                     V75
*A*         JAZ     MACD        2-COMMA OP ?                        V75
*A*         TNR     MACCMC      YES, BUMP COMMA COUNT               V75
*A*         LDA     MACCMC                                          V75
*A*         SUB     TWO                                             V75
*A*         JAP     MACD        EXIT IF MORE THAN 2 COMMAS          V75
*A*MACC6    LDR     =',         '                                   V75
*A*         JMP     MACC1       STORE COMMA                         V75
*A*MACCHC   DATA    0           CHARACTER COUNTER                   V75
*A*MACCMC   DATA    0           COMMA COUNTER                       V75
*A*1        CONT                                                    V75
   .REPLACE,1769                V$DASMR  INDIR  03/04/75  KADISON   E.2
*D*         JMP     T1D                                             
*A*         JMPM    ILV                                             E.2
```

```
*A*         LDA     SPCW                                                        E.2
*A*         ANA     BS4                                                         E.2
*A*         JAZ     RCK             EXIT IF NO I OPTION                         E.2
*A*         LDR     QITE                                                        E.2
*A*         JMPM    FRR             FLAG ERROR                                  E.2
*A*         JMP     RCK                                                         E.2
   .INSERT,1865                     V5DASMR   INDIR    03/04/75  KADISON       E.2
*A*         JMPM    LLV                                                         E.2
*A*         LDA     SPCW                                                        E.2
*A*         ANA     BS4                                                         E.2
*A*         JAZ     RCK             EXIT IF NO I OPTION                         E.2
*A*         LDR     QITE                                                        E.2
*A*         JMPM    FRR                                                         E.2
*A*         JMP     RCK                                                         E.2
   .INSERT,1867                     V5DASMR   INDIR    03/04/75  KADISON       E.2
*A*QTIE     DATA    'IT'                                                        E.2
   .REPLACE,1901                    V5DASMR   V75      02/01/75  MCMURRAY      V75
*D*         LDA     TWA                                                         
*A*T41      LDA     TWA                                                         V75
   .INSERT,1932                     V5DASMR   V75      02/01/75  MCMURRAY      V75
*A*         TFF     V75                                                         V75
*A*         GOTO    1                                                           V75
*A**                                                                            V75
*A** PROCESS V75 INSTRUCTION SET                                                V75
*A**                                                                            V75
*A*V75INS   LDA     SPCW                                                        V75
*A*         RT      C43,ASR5        WAS THERE AN 'E' PARAMETER ON '/DASMR' ?V75
*A*         LDA     TADF            YES                                         V75
*A*         STA     V75TMP          SAVE INDIRECT ADDRESS FLAG                  V75
*A*         TZA                                                                 V75
*A*         STA     TADF            CLEAR INDIRECT ADDRESS FLAG                 V75
*A*         LDB     AS                                                          V75
*A*         LDRE    V75JT,B         INDEX INTO JUMP TABLE                       V75
*A*         TJMP    0,B                                                         V75
*A*V75JT    PZE     V751            TYPE 0                                      V75
*A*         PZE     V757            TYPE 1                                      V75
*A*         PZE     V752            TYPE 2                                      V75
*A*         PZE     V752A           TYPE 3                                      V75
*A*         PZE     V752A           TYPE 4                                      V75
*A*         PZE     V751            TYPE 5                                      V75
*A*         PZE     V752            TYPE 6                                      V75
*A*         PZE     V752            TYPE 7                                      V75
*A**                                                                            V75
```

```
*A** V75 INSTRUCTIONS WITH REGISTER OP IN BITS 3-5(TYPE 0,5)                V75
*A**                                                                       V75
*A*V751   CALL    GV75R           GET REGISTER NUMBER                      V75
*A*       LRLA    3               POSITION                                 V75
*A*       JMP     V753                                                     V75
*A**                                                                       V75
*A** V75 INSTRUCTIONS WITH REGISTER OP IN BITS 0-2(TYPE 2,6,7)              V75
*A**                                                                       V75
*A*V752   CALL    GV75R           GET REGISTER NUMBER                      V75
*A*       JMP     V753            AND EXIT                                 V75
*A**                                                                       V75
*A** V75 INSTRUCTIONS WHERE REG MUST BE 0 OR 4(TYPE 3,4)                    V75
*A**                                                                       V75
*A*V752A  CALL    GV75R           GET REGISTER NUMBER                      V75
*A*       JAZ     V753            ZERO OK                                  V75
*A*       SUB     FOUR                                                     V75
*A*       JAZ     V752B           FOUR OK                                  V75
*A*       LDA     ='52'                                                    V75
*A*       CALL    ERR             OUTPUT ERROR MESSAGE                     V75
*A*V752B  LDA     AS              GET INSTRUCTION TYPE                     V75
*A*       SUB     THREE                                                    V75
*A*       TAB                                                              V75
*A*       LDA     FOUR                                                     V75
*A*       JBZ     V753            TYPE 3 ?                                 V75
*A*       LRLA    4               NO. TYPE 4                               V75
*A*V753   ORA     TWA             MERGE IN INSTRUCTION                     V75
*A*       STA     TWA                                                      V75
*A*       LDA     AS              GET TYPE                                 V75
*A*       SUB     FIVE                                                     V75
*A*       TNRE    V75ED           BUMP EDIT FLAG                           V75
*A*       JANZ    V756            TYPE 5 ?                                 V75
*A*       TNRE    V75ED           YES. BUMP EDIT FLAG                      V75
*A*       CALL    CTST                                                     V75
*A*       JAZ     V754            COMMA ?                                  V75
*A*       TZA                     NO                                       V75
*A*       JMP     V755                                                     V75
*A*V754   CALL    GV75R           GET DESTINATION REGISTER                 V75
*A*V755   ORA     TWA             MERGE IN INSTRUCTION                     V75
*A*       JMP     T21             TYPE 5 EXIT                              V75
*A*V756   DAR                                                              V75
*A*       JAZ     T41             TYPE 6 EXIT                              V75
*A*       LDX     EIGHT                                                    V75
*A*       CALL    SBK             SKIP BLANKS                              V75
```

```
*A*         LDA     AS                                                  V75
*A*         SUB     SEVEN                                               V75
*A*         JAP     T2              TYPE 7 EXIT                         V75
*A*         CALL    CTST            COMMA OK                            V75
*A**                                                                    V75
*A** V75 INSTRUCTIONS WITH NO REGISTER FIELD (TYPE 1)                   V75
*A**                                                                    V75
*A*V757     CALL    FXP             GET OPERAND                         V75
*A*         STA     T5T             SAVE VALUE                          V75
*A*         STB     T5T+1           AND ATTRIBUTES                      V75
*A*         LDA     AS                                                  V75
*A*         SUB     TWO                                                 V75
*A*         JAZ     V759            TYPE 2 EXIT                         V75
*A*         DAR                                                         V75
*A*         JAZ     V759            TYPE 3 EXIT                         V75
*A*         CALL    CTST                                                V75
*A*         JANZ    V759            COMMA ?                             V75
*A*         CALL    GV75R           YES. GET INDEX REGISTER             V75
*A*         ORA     IWA             MERGE IN INSTRUCTION                V75
*A*         STA     IWA                                                 V75
*A*V759     LDA     V75TMP                                              V75
*A*         ORA     T5T+1           RESTORE INDIRECT ADDRESS FLAG       V75
*A*         STA     T5T+1                                               V75
*A*         JMP     T22                                                 V75
*A*V75TMP   DATA    0               TEMP STORE                          V75
*A*1        CONT                                                        V75
   .REPLACE,2747,2749               V8DASMR   NO SMR    02/27/75 KERNS  E.2
*D*         LDA     LORMD           IS LO AN RMD                        C.1
*D*         JAZ     STEND2          NO                                  C.1
*D*         CLOSE   LOFCB,LO,0,1    CLOSE AND UPDATE LO
   .REPLACE,3318                    V8DASMR   V75       02/01/75 MCMURRAY V75
*D*         LDA*    FRMS
*A*         LDAE*   FRMS                                                V75
   .INSERT,3520                     V8DASMR   V75       02/01/75 MCMURRAY V75
*A*         TFT     V75                                                 V75
*A*         JAZ     LLNV75          JUMP IF COMMA                       V75
*A*         TFF     V75                                                 V75
   .REPLACE,3522                    V8DASMR   V75       02/01/75 MCMURRAY V75
*D*         TBA
*A*LLNC1    TBA                                                         V75
   .INSERT,3523                     V8DASMR   V75       02/01/75 MCMURRAY V75
*A*         TFF     V75                                                 V75
*A*         GOTO    1                                                   V75
```

```
*A*LLNV75 LDAE    V75ED                                                  V75
*A*       JAZ     LLND        ANY V75 EDITING ?                          V75
*A*       DAR                 YES. DROP COMMA COUNT                      V75
*A*       STAE    V75ED                                                  V75
*A*       JMP     LLNC1                                                  V75
*A*1      CONT                                                           V75
   .INSERT,3532               VSDASMR  V75      02/01/75  MCMURRAY       V75
*A*       TFF     V75                                                    V75
*A*       GOTO    1                                                      V75
*A*       LDA     SBTA                                                   V75
*A*       SUB     LB2P36                                                 V75
*A*       JAP     LLNM+1      DONT RESET BYTE POINTER IF OVERRUN         V75
*A*1      CONT                                                           V75
   .INSERT,3538               VSDASMR  V75      02/01/75  MCMURRAY       V75
*A*       TFF     V75                                                    V75
*A*       GOTO    1                                                      V75
*A*       LDA     SBTA                                                   V75
*A*       SUB     LB2P51                                                 V75
*A*       SUBI    43                                                     V75
*A*       JAP     LLNJ        EXIT AT END OF COMMENT FIELD               V75
*A*       TBA                 RESTORE A-REG                              V75
*A*1      CONT                                                           V75
   .INSERT,3955               VSDASMR  V75      02/01/75  MCMURRAY       V75
*A**                                                                     V75
*A** GET V75 REGISTER NUMBER                                             V75
*A**                                                                     V75
*A*       TFF     V75                                                    V75
*A*       GOTO    1                                                      V75
*A*GV75R  ENTR                                                          V75
*A*       CALL    TAC         GET ABS REGISTER NUMBER                    V75
*A*       LDB     ='SZ'                                                  V75
*A*       LSRA    3                                                      V75
*A*       JANZM   ERR         ERROR IF REG NOT 0-7                       V75
*A*       LDA     VAL         RESTORE A                                  V75
*A*       JAZ*    GV75R       A=0 OK                                     V75
*A*       TAB                                                            V75
*A*       SUB     THREE                                                  V75
*A*       ROF                                                            V75
*A*       XAN     SOF         SET OVFL ON 1,2                            V75
*A*       TBA                                                            V75
*A*       XUF     GV75RE      FLIP 1,2                                   V75
*A*       JMP*    GV75R       EXIT                                       V75
*A*GV75RF FRA     THREE                                                  V75
```

```
*A*SOF     SOF                                                          V75
*A*1       CONT                                                         V75
   .INSERT,4335                V$DASMR   V75      02/01/75  MCMURRAY   V75
*A*        IFT     V75                                                  V75
*A*        PZE     V75INS      V75 INSTRUCTION SET                      V75
   .INSERT,4708                V$DASMR   V75      02/01/75  MCMURRAY   V75
*A**                                                                    V75
*A** V75 INSTRUCTIONS                                                   V75
*A**                                                                    V75
*A*        IFF     V75                                                  V75
*A*        GOTO    1                                                    V75
*A**                                                                    V75
*A** TYPE 0                                                             V75
*A**                                                                    V75
*A*        DATA    'LD     ',020000,07000                               V75
*A*        DATA    'ST     ',020000,07100                               V75
*A*        DATA    'AD     ',020000,07200                               V75
*A*        DATA    'SB     ',020000,07300                               V75
*A**                                                                    V75
*A** TYPE 1                                                             V75
*A**                                                                    V75
*A*        DATA    'LRT    ',020001,07460                               V75
*A*        DATA    'SRT    ',020001,07470                               V75
*A**                                                                    V75
*A** TYPE 2                                                             V75
*A**                                                                    V75
*A*        DATA    'J7     ',020002,06720                               V75
*A*        DATA    'JNZ    ',020002,06730                               V75
*A*        DATA    'JN     ',020002,06740                               V75
*A*        DATA    'JP     ',020002,06750                               V75
*A**                                                                    V75
*A** TYPE 3                                                             V75
*A**                                                                    V75
*A*        DATA    'JDZ    ',020003,06760                               V75
*A*        DATA    'JDN7   ',020003,06770                               V75
*A**                                                                    V75
*A** TYPE 4                                                             V75
*A**                                                                    V75
*A*        DATA    'DLD    ',020004,04600                               V75
*A*        DATA    'DST    ',020004,04610                               V75
*A*        DATA    'DADD   ',020004,04620                               V75
*A*        DATA    'DSUB   ',020004,04630                               V75
*A*        DATA    'DAN    ',020004,04640                               V75
```

```
*A*       DATA    'DDR    ',020004,04650                          V75
*A*       DATA    'DFR    ',020004,04660                          V75
*A**                                                              V75
*A** TYPE 5                                                       V75
*A**                                                              V75
*A*       DATA    'ADR    ',020005,07500                          V75
*A*       DATA    'SRR    ',020005,07600                          V75
*A*       DATA    'T      ',020005,07700                          V75
*A**                                                              V75
*A** TYPE 6                                                       V75
*A**                                                              V75
*A*       DATA    'INC    ',020006,07410                          V75
*A*       DATA    'DEC    ',020006,07420                          V75
*A*       DATA    'COM    ',020006,07430                          V75
*A**                                                              V75
*A** TYPE 7                                                       V75
*A**                                                              V75
*A*       DATA    'LDI    ',020007,07440                          V75
*A*       DATA    'ADI    ',020007,07450                          V75
*A*1      CONT                                                    V75
    .REPLACE,4935,4968          VSOASMR  V75      02/01/75 MCMURRAY V75
*D**
*D**                         END OF VORTEX MACRO MOVE AREA
*D**
*D**              END INSTRUCTION TABLE
*D**
*D*DASMR  ORG     *           ASSEMBLER ENTERED HERE
*D*SO     EQU     *
*D**
*D**      THIS IS SOME PRE-PASS ONE PROCESSING THAT HAS TO BE
*D**      EXECUTED ONLY ONCE
*D**
*D**
*D*       LDAI    LBUF
*D*       ADDI    LBUF+8
*D*       STA     LB2P8
*D*       LDAI    LBUF
*D*       ADDI    LBUF+1
*D*       STA     LB2P1
*D*       LDAI    LBUF
*D*       ADDI    LBUF+51
*D*       STA     LB2P51      (=LBUF*2+51)
*D*       LDAI    LBUF
```

```
*D*         ADDI    LBUF+28
*D*         STA     LB2P28
*D*         LDAI    LBUF
*D*         ADDI    LBUF+22
*D*         STA     LB2P22
*D*         LDAI    LBUF
*D*         ADDI    LBUF+36
*D*         STA     LB2P36          (=LBUF*2+36)
*D**
*D**                                CHECK ASSEMBLER OPTIONS IN V$JFCB
*D**                                    AND SET BITS IN PROG CONT WORD (SPCW.
*D**
*A*         FJFC                                                              V75
*A*****************************************************************************V75
*A**                                                                           V75
*A**          I N I T I A L I Z E   A S S E M B L E R ( I N I T )              V75
*A**                                                                           V75
*A** FUNCTION: TO INITIALIZE VORTEX/VORTEX II DASMR ASSEMBLER                  V75
*A**                                                                           V75
*A** ENTRY: DIRECT TO LABEL DASMR FROM VORTEX DISPATCHER                       V75
*A**                                                                           V75
*A** EXIT : DIRECT TO PIINIT                                                  V75
*A**        INIT IS OVERLAYED BY DASMR TABLES                                  V75
*A**                                                                           V75
*A*****************************************************************************V75
*A*         SPAC                                                              V75
*A*********************************************                               V75
*A** CONSTRUCT BYTE POINTERS FOR LIST BUFFER *                                 V75
*A*********************************************                               V75
*A*         SPAC                                                              V75
*A*DASMR    LDAI    LBUF                                                      V75
*A*         LRLA    1               GET BYTE POINTER TO LBUF                  V75
*A*         TAR                                                               V75
*A*         STA     LB2P1           +1  ADDRESS FIELD                         V75
*A*         ADD     SEVEN                                                     V75
*A*         STA     LB2P8           +8  VALUE FIELD                           V75
*A*         ADDI    14                                                        V75
*A*         STA     LB2P22          +22 LABEL FIELD                           V75
*A*         ADD     SIX                                                       V75
*A*         STA     LB2P28          +28 OP CODE FIELD-1                       V75
*A*         ADD     EIGHT                                                     V75
*A*         STA     LB2P36          +36 OPERAND FIELD-1                       V75
*A*         ADD     KD15                                                      V75
```

```
*A*        STA     LB2P51        +51 COMMENT FIELD                          V75
*A*        SPAC                                                             V75
*A*************************************************                          V75
*A** MAP '/DASMR' PARAMETER CHARS INTO $PCW BITS *
*A*************************************************                          V75
*A*        SPAC                                                             V75
   .REPLACE,4984                V$DASMR   V75       02/01/75  MCMURRAY     V75
*D*        SUBI    012
*A*        SUB     THREE                                                    V75
*A*        JAZ     FOPT          E OPTION(V75 EXTENDED INSTRUCTIONS)        V75
*A*        SUB     FOUR                                                     E.2
*A*        JAZ     IOPT          I OPTION [DISALLOW INDIRECTS TO BASE PAGE]
*A*        SUB     THREE                                                    V75
   .INSERT,5003                 V$DASMR   V75       02/01/75  MCMURRAY     V75
*A*EOPT    LDA     BS3                                                      V75
*A*        ORA     $PCW          SET E FLAG                                 V75
*A*        STA     $PCW                                                     V75
*A*        JMP     VDASM2                                                   V75
*A*IOPT    LDA     BS4           I OPTION BIT                               E.2
*A*        ORA     $PCW                                                     E.2
*A*        STA     $PCW                                                     E.2
*A*        JMP     VDASM2                                                   E.2
   .REPLACE,5037                V$DASMR   V75       02/01/75  MCMURRAY     V75
*D*        STB     SILUN         STORE ACTUAL SI LUN
*A*        STB     SILUN         SAVE SI DST INDEX                          V75
   .REPLACE,5040,5042           V$DASMR   V75       02/01/75  MCMURRAY     V75
*D*        STB     PILUN         SAVE ACTUAL LUN
*D*        STA     PIRMD         PI RMD FLAG= 0, IF PI IS RMD: OTHERWISE =0
*D*        TBA                   GET PI LUN
*A*        STB     PILUN         SAVE PI DST INDEX                          V75
*A*        STA     PIRMD         PI RMD FLAG=1 IF PI=RMD,OTHERWISE ZERO     V75
*A*        TBA                   GET A=PI DST INDEX                         V75
   .REPLACE,5048,5050           V$DASMR   V75       02/01/75  MCMURRAY     V75
*D*        JIF     022,DASO      JMP IF PI=RMD AND PI=SI
*D*        JAP     DAS$00        JMP IF PI=RMD AND PI .NE. SI
*D*        TZA
*A*        JIF     022,DASO      JUMP IF PI=RMD=SI                          V75
*A*        JAP     DAS$00        JUMP IF PI=RMD#SI                          V75
*A*        TZA                   PI NOT RMD                                 V75
   .REPLACE,5053,5056           V$DASMR   V75       02/01/75  MCMURRAY     V75
*D*DASO    EQU     *
*D*        LDA     5,X
*D*        JANZ    DAS$00
```

```
*D*         STB     PIRMD           TURN OFF PI=RMD FLAG
*A*DAS0     LDA     5,X             PI=RMD=ST                                   V75
*A*         JANZ    DAS$00          IS PI GLOBAL FCB ACTIVE ?                   V75
*A*         STB     PIRMD           NO. THIS IS RMD JOB STACK                   V75
  .REPLACE,5070,5071                V$DASMR   V75     02/01/75  MCMURRAY    V75
*D*         STB     POLUN           STORE ACTUAL LUN
*D*         STA     PORMD
*A*         STB     POLUN           STORE PO DST INDEX                          V75
*A*         STA     PORMD           A=0 IF PO NOT RMD                           V75
  .REPLACE,5074,5075                V$DASMR   V75     02/01/75  MCMURRAY    V75
*D*         XAZ     DECB
*D*         STB     PORF            SET= 0, IF PO=RMD/PI NOT = RMD,OTHERWISE=-
*A*         XAZ     DECB            SET PORF=-1 IF PO NOT RMD                   V75
*A*         STB     PORF            OTHERWISE, LEAVE ALONE                      V75
  .REPLACE,5089,5090                V$DASMR   V75     02/01/75  MCMURRAY    V75
*D*         STB     SSLUN           STORE ACTUAL LUN
*D*         STA     SSRMD
*A*         STB     SSLUN           SAVE SS DST INDEX                           V75
*A*         STA     SSRMD           A=1 IF SS=RMD,ZERO OTHERWISE                V75
  .REPLACE,5095                     V$DASMR   V75     02/01/75  MCMURRAY    V75
*D*         ERA     POLUN           AND PO LUN FOR EQUALITY
*A*         ERA     POLUN           AND PO FOR SAME DEVICE OR RMD PARTITION     V75
  .REPLACE,5107,5108                V$DASMR   V75     02/01/75  MCMURRAY    V75
*D*         LDA     SILUN           COMPARE SI LUN
*D*         ERA     SSLUN           WITH SS LUN
*A*         LDA     SILUN           COMPARE SI DEVICE/PARTITION                 V75
*A*         ERA     SSLUN           WITH SS                                     V75
  .REPLACE,5112,5113                V$DASMR   V75     02/01/75  MCMURRAY    V75
*D*         LDA     PILUN           COMPARE PI
*D*         ERA     SSLUN           WITH SS LUN
*A*         LDA     PILUN           COMPARE PI DEVICE/PARTITION                 V75
*A*         ERA     SSLUN           WITH SS                                     V75
```

```
   .INSERT,1                        V$DEBUG   V75                    METR      V75
*A*V75      SET      1                                                          V75
   .REPLACE,79                      V$DEBUG   V75                    METR      V75
*D**        BIT 14  -  UNUSED
*A**       BIT 14  V75 FLAG  1=V75 0=NOT A V75 MACHINE                          V75
*A*TB75    EQU      2               WORD 2                                      V75
*A*TB75B   EQU      14              BIT 14                                      V75
*A*TB75Z   EQU      1               1 BIT LONG                                  V75
   .INSERT,326                      V$DEBUG   V75                    METR      V75
*A*         IFT      V75-1                                                      V75
*A*         GOTO     1                                                          V75
*A**            A VTAM TYPE MACRO TO TEST BIT FLAG                              V75
*A*TESTE   MAC                                                                  V75
*A*        LDA      P(2),P(1)       PICK UP WORD CONTAINING FLA                 V75
*A*        ANA      BSO+P(3)                                                    V75
*A*        EMAC                                                                 V75
*A**                SAVE V75 REGISTERS                                          V75
*A*SAVE75  MAC                                                                  V75
*A*        ST,3     P(1)                                                        V75
*A*        ST,4     P(2)                                                        V75
*A*        ST,5     P(3)                                                        V75
*A*        ST,6     P(4)                                                        V75
*A*        ST,7     P(5)                                                        V75
*A*        EMAC                                                                 V75
*A*LOAD75  MAC                                                                  V75
*A*        LD,3     P(1)                                                        V75
*A*        LD,4     P(2)                                                        V75
*A*        LD,5     P(3)                                                        V75
*A*        LD,6     P(4)                                                        V75
*A*        LD,7     P(5)                                                        V75
*A*        EMAC                                                                 V75
*A*1        CONT                                                                V75
   .REPLACE,344,346                 V$DEBUG   V75                    METR      V75
*D*        STA      AP              SAVE A,B,X, AND OVERFLOW                    C,1
*D*        STB      BP                                                          C,1
*D*        STX      XP                                                          C.1
*A*        STAE     AP              SAVE A,B,X AND OVERFLOW                      V75
*A*        STBE     BP                                                           V75
*A*        STXE     XP                                                           V75
   .REPLACE,350                     V$DEBUG   V75                    METR      V75
*D*        STA      OP                                                          C.1
*A*        STAE     OP                                                           V75
   .INSERT,371                      V$DEBUG   V75                    METR      V75
```

```
*A*          IFT     V75-1                                                    V75
*A*          GOTO    2                                                        V75
*A**         TEST IF A V75 MACHINE                                            V75
*A*          LDA     DMSBUF+TB75                                              V75
*A*          ANA     BS14                                                     V75
*A*          STAE    V75FLG          STORE IN DEBUG PROPER                    V75
*A*2          CONT                                                            V75
   .INSERT,382                       VSDEBUG   V75              METR          V75
*A*          IFT     V75-1                                                    V75
*A*          GOTO    3                                                        V75
*A*          TESTF   B,TB75,TB75B,TB75Z                                       V75
*A*          STAE    V75FLG                                                   V75
*A*3          CONT                                                            V75
   .INSERT,385                       VSDEBUG   V75              METR          V75
*A*          IFT     V75-1                                                    V75
*A*          GOTO    4                                                        V75
*A**         SAVE THE ADDITIONAL REGISTERS IF V75                             V75
*A*          LDAE    V75FLG                                                   V75
*A*          JA7     FEF             JMP IF NOT V75                           V75
*A*          SAVE75  R3P,R4P,R5P,R6P,R7P                                      V75
*A*FEF       EQU     *                                                        V75
*A*4          CONT                                                            V75
   .REPLACE,397                      VSDEBUG   V75              METR          V75
*D*          JAN     SOF-2           NO,
*A*          JAN     ICALL           NO.                                      V75
   .REPLACE,411                      VSDEBUG   V75              METR          V75
*D*          JA7     SOF-2           YES
*A*          JA7     ICALL           YES                                      V75
   .REPLACE,415                      VSDEBUG   V75              METR          V75
*D*          CALL    DINPT           READ INPUT
*A*ICALL     CALL    DINPT           READ INPUT                               V75
*A*           IFT     V75-1                                                   V75
*A*           GOTO    5                                                       V75
*A**                                                                         V75
*A**         THE V75 REGISTERS ARE DESIGNATED R0 THROUGH R7                   V75
*A**         WHERE : R0 IS THE SAME AS A REGISTER                             V75
*A**                 R1 IS THE SAME AS B REGISTER                             V75
*A**                 R2 IS THE SAME AS X REGISTER                             V75
*A**                 R3 THRU R7 ARE THE NEW REGISTERS                         V75
*A**         EACH OF THE RX 2 LETTER REGISTERS IS REPLACED BY                 V75
*A**         A ONE LETTER CODE TO CONFORM WITH THE GENERAL DEA                V75
*A**         PHILOSOPHY.                                                      V75
*A**                 R3 THRU R7 BECOMES J,K,L,M,N RESPECTIVELY                V75
```

```
*A**                IS DONE VIA THE RTABLE TABLE                          V75
*A**                                                                      V75
*A*       LDA       V75FLG                                                V75
*A*       JAZ       SOF             JUMP IF NOT V75                       V75
*A*       LDR       0,1             TEST FOR RX CODE                      V75
*A*       TBA                                                             V75
*A*       SUBI      'R0'                                                  V75
*A*       JAN       OUT             NOT AN R REGISTER                     V75
*A*       TBA                                                             V75
*A*       SUBI      'R7'+1                                                V75
*A*       JAP       OUT             NOT AN R REGISTER                     V75
*A*       TBA                                                             V75
*A*       ADDI      RTABLE-'R0'     GET PROPER RTABLE ENTRY               V75
*A*       TAR                                                             V75
*A*       LDA       0,B             GET REGISTER CODE IN THE FD           V75
*A*LOOP   LDR       1,X             GET NEXT INPUT WORD FOR SPA           V75
*A*       LLRL      8               SHIFT 1 BYTE TO LEFT AND              V75
*A*       STA       0,X             STORE BACK                            V75
*A*       LSRB      8                                                     V75
*A*       LDA       1,X             CHECK IF NO MORE INPUT CHAR           V75
*A*       SUBI      '  '            SPACE SPACE ?                         V75
*A*       JAZ       OUT             IF YES, GET OUT                       V75
*A*       TBA                                                             V75
*A*       TXR                       X REG POINTS TO INPUT BUF             V75
*A*       JMP       LOOP                                                  V75
*A*OUT    LDXI      BUF             RESTORE X TO BEGINNING OF B           V75
*A*5       CONT                                                          V75
    .INSERT,532                     V$DEBUG   V75                 METR    V75
*A*        IFT      V75-1                                                 V75
*A*        GOTO     1                                                     V75
*A**                                                                      V75
*A**      RTABLE CONVERTS A TWO LETTER REGISTER TO A ONE LE               V75
*A**      REGISTER CODE. FOR V75 USE ONLY (RIGHT JUSTIFIED)               V75
*A**                                                                      V75
*A*RTABLE EQU       *                                                     V75
*A*       DATA      0301            R0--0A                                V75
*A*       DATA      0302            R1--0B                                V75
*A*       DATA      0330            R2--0X                                V75
*A*       DATA      0312            R3--0J                                V75
*A*       DATA      0313            R4--0K                                V75
*A*       DATA      0314            R5--0L                                V75
*A*       DATA      0315            R6--0M                                V75
*A*       DATA      0316            R7--0N                                V75
```

```
*A**                                                                    V75
*A*1        CONT                                                        V75
   .INSERT,560                  V$DEBUG   V75                 METR      V75
*A*         IFT     V75-1                                               V75
*A*         GOTO    2                                                   V75
*A*        LDA     V75FLG        TEST IF V75                             V75
*A*        JAZ     BRK2          NO, JUMP                                V75
*A*        SAVE75  R3P,R4P,R5P,R6P,R7P                                   V75
*A*BRK2    EQU     *                                                    V75
*A*2        CONT                                                        V75
   .REPLACE,583                 V$DEBUG   V75       05/30/75  MCMURRAY  V75
*D*        SUB     ANUM
*A*        SUBI    OP+1                                                E.2
   .INSERT,585                  V$DEBUG   V75                 METR      V75
*A*         IFT     V75-1                                               V75
*A*         GOTO    1                                                   V75
*A**       IF V75 PRINT OUT R3 THRU R7                                   V75
*A**                                                                    V75
*A*        LDA     V75FLG                                              V75
*A*        JAZ     T4                                                  V75
*A*        LDX     TOBF          OUTPUT BUF ADDRESS                      V75
*A*        LDA     DBLK          SPACE                                   V75
*A*        STA     0,X                                                 V75
*A*         IXR                                                         V75
*A*        LDBI    R7P+1                                               V75
*A*        STB     RLIMIT                                              V75
*A*        LDBI    R3P                                                 V75
*A*T3      STB     TEMP                                                V75
*A*        LDB     0,2           REGISTER VALUE                          V75
*A*        CALL    CNV3          CONVERT AND STORE                       V75
*A*        LDB     TEMP                                                V75
*A*        INCR    023                                                 V75
*A*        SUB     RLIMIT                                              V75
*A*        JAN     T3                                                  V75
*A*        WRITE   DCBXR,DO,W,0                                        V75
*A*T4      EQU     *                                                    V75
*A*1        CONT                                                        V75
   .INSERT,589                  V$DEBUG   V75                 METR      V75
*A*         IFF     V75-1                                               V75
*A*RLIMIT DATA     0                                                   V75
   .INSERT,644                  V$DEBUG   V75                 METR      V75
*A*         IFT     V75-1                                               V75
*A*         GOTO    1                                                   V75
```

```
*A*R7      TBR                    R7-R3 ARE THE V75 REGISTERS                V75
*A*R6      TBR                                                               V75
*A*R5      TBR                                                               V75
*A*R4      TBR                                                               V75
*A*R3      TBR                                                               V75
*A*1        CONT                                                             V75
   .INSERT,690                    V$DEBUG   V75                  MEIR        V75
*A*         IFT     V75-1                                                    V75
*A*         GOTO    1                                                        V75
*A*         LDA     V75FLG                                                   V75
*A*         JAZ     GG1                                                      V75
*A**                RESTORE R3 THRU R7 (V75)                                 V75
*A*         LUAD75  R3P,R4P,R5P,R6P,R7P                                      V75
*A*GG1      EQU     *                                                        V75
*A*1        CONT                                                             V75
   .INSERT,697                    V$DEBUG   V75                  MEIR        V75
*A*         IFF     V75-1                                                    V75
*A*V75FLG DATA      0                                                        V75
   .INSERT,701                    V$DEBUG   V75                  MEIR        V75
*A*         IFT     V75-1                                                    V75
*A*         GOTO    1                                                        V75
*A*R3P      DATA    0                                                        V75
*A*R4P      DATA    0                                                        V75
*A*R5P      DATA    0                                                        V75
*A*R6P      DATA    0                                                        V75
*A*R7P      DATA    0                                                        V75
*A*1        CONT                                                             V75
   .REPLACE,893                   V$DEBUG   NO SMR    05/16/75   KERNS       E.2
*D*         STAT    DINRD,DICONT,PCEOF,PCEOF                                 D.1
*A*         STAT    DINRD,DICONT,PCEOF,PCEOF,*                               E.2
   .INSERT,946                    V$DEBUG   V75                  MEIR        V75
*A*         IFF     V75-1                                                    V75
*A*DCBXR    DCB     20,LOBF,0          R3 THRU R7                            V75
   .REPLACE,964,965               V$DEBUG   V75                  MEIR        V75
*D*TABL     DATA    A,BR,C,0,0,0,G,0,I,0,0,0                                 D.1
*D*         DATA    0,0,0,P,0,0,S,T,0,0,0,XR,0,0                             V75
*A*         IFT     V75-1                                                    V75
*A*         GOTO    1                                                        V75
*A**                R0,R1,R2 ARE A,B,X RESPECTIVELY                          V75
*A**                R3 THRU R7 ARE J,K,L,M,N RESPECTIVELY                    V75
*A**                                                                         V75
*A*TABL     DATA    A,BR,C,0,0,0,G,0,I,R3,R4,R5                              V75
*A*         DATA    R6,R7,0,P,0,0,S,T,0,0,0,XR,0,0                           V75
```

```
*A*1        CONT                                                    V75
*A*         IFF     V75-1                                           V75
*A*         GOTO    2                                               V75
*A*TABL     DATA    A,BR,C,0,0,0,G,T,0,0,0                          V75
*A*         DATA    0,0,0,P,0,0,S,T,0,0,0,XR,0,0                    V75
*A*2        CONT                                                    V75
```

```
   .INSERT,1                        VSSNAP    V75                  MEIR      V75
*A*V75      SET      1                                                       V75
   .INSERT,64                       VSSNAP    V75                  MEIR      V75
*A*BSO      EQU      MT+1                                                    V75
*A*TB75     EQU      2                                                       V75
*A*TB75B    EQU      14                                                      V75
*A*TB75Z    EQU      1                                                       V75
*A*         IFT      V75-1                                                   V75
*A*         GOTO     1                                                       V75
*A**                          TEST BI MACRO                                  V75
*A*TESTF    MAC                                                              V75
*A*         LDA      P(2),P(1)        PICK UP WORD CONTAINING                V75
*A*         ANA      BSO+P(3)                                                V75
*A*         EMAC                                                             V75
*A**                                  SAVE AND RESTORE V75 REG               V75
*A*SAVE75 MAC                                                                V75
*A*         ST,3     P(1)                                                    V75
*A*         ST,4     P(2)                                                    V75
*A*         ST,5     P(3)                                                    V75
*A*         ST,6     P(4)                                                    V75
*A*         ST,7     P(5)                                                    V75
*A*         EMAC                                                             V75
*A*LOAD75 MAC                                                                V75
*A*         LD,3     P(1)                                                    V75
*A*         LD,4     P(2)                                                    V75
*A*         LD,5     P(3)                                                    V75
*A*         LD,6     P(4)                                                    V75
*A*         LD,7     P(5)                                                    V75
*A*         EMAC                                                             V75
*A*1        CONT                                                             V75
   .REPLACE,90                      VSSNAP    V75                  MEIR      V75
*D*         PASS     29,*-*,SNBUF                                            V2
*A*         PASS     37,*-*,SNBUF                                            V75
   .INSERT,99                       VSSNAP    V75                  MEIR      V75
*A*         IFT      V75-1                                                   V75
*A*         GOTO     2                                                       V75
*A**                          TEST IF A V75 MACHINE                          V75
*A*         LDA      TB75,X   V75 FLAG IN TIDB                               V75
*A*         ANAI     040000                                                  V75
*A*         STAE     V75FLG   SET/RESET V75 FLAG                             V75
*A*2        CONT                                                             V75
   .INSERT,110                      VSSNAP    V75                  MEIR      V75
*A*         IFT      V75-1                                                   V75
```

```
*A*          GOTO     3                                                    V75
*A*          TESTF    B,TB75,TB75B,TB757                                   V75
*A*          STAE     V75FLG                                               V75
*A*3         CONT                                                          V75
   .INSERT,112                    V$SNAP    V75                  MEIR     V75
*A*          IFT      V75-1                                                V75
*A*          GOTO     4                                                    V75
*A**                          SAVE V75 REGISTERS                           V75
*A*          LDAE     V75FLG                                               V75
*A*          JAZ      EFE      JMP IF NOT V75                              V75
*A*          SAVE75   R3P,R4P,R5P,R6P,R7P                                  V75
*A*EFE       EQU      *                                                    V75
*A*4         CONT                                                          V75
   .INSERT,128                    V$SNAP    V75                  MEIR     V75
*A*          IFT      V75-1                                                V75
*A*          GOTO     5                                                    V75
*A**                          PRINT   V75     REGISTERS                    V75
*A*          LDA      V75FLG                                               V75
*A*          JAZ      SN12     JMP IF NOT A V75                            V75
*A*          LDXT     SNR3                                                 V75
*A*          LDB      R3P                                                  V75
*A*          CALL     CONV     R3                                          V75
*A*          LDB      R4P      R4                                          V75
*A*          CALL     CONV                                                 V75
*A*          LDB      R5P      R5                                          V75
*A*          CALL     CONV                                                 V75
*A*          LDB      R6P      R6                                          V75
*A*          CALL     CONV                                                 V75
*A*          LDB      R7P      R7                                          V75
*A*          CALL     CONV                                                 V75
*A*          WRITE    SNDCBR,DO,W,1   PRINT REGISTERS R3-R7                V75
*A*SN12      EQU      *                                                    V75
*A*5         CONT                                                          V75
   .REPLACE,141,142               V$SNAP    NO SMR    04/21/75  KERNS      E.2
*D*          IFT      VORTEX-2                                             V2
*D*          GOTO     SN2                                                  V2
*A*          IFT      VORTEX-1      GO TO 1 IF VORTEX II                   E.2
*A*          GOTO     1                                                    E.2
*A*          TXA                    TIDB LOCATION                          E.2
*A*          ANAI     0177770       SET TO DUMP BOUNDARY                   E.2
*A*          STA      SNAPA         SET START DUMP ADDR                    E.2
*A*1         CONT                                                          E.2
*A*          IFT      V75-1                                                V75
```

```
*A*          GOTO     SN133                                                  V75
*A*          LDA      V75FLG                                                 V75
*A*          JAZ      SN133                                                  V75
*A*          LDA      SNAPA                                                  V75
*A*          ADDI     36                                                     V75
*A*          JMP      SN144                                                  V75
*A*SN133     EQU      *                                                      V75
   .INSERT,144                        V$SNAP    V75                  MEIR    V75
*A*SN144     EQU      *                                                      V75
   .REPLACE,147                       V$SNAP    NO SMR     04/21/75  KERNS   E.2
*D*SN2       CONT                                                            V2
   .REPLACE,149                       V$SNAP    SMR-       04/28/75  KERNS   E.2
*D*          WRITE    SNDMP,DO,0,1    OUTPUT DUMP HEADER                     C
   .INSERT,156                        V$SNAP    SMR-       04/28/75  KERNS   E.2
*A*          WRITE    SNDMP,DO,0,1      OUTPUT DUMP HEADER                   E.2
   .INSERT,263                        V$SNAP    V75                  MEIR    V75
*A*          IFT      V75-1                                                  V75
*A*          GOTO     6                                                      V75
*A*SNR       DATA     '  '                                                   V75
*A*SNR3      DATA     '          '      R3                                   V75
*A*SNR4      DATA     '          '      R4                                   V75
*A*SNR5      DATA     '          '      R5                                   V75
*A*SNR6      DATA     '          '      R6                                   V75
*A*SNR7      DATA     '          '      R7                                   V75
*A*6         CONT                                                            V75
   .INSERT,268                        V$SNAP    V75                  MEIR    V75
*A*          IFF      V75-1                                                  V75
*A*SNDCBR    DCB      21,SNR,0                                               V75
   .INSERT,273                        V$SNAP    V75                  MEIR    V75
*A*          IFT      V75-1                                                  V75
*A*          GOTO     7                                                      V75
*A*R3P       DATA     0                  R3                                   V75
*A*R4P       DATA     0          R4                                          V75
*A*R5P       DATA     0          R5                                          V75
*A*R6P       DATA     0          R6                                          V75
*A*R7P       DATA     0          R7                                          V75
*A*V75FLG    DATA     0          V75 FLAG. 1 MEANS V75 MACHINE               V75
*A*7         CONT                                                            V75
```

```
   .REPLACE,5                                                          E.2
*D**                                    RELEASED    03/01/74
*A**                                                                   E.2
   .REPLACE,831                                                        E.2
*D*FASCMB GEN      /N(FASCMA),10(STACK),23(POPDEL),LB3,5(0),6(0),AA4
*A*FASCMB GEN      /P(FASCMA+P),10(STACK),23(POPDEL),LB3,5(0),6(0),AA4  E.2
   .REPLACE,923                                                        E.2
*D*                C24(R7),12(A$SPEC),22(AONE)
*A*                C24(R9),12(A$SPEC),22(AONE)                         E.2
   .REPLACE,925                                                        E.2
*D*FORTS1 GEN      /*,10(OF$ALU),12(A$GPR),24(R7),14(INCA),16(CRY1),
*A*FORTS1 GEN      /*,10(OF$ALU),12(A$GPR),24(R9),14(INCA),16(CRY1),    E.2
   .REPLACE,940                                                        E.2
*D*                C12(A$GPR),24(R7)
*A*                C12(A$GPR),24(R9)                                   E.2
   .REPLACE,942                                                        E.2
*D*FORTS4 GEN      /N(FORTS7),10(IF$P),12(A$GPR),24(R7),14(INCA),
*A*FORTS4 GEN      /N(FORTS7),10(IF$P),12(A$GPR),24(R9),14(INCA),      E.2
   .REPLACE,1052                                                       E.2
*D*FORTS7 GEN      /P(SS3M),10(IF$P),12(A$GPR),24(R7),14(TRNA),
*A*FORTS7 GEN      /P(SS3M),10(IF$P),12(A$GPR),24(R9),14(TRNA),        E.2
   .REPLACE,1138                                                       E.2
*D*VORT17 GEN      /N(VORT18),12(A$GPR),24(R4),14(DECA)
*A*VORT17 GEN      /N(VORT18),12(A$GPR),24(R4),14(DECA),               E.2
*A*                C10(WAITMD),6(SPEC)                                 E.2
   .REPLACE,1182                                                       E.2
*D*                C11(B$SPEC),17(GPROUT),24(R7),       PTR IN 'P', TEST FOR
*A*                C11(B$SPEC),17(GPROUT),24(RD),      PTR IN 'P', TEST  E.2
   .REPLACE,1245                                                       E.2
*D*MCTOS  GEN      /N(MCTOSX),10(OS$ALU),6(MEMC),24(R7),13(POUT)
*A*MCTOS  GEN      /N(MCTOSX),10(OS$ALU),6(MEMC),24(RD),13(POUT)       E.2
   .REPLACE,1276                                                       E.2
*D*                C16(R7),11(LIT),MKFFFC,14(ADD),13(POUT)
*A*                C16(RD),11(LIT),MKFFFC,14(ADD),13(POUT)             E.2
   .REPLACE,1345,1347                                                  E.2
*D**       RESTORE P FROM R7 AND RETURN
*D*MBS24  GEN      /N(SS3M),7(PJMP$),10(IF$ALU),6(MEMC$),12(A$GPR),24(R7),
*D*                C14(INCA),16(CRY1),13(POUT)
*A**       RESTORE P FROM R9 AND RETURN                                E.2
*A*MBS24  GEN      /P(SS2M),10(PJMP),12(A$GPR),24(R9),14(TRNA),15(LOG), E.2
*A*                C13(POUT)                                           E.2
   .REPLACE,1456,1457                                                  E.2
*D**       SAVE P IN R7
```

```
*D*MBS    GEN     /N(MBS1),12(A$P),14(TRNA),15(LOG),17(GPROUT),24(R7)
*A**      SAVE P IN R9                                                    E.2
*A*MBS    GEN     /N(MBS1),12(A$P),14(TRNA),15(LOG),17(GPROUT),24(R9)    E.2
```

```
   .INSERT,46                                                          F
A*        NAME     MMEM,EMEM                                          F
A*MMEM    BSS      0200          MAIN MEMORY BLOCK                    F
A*EMEM    BES      0             END OF MAIN MEMORY                   F
A*BUFR    DATA     0120240       GENERAL USAGE BUFFER                 F
A**                                                                   F
A** NOTE--THIS SECTION WILL BE OVERLAYED BY THE BUFFER BUFR            F
A**                                                                   F
   .INSERT,101                   MICSIM    NO SMR    01/21/75  KERNS  F
A*        BSS      121+BUFR-*    FILL OUT TO MAKE 120 WORD BUFR SPACE F
   .REPLACE,118                  MICSIM    NO SMR    01/21/75  KERNS  F
D*$DBUF   BSS      2048          CONTROL STORE PAGE                     S
A*        EXT      $DBUF         CONTROL STORE PAGE 0                 F
   .REPLACE,124,126              MICSIM    NO SMR    01/21/75  KERNS  F
D*$DRM1   BSS      16            DCS =1                                 S
D*$DRM2   BSS      16            DCS =2                                 S
D*        BSS      96            DCS FOR PAGES 1,2,3                    S
A*        EXT      $DRM1         DECODER STORE PAGE 0 DECODE 1        F
A*        EXT      $DRM2         DECODER STORE PAGE 0 DECODE 2        F
A*        EXT      $DRM1B,$DRM1C,$DRM1D                               F
   .REPLACE,134,135              MICSIM    NO SMR    01/21/75  KERNS  F
D*BUFR    DATA     0120240                                              S
D*        BSS      120                                                  S
   .REPLACE,155                  MICSIM    NO SMR    01/21/75  KERNS  F
D*CBUF    BSS      512           BUFFER TO HOLD COUNTS FOR NUMBER OF TIMES S
   .INSERT,160                   MICSIM    SMR-      04/21/75  KERNS  E,2
A*PPAG    DATA     0                                                  E,2
   .INSERT,359                   MICSIM    NO SMR    01/25/75  KERNS  F
A*        DATA     0332          7  ZERO TABLES                       F
   .INSERT,374                   MICSIM    NO SMR    01/25/75  KERNS  F
A*        DATA     EXCS          7  ZERO TABLES                       F
   .INSERT,377                   MICSIM    NO SMR    01/27/75  KERNS  F
A*        NAME     EXC                                                F
   .REPLACE,384,386              MICSIM    NO SMR    01/25/75  KERNS  F
D*        JMPM     SAR           HOUSEKEEP TABLES                       S
D*        DATA     COMM-STEP     WORD COUNT:CLEAR MEM FROM STEP TO COMM  C
D*        DATA     STEP          START ADDRESS                          C
   .INSERT,432                   MICSIM    SMR-      01/21/75  KERNS  F
A*EXC1A   CALL     SIOUT,7,EXC5  REQUEST MEMORY TYPE                  F
A*        CALL     SITN          GET MEMORY TYPE                      F
A*        JMPM     FETCH                                              F
A*        SUBI     0260                                               F
A*        TAB                                                         F
```

```
:A*          SUBI    3                                                    F
:A*          JAN     EXC1A           LT 2 NOT VALID                       F
:A*          SUBI    3                                                    F
:A*          JAP     EXC1A           GT 5 NOT VALID                       F
:A*          STR     MTYP            SAVE MEMORY TYPE                     F
:A*EXCS      EQU     *                                                    F
:A*          TZA                                                          F
:A*          JMPM    SAR             ZERO ALL TABLES AND FLAGS            F
:A*          DATA    COMM-V          WORD COUNT                           F
:A*          DATA    V               START ADDR                           F
:A*          DECR    01                                                   E,2
:A*          STA     R5              SET R5 TO ALL 1'S FOR 620 EMULATION  E,2
   .INSERT,438                       MICSIM    SMR-      01/21/75  KERNS  F
:A*EXCS      DATA    '  MEMORY TYPE?'                                     F
:A*MTYP      DATA    0               MEMORY TYPE, 3=SC,4=CORE,5=SLOW CORE F
   .INSERT,491                       MICSIM    SMR-      04/21/75  KERNS  E,2
:A*          STAE    PPAG                                                 E,2
   .REPLACE,500,502                  MICSIM    SMR-      01/21/75  KERNS  F
:D*          DATA    $DRM1+32        PAGE 1 DCS                           S
:D*          DATA    $DRM1+64        PAGE 2 DCS                           S
:D*          DATA    $DRM1+96        PAGE 3 DCS                           S
:A*          DATA    $DRM1B          PAGE 1 DCS                           F
:A*          DATA    $DRM1C          PAGE 2 DCS                           F
:A*          DATA    $DRM1D          PAGE 3 DCS                           F
   .REPLACE,546                      MICSIM    NO SMR    01/25/75  KERNS  F
:D*          CALL    SIOUT,21,PAGS   OUTPUT THE REQUEST                   S
:A*          CALL    SIOUT,7,PAGS    OUTPUT REQUEST                       F
   .REPLACE,572                      MICSIM    NO SMR    01/25/75  KERNS  F
:D*PAGS      DATA    '  INPUT HIGHEST NUMBER WCS PAGE DESIRED  '          S
:A*PAGS      DATA    '  PAGE LIMIT? '                                     F
   .INSERT,634                       MICSIM    SMR-      04/21/75  KERNS  E,2
:A*          LDA     EXLIM           EXECUTION LIMIT                      E,2
:A*          STA     RCNT            RESET RUN COUNT                      E,2
:A*          LDA     CPAG                                                 E,2
:A*          STA     PPAG            SET CURRENT MICRO PAGE               E,2
   .REPLACE,642                      MICSIM    SMR-      04/21/75  KERNS  E,2
:D*          CALL    S120            CHECK FOR ROM HALT ADDRESS           S
   .REPLACE,646,653                  MICISM    SMR-      01/21/75  KERNS  F
:D**   CHECK IF REQUIRED TO WAIT ON MEMORY                                 S
:D**   YES IF S=0&IMC=0001                                                 S
:D*S122      LDA     XS,1            S=0 ?                                S
:D*          JAZ     *+4                                                  S
:D*          JMP     S103            NO-NO WAIT                           S
```

```
*D*        LDA     XIMC,1                                                  S
*D*        DAR                     IMC=0001                                S
*D*        JAZ     S107            YES-WAIT                                S
   .INSERT,659                     MICSIM    NO SMR    04/21/75  KERNS     E,2
*A*        LDA     CPAG            NEXT MICRO PAGE                         E,2
*A*        STA     PPAG            SET CURRENT PAGE TO NEXT PAGE           E,2
   .REPLACE,662,664                MICISM    SMR-      01/21/75  KERNS     F
*D*S107    CALL    MOPA            MEMORY OPERATIONS                       S
*D*        CALL    S300            DATA LOOP PROCESSING                    S
*D*        JMP     S108                                                    S
   .INSERT,670                     MICSIM    SMR-      04/21/75  KERNS     E,2
*A*        JSS3    EXC10           RETURN TO EXEC IF ABORT RUN SET         E,2
*A*        CALL    S120            CHECK FOR ROM HALT ADDR                 E,2
   .REPLACE,721,723                MICSIM    NO SMR    01/23/75  KERNS     F
*D*        LDA     MOPC,1                                                  D,1
*D*        JAZ     S205            IF NO MEM REQ                           D,1
*D*        TAX                                                             S
*A*        LDB     MOPC,1                                                  F
*A*        JBZ     S205            IF NO MEM REQ                           F
*A*        TAX                                                             F
   .REPLACE,727,729                MICSIM    SMR-      04/21/75  KERNS     E,2
*D*        LDX     CPAG                                                    D
*D*        LDAE    TRACE,1         TRACE FLAG FOR PAGE                     D
*D*        JAZ     S201            JUMP IF NOT TRACE MODE                  S
   .REPLACE,742                    MICSIM    NO SMR    11/23/74  KERNS     FF
*D*        CALL    LOOUT,9,5219    OUTPUT TO LO                            D,1
*A*        CALL    LOOUT,9,S219    OUTPUT TO LO                            FF
   .REPLACE,767,770                MICSIM    NO SMR    01/21/75  KERNS     F
*D*        LDA     NROM,1          INCREMENT EXECUTION                     S
*D*        ADDI    CBUF            COUNT FOR                               S
*D*        TAB                     THIS                                    S
*D*        INR     0,2             CONTROL ROM WORD.                       S
   .REPLACE,805                    MICSIM    NO SMR    12/18/74  KERNS     FF
*D*        JMP     CRA4            YES                                     S
*A*        JMP     CRAX            YES                                     FF
   .REPLACE,830                    MICSIM    NO SMR    11/23/74  KERNS     FF
*D*        JMP     CRA7            NO-                                     S
*A*        JMP     CRA8            NO-                                     FF
   .REPLACE,866                    MICSIM    NO SMR    11/22/74  KERNS     FF
*D**   DON'T USE TS FIELD FOR B-OFFSET IF I/O REQUEST(IMC=3)                S
*A*        JMP     CRA7            IF S=1 OR 2 THEN USE TS FOR OFFSET      F
*A**   DON'T USE TS FIELD FOR B-OFFSET IF I/O REQUEST (IM=E/F)              FF
   .INSERT,867                     MICSIM    NO SMR    11/22/74  KERNS     FF
```

```
*A*         SUBI    14                                                  FF
*A*         JAZ     CRA6            IM=E                                FF
*A*         DAR                     IM=F                                FF
   .REPLACE,1004,1007               MICSIM   NO SMR   12/18/74  KERNS   FF
*D*         STA     CPAG            BRANCH PAGE                          S
*D*         LDA     PAG4            PAGE LIMIT                           S
*D*         SUB     CPAG                                                 S
*D*         JAP     *+4             LEGAL JMUP                           S
*A*         CALL    PGOK            CHECK IF LEGAL PAGE                 FF
*A*         JAN     *+4             YES                                 FF
   .REPLACE,1418,1420               MICSIM   SMR-     04/21/75  KERNS    E,2
*D*         LDBI    TRACE                                                S
*D*         LDA     0,2             TRACE FLAG ON ?                      S
*D*         JAZ*    CRXA            NO-EXIT                              S
*A*         JMPM    TRSETA          CHECK IF WITHIN TRACE BOUNDS        E,2
*A*         JAN*    CRXA            RETURN IF NOT TO TRACE              E,2
   .REPLACE,1494,1496               MICSIM   SMR-     04/21/75  KERNS    E,2
*D*S305     LDB     CPAG            CURRENT PAGE                        C,1
*D*         LDAE    TRACE,2         TRACE FLAG FOR PAGE                 C,1
*D*         JAZ*    $300            JUMP IF NOT TRACE MODE               S
*A*S305     EQU     *                                                   E,2
   .REPLACE,1653,1655               MICSIM   SMR-     04/21/75  KERNS    E,2
*D*         LDB     CPAG                                                 D
*D*         LDAE    TRACE,2         TRACE FLAG FOR PAGE ON ?             D
*D*         JAZ*    LTMX            NO-EXIT                              S
   .REPLACE,2160                    MICSIM   SMR-     04/21/75  KERNS    E,2
*D*         JBZ     X026                                                 S
*A*         STB     XCIN,1          CLEAR CIN IF NOT TO BE SET          F,2
*A*         JBZ     X026+1                                              E,2
   .INSERT,2560                     MICSIM   NO SMR   04/21/75  KERNS    E,2
*A*X096     EQU     *                                                   E,2
   .REPLACE,2564                    MICSIM   NO SMR   04/21/75  KERNS    E,2
*D*X096     LDA     X099            ALU RESULT                           S
*A*X096A    LDA     X099            ALU RESULT                          F,2
   .REPLACE,2572                    MICSIM   NO SMR   04/21/75  KERNS    E,2
*D*         JMP     X096            GET ALU RESULT AND EX                S
*A*         JMP     X096A           GET ALU RESULT AND EX               E,2
   .INSERT,2580                     MICSIM   SMR-     11/27/74  KERNS    FF
*A*         LDA     XS,1                                                FF
*A*         SUBI    2                                                   FF
*A*         ORA     XT,1                                                FF
*A*         JAZ     X09C            IF SF=2 AND TF=0                    FF
   .REPLACE,2783,2792               MICSIM   NO SMR   04/21/75  KERNS    E,2
```

```
*D**   TEST BIT 13: STORES SIGN BIT(DAL15) OF ALU OUTPUT WHENEVER          S
*D**                 FILE REG.0(A REG) IS WRITTEN INTO.                    S
*D*        LDA     XW,1          DON'T UPDATE                              S
*D*        JAZ     XS30          A15 F/F IF NOT                            S
*D*        LDA     XLB,1         LB IS SPECIFYING                          S
*D*        SUBI    2             MASK FIELD                                S
*D*        JAP     XS30          DO NOT SET A15 F/F.                       S
*D*        LDA     XA,1          WRITING INTO                              S
*D*        JAZ     *+4                                                     S
*D*        JMP     XS30                                                    S
*A**   TEST BIT 13: STORES SIGN BIT(DAL15) OF R0 IN A0R0                 E.2
*A*        LDA     R0                                                      E.2
   .REPLACE,2848,2849            MICSIM    SMR-      11/27/74  KERNS     FF
*D*        JAZ     *+4                                                     S
*D*        JMP     XS90          NO                                        S
*A*        JAZ     XS64A         IF SF=1                                 FF
*A*        DAR                                                           FF
*A*        ORA     XT,1                                                  FF
*A*        JAZ     *+4           IF SF=2 AND TF=0                        FF
*A*        JMP     XS90          IF SF=3 OR TF NE 0                      FF
*A*        LDA     XG,1                                                  FF
*A*        ANAI    2                                                     FF
*A*        JAZ     XS90                                                  FF
*A*        JMP     XS65          IF SF=2, TF=0, AND GF=XX1X              FF
*A*XS64A   EQU     *                                                     FF
   .REPLACE,3255                 MICISM    SMR-      01/21/75  KERNS      F
*D**       (SC MEMORY)                                                     S
   .REPLACE,3276,3277            MICISM    SMR-      01/21/75  KERNS      F
*D**       MCCO=1: ACTIVE BUT NOT DONE                                     S
*D**       MCCO=2: ACTIVE AND DONE                                         S
*A**       MCCO=1: IDLE WITH REQUEST PENDING                             F
*A**       MCCO=2-N: ACTIVE  (N=MEMORY TYPE)                             F
   .REPLACE,3285                 MICSIM    SMR-      01/21/75  KERNS      F
*D**   RETURN: MEM OPERATIONS COMPLETE                                     S
   .REPLACE,3296                 MICSIM    SMR-      01/21/75  KERNS      F
*D*        JMP     M032          0=IDLE 1=ACTIVE BUT NOT DONE 2=ACTIVE +DONS
*A*        JMP     M032          MEMORY IS ACTIVE                        F
   .INSERT,3375                  MICSIM    SMR-      01/21/75  KERNS      F
*A*        TZA                                                           F
*A*        STA     MLTP          CLEAR COMPLETE MEMORY FLAG              F
   .REPLACE,3378                 MICSIM    SMR-      01/21/75  KERNS      F
*D**   COME HERE IF MCCO IS 1 OR 2                                         S
*A**   COME HERE IF MCCO IS 1 TO N  (MEMORY ACTIVE) 01/21/75  KERNS      F
```

```
   .REPLACE,3381,3382                MICSIM    SMR-      01/21/75  KERNS     F
*0*        DAR                       MCCO=2                                    S
*0*        JAZ       M050            YES                                       S
*A*        JMP       M036            CHECK FOR COMPLETION                      F
   .REPLACE,3385                     MICSIM    SMR-      01/21/75  KERNS     F
*0*        JAZ       M050            YES                                       S
*A*        JAZ       M050            YES                                       F,2
   .REPLACE,3388,3392                MICSIM    SMR-      01/21/75  KERNS     F
*0**   MCCO=1 AND NO OVERRIDE                                                  S
*0*        INCR      01              A=1                                       S
*0*        STA       MLTP            FLAG: COMPLETE MEMORY OPERATION           S
*0*M036    INR       MCCO,1          SET MCCO=2 (ACTIVE&DONE)                  S
*0*        JMP       M070            RTN VIA LISTING ROUTINE                   S
*A*        TZA                                                                 F
*A*        STA       MPLE            CLEAR MPLE FLAG                           F
*A*        LDA       MOPC,1          CHECK IF STORE REQ                        F
*A*        SUBI      3                                                         F
*A*        JAN       M036            NO                                        F
*A*        LDA       LREG,1                                                    F
*A*        SUB       PREG,1          IF LREG=PREG, SET MPLE                    F
*A*        JAZ       *+4             YES                                       F
*A*        JMP       M036            NO                                        F
*A*        INCR      1                                                         F
*A*        STA       MPLE                                                      F
*A**   MEMORY REQUESTED OR ACTIVE                                              F
*A*M036    INR       MCCO,1          UPDATE MEMORY CONDITION CODE              F
*A*        JMP       M040            CHECK FOR MEMORY COMPLETE OR WAIT         F
   .REPLACE,3399,3400                MICSIM    SMR-      01/21/75  KERNS     F
*0**   COME HERE IF MCCO=2                                                      S
*0**   PERFORM SPECIFIED MEMORY OPERATIONS                                      S
*A**   ENTER HERE IF MEMORY IS REQUESTED OR ACTIVE                             F
*A**                                                                            F
*A**    MEMORY CYCLE COMPLETE?                                                  F
*A*M040    EQU       *                                                         F
*A*        TZA                                                                 F
*A*        STA       MLTP            CLEAR COMPLETE MEMORY FLAG                F
*A*        LDA       MCCO,1                                                    F
*A*        SUB       MTYP                                                      F
*A*        JAP       M050            IF MEMORY DONE                            F
*A**   WAIT FOR MEMORY DONE?                                                   F
*A*        LDA       YS,1                                                      F
*A*        JAZ       *+4             IF S=0                                    F
*A*        JMP       M041                                                      F
```

```
*A*         LDA     XIMC,1                                                  F
*A*         DAR                                                             F
*A*         IAZ     M045            IF IMC=1                                F
*A**    PENDING MEMORY REQUEST?                                             F
*A*M041     CALL    MREQ            ANY NEW MEMORY REQUESTS                 F
*A*         IAZ     M070            NO, RETURN VIA LISTING ROUTINE          F
*A*         IMP     M045            COMPLETE CURRENT REQUEST                F
*A*M045     EQU     *                                                       F
*A*         LDA     MTYP            MEMORY COMPLETION COUNT                 F
*A*         STA     MCCO,1          SET COMPLETION CODE TO COMPLETE         F
*A*         INCR    1                                                       F
*A*         STA     MLTP            SET COMPLETE MEMORY FLAG                F
*A*         IMP     M070            RETURN VIA LISTING ROUTINE              F
    .INSERT,3432                    MICSIM    SMR-      01/25/75  KERNS     F
*A*         STA     MLTP            CLEAR COMPLETE MEMORY FLAG              F
    .REPLACE,3447,3450              MICSIM    SMR-      01/28/75  KERNS     F
*0**    IF PREG=LREG, WRITE OUTPUT INTO IRG1(FOR PIPELINE PURPOSES)           S
*0*         LDA     LREG,1                                                  S
*0*         SUB     PREG,1                                                  S
*0*         IAZ     *+4                                                     S
*A**    IF MPLE SET, WRITE OUTPUT INTO IRG1 (FOR PIPELINE PURPOSES)         F
*A*         LDA     MPLE                                                    F
*A*         DAR                                                             F
*A*         IAP     *+4             IF MPLE SET                             F
    .INSERT,3453                    MICSIM    SMR-      01/28/75  KERNS     F
*A*M059     TZA                                                             F
*A*         STA     MPLE            CLEAR MPLE                              F
    .REPLACE,3477,3480              MICSIM    SMR-      01/28/75  KERNS     F
*0**    IF PREG>LREG, PUT NEW MEMORY WORD INTO IRG1,FOR PIPELINE PURPOSES)    S
*0*M062     LDA     LREG,1                                                  S
*0*         SUB     PREG,1                                                  S
*0*         IAZ     *+4                                                     S
*A**    IF MPLE SET, PUT NEW MEMORY WORD INTO IRG1(FOR PIPELINE PURPOSES)F
*A*M062     LDA     MPLE                                                    F
*A*         DAR                                                             F
*A*         IAP     *+4             IF MPLE SET                             F
    .REPLACE,3484                   MICSIM    SMR-      01/28/75  KERNS     F
*0*         IMP     M053                                                    S
*A*         IMP     M059                                                    F
    .REPLACE,3500                   MICSIM    SMR-      01/27/75  KERNS     F
*0*         SUBI    01000                                                   S
*A*         SUBI    MMEM            START OF MAIN MEMORY                    F
    .REPLACE,3632,3634              MICSIM    SMR-      04/21/75  KERNS     E,2
```

```
*D*MOZO   LDB     CRAG                                                  D
*D*       LDAE    TRACE,2       TRACE FLAG FOR PAGE                     D
*D*       JAZ     MOZ2                                                    S
*A*MOZO   EQU     *                                                     E.2
   .INSERT,3697                 MICSIM    SMR-      01/28/75  KERNS     F
*A*MPLE   DATA    0             PIPELINE FLAG                           F
   .REPLACE,3796,3802           MICSIM    NO SMR    04/21/75  KERNS     F.2
*D*SETMUX LDXI    16                                                    D
*D*       TZB                                                           D
*D*       LLRL    1             POSITION STATUS BIT                     D
*D*       DXR                                                           D
*D*       STBE    TMUX,1        STORE IN TMUX TABLE                     D
*D*       JXZ     FAC1          IF ALL 16 BITS SET                      D
*D*       JMP     SETMUX+1                                              D
*A*SETMUX LDAI    0             STATUS WORD VALUE                       E.2
*A*       TZB                                                           E.2
*A*       LLRL    7                                                     E.2
*A*       LDXI    4             SET FOLLOWING TEST CONDITIONS           E.2
*A*SETM1  STBE    TMUX+5,1                ALUZ                          E.2
*A*       TZB                             ALUC                          E.2
*A*       DXR                             ALUS                          E.2
*A*       JXZ     SETM3                   ALUO                          E.2
*A*       LLRL    1                                                     E.2
*A*       JMP     SETM1                                                 E.2
*A*SETM3  LLRL    6             POSITION TO OVFL BIT                    E.2
*A*       STBE    TMUX                                                  E.2
*A*       JMP     FAC1                                                  E.2
   .INSERT,3907                                                         F
*A*       STA     LDCTL         SAVE LOAD TYPE                          F
   .INSERT,3912                                                         F
*A*       SUBI    'M'-'D'       LOADING MAIN MEMORY                     F
*A*       JAZ     LDMM                                                  F
   .INSERT,3935                                                         F
*A*LDMM   EQU     *                                                     F
   .INSERT,3992                                                         F
*A*       LDA     LDCTL         LOAD TYPE                               F
*A*       SUBI    'M'                                                   F
*A*       JAZ     LDMEM         JUMP IF LOAD MAIN MEMORY                F
   .INSERT,4032                                                         F
*A*LDMEM  EQU     *             LOAD MAIN MEMORY                        F
*A*       LDA     0,1           GET WORD                                F
*A*       LSRA    13            FETCH LOAD ADDR                         F
*A*       TAB                                                           F
```

```
*A*         LDAE    CODE,2          CODE SERVICE ADDR                       F
*A*         IAZ     LDCDE           ERROR IF UNSERVICED CODE                F
*A*         STA     LDSRV+2         SET SERVICE ADDR                        F
*A*         IMPM    LDSRV           PERFORM SERVICE                         F
*A*         LDA     0,1                                                     F
*A*         LSRA    8               FETCH SUBCODE                           F
*A*         ANAI    017                                                     F
*A*         TAR                                                             F
*A*         LDAE    SCOD,2          SUBCODE SERVICE ADDR                    F
*A*         IAZ     LDCDE           ERROR IF UNSERVICED SUBCODE             F
*A*         STA     LDSRV+2                                                 F
*A*         IMPM    LDSRV           PERFORM SERVICE                         F
*A*         LDA     0,1                                                     F
*A*         LSRA    4               FETCH POINTER                           F
*A*         ANAI    037                                                     F
*A*         SUBI    037                                                     F
*A*         IAZ     LDPTR           ONLY ABSOLUTE IS VALID POINTER          F
*A*         JMP     LDCDE           ERROR                                   F
*A**                                                                        F
*A*LDPTR    JMP     0               ADDRESS SET BY SERVICE ROUTINES         F
*A*         EJEC                                                            F
*A*LDLDR    ENTR                    CODE+4, LOAD DATA WORDS                 F
*A*         JMP     BUFPTR                                                  F
*A*         LDA     0,1                                                     F
*A*         ANA     LSB13           ISOLATE WORD                            F
*A*         TAR                                                             F
*A*         STA     LDMV+2          SET MOVE COUNT                          F
*A*         LDA     BUFPTR                                                  F
*A*         STA     LDMV+3          SET FROM  ADDR                          F
*A*         ADD     LDMV+2                                                  F
*A*         STA     BUFPTR          UPDATE BUFFER POINTER                   F
*A*         LDA     LDADR                                                   F
*A*         STA     LDMV+4          SET TO ADDR                             F
*A*         ADD     LDMV+2          CALCULATE RANGE                         F
*A*         STA     LDADR           UPDATE LOAD ADDR                        F
*A*         SUBI    EMEM                                                    F
*A*         JAP     LDSIZ           ERROR IF MAIN MEMORY EXCEEDED           F
*A*LDMV     CALL    MOVW,0,0,0      MOVE DATA WORDS TO MEMORY               F
*A*         JMP     LDWD            CONTINUE WITH OBJECT RECORDS            F
*A**                                                                        F
*A*LDADR    ENTR                    SET LOAD ADDR                           F
*A*         LDAI    LDADD                                                   F
*A*         STA     LDPTR+1         SET FOR ORG ADDR ROUTINE                F
```

```
*A*          RETU*   LDMOR                                                    F
*A**                                                                          F
*A*LDMEX     ENTR                    SET EXECUTION ADDR                       F
*A*          IMP     ICOM            EXECTION ADDR IS END OF LOAD             F
*A**                                                                          F
*A*LDADR     EQU     *                                                        F
*A*          INR     BUFPTR                                                   F
*A*          LDR     BUFPTR                                                   F
*A*          LDA     0,2             LOAD ADDR                                F
*A*          STA     LDADR           SET LOAD ADDR                            F
*A*          SUBI    MMEM                                                     F
*A*          TAN     LDSI7           IF BELOW MAIN MEMORY                     F
*A*          SUBI    EMEM-MMEM+1                                              F
*A*          TAP     LDSI7           IF ABOVE MAIN MEMORY                     F
*A*          INR     BUFPTR                                                   F
*A*          IMP     LDWD            CONTINUE LOAD                            F
*A**                                                                          F
*A*LDSRV     ENTR                                                             F
*A*          IMPM    0               SERVICE LOADER CODE (ADDR SET UPON CALL) F
*A*          RETU*   LDSRV                                                    F
*A**                                                                          F
*A*LDSCD     ENTR                    USE SUBCODE, CODE NOT USED               F
*A*          RETU*   LDSCD                                                    F
*A**                                                                          F
*A*LDIGN     ENTR                    IGNORE ENTRY                             F
*A*          INR     BUFPTR          BYPASS WORD                              F
*A*          IMP     LDWD            CONTINUE LOAD                            F
   .INSERT,4060                                                               F
*A*LDSI7     CALL    SIOUT,3,LSI7    OUTPUT ERROR                             F
*A*          IMP     EXC10                                                    F
*A**                                                                          F
*A** CODE SERVICE ADDR TABLE                                                  F
*A*CODE      DATA    LDSCD,0,0,0,LDIOR                                        F
*A** SUBCODE SERVICE ADDR TABLE                                               F
*A*SCOD      DATA    LDIGN,LDMOR,0,0,0,0,0,LDMEX                              F
*A*          DATA    0,0,0,0,0,0,0,0                                          F
*A**                                                                          F
*A*LDCTL     DATA    0               LOAD TYPE FLAG                           F
*A**                                                                          F
   .INSERT,4162                                                               F
*A*LSI7      DATA    !  MS151        MAIN MEMORY SIZE VIOLATED                F
   .INSERT,4211                       MTCSTM    NO-SMR    01/07/75  KERNS     F
*A*E114A     EQU     *                                                        F
```

```
   .INSERT,4278                    MICSIM    SMR-      04/21/75  KERNS     E.2
*A*        STA      EXLIM          SET EXECUTION LIMIT                     E.2
   .INSERT,4281                    MICSIM    NO SMR    04/21/75  KERNS     E.2
*A*        STA      EXLIM                                                  E.2
   .INSERT,4312                    MICSIM    SMR-      04/21/75  KERNS     E.2
*A*EXLIM   DATA     0              EXECUTION LIMIT                         E.2
   .REPLACE,4378                   MICSIM    NO SMR    01/21/75  KERNS     F
*D*        CALL     LOOUT,37,E158   OUTPUT HEADER                            S
*A*        CALL     LOOUT,26,E158   OUTPUT HEADER                          F
   .REPLACE,4423,4430              MICSIM    NO SMR    01/21/75  KERNS     F
*D*        TXR                     SET-UP TO LIST                           S
*D*        TXR                     EXECUTION COUNT NUMBER.                  S
*D*        TXR                                                              S
*D*        LDA      E158           ROM ADDR(0-512HEX)                       S
*D*        ADDI     CBUF           ADDR OF COUNT TABLE                      S
*D*        TAB                                                              S
*D*        LDB      0,2                                                     S
*D*        CALL     OH             CONVERT TO ASCII                         S
   .REPLACE,4431                   MICSIM    NO SMR    01/21/75  KERNS     F
*D*        CALL     LOOUT,36,BUFR   OUTPUT LINE                              S
*A*        CALL     LOOUT,31,BUFR   OUTPUT LINE                            F
   .REPLACE,4464                   MICSIM    SMR-      01/21/75  KERNS     F
*D*        DATA     '              EXECUTED'                                S
   .REPLACE,5049                   MICSIM    NO-SMR    01/07/75  KERNS     F
*D*        JMP      *+4                                                     S
*A*        STA      INHA                                                   F
*A*        SUBI     '='                                                    F
*A*        JAZ      INH6           IF ZERO BEFORE STORE FIELD VALUE        F
*A*        SUBI     '+'-'='                                                F
*A*        JAZ      INH7           IF DO NOT ZERO BEFORE STORE FIELD VALUE F
*A*        LDA      INHA                                                   F
*A*        JMP      INH4           CONTINUE PROCESSING                     F
   .INSERT,5051                    MICSIM    NO-SMR    01/07/75  KERNS     F
*A*INH4    EQU      *                                                      F
   .INSERT,5090                    MICSIM    NO-SMR    01/10/75  KERNS    FF
*A*INH6    LDB      E113           ROM WORD ADDR                           F
*A*        TZA                                                             F
*A*        STA      0,2            ZERO                                    F
*A*        STA      1,2            ENTIRE                                  F
*A*        STA      2,2            ROM                                     F
*A*        STA      3,2            WORD                                    F
*A*INH7    JMP      FLDCG          PROCESS FIELD CHANGE DIRECTIVE          F
   .INSERT,5100                    MICSIM    NO-SMR    01/10/75  KERNS    FF
```

```
*A*         EJEC                                                          F
*A**        THIS IS A SUBROUTINE TO HANDLE THE CHANGE CENTRAL CONTROL     F
*A**        STORE BY FIELD SPECIFICATION                                  F
*A**                                                                      F
*A**        CALLING SEQUENCE                                              F
*A**        JMP     FLDCC                                                 F
*A**                                                                      F
*A**        RETURNS TO STMULATOR EXEC FOR PROCESSING OF NEXT DIRECTIVE    F
*A**                                                                      F
*A*FLDCC    EQU     *                                                     F
*A*         IMPM    FETCHA                                                F
*A*FLD05    LRLA    8                                                     F
*A*         STA     FLD90           SAVE LEFT BYTE                        F
*A*         IMPM    FETCHA                                                F
*A*         ERA     FLD90           FORM 2 CHAR NAME                      F
*A*         STA     FLD90                                                 F
*A*         LDXI    FLNAM           FIELD NAME TABLE                      F
*A*FLD10    LDA     0,1             FIELD NAME                            F
*A*         TAZ     FLD60           IF END OF TABLE                       F
*A*         ERA     FLD90                                                 F
*A*         TAZ     FLD20           IF MATCH FOUND                        F
*A*         TXR     1                                                     F
*A*         TXR     1                                                     F
*A*         TXR     1                                                     F
*A*         IMP     FLD10           CONTINUE SCAN THROUGH TABLE           F
*A*FLD20    EQU     *                                                     F
*A*         STX     FLD92           SAVE TABLE PTR                        F
*A*         IMPM    TNA             FETCH CHANGE VALUE                    F
*A*         DATA    1,1                                                   F
*A*         STX     FLD93           SAVE TERM CHAR                        F
*A*         STA     FLD91           SAVE VALUE                            F
*A*         LDX     FLD92           FIELD TABLE PTR                       F
*A*         LDB     1,X             FIELD SIZE                            F
*A*         LDA     FLD25                                                 F
*A*         ANAI    07740                                                 F
*A*         MERGE   031             FORM RIGHT SHIFT                      F
*A*         STA     FLD25                                                 F
*A*         LDA     FLD91           VALUE                                 F
*A*FLD25    LSRA    0                                                     F
*A*         TAZ     FLD30           IF VALUE WITHIN BOUNDS                F
*A*         IMP     FLD65           RANGE ERROR                           F
*A*FLD30    LDA     2,X             LOW ORDER BIT POSITION                F
*A*         LSRA    4                                                     F
```

```
*A*          CPA
*A*          ANAI    3
*A*          ADD     F113          FETCH 16 BIT WORD ADDR
*A*          STA     FLD95
*A*          LDA     2,X
*A*          TAB
*A*          SUBI    31            CHECK IF IM FIELD
*A*          JAZ     FLD50         YES, SPECIAL PROCESSING
*A*          SUBI    15            CHECK IF FS FIELD
*A*          JAZ     FLD50A        YES, SPECIAL PROCESSING
*A*          TBA                   NEITHER IM NOR FD, RESTORE VALUE
*A*          ANAI    017           WORD DISPLACEMENT
*A*          TAB
*A*          LDA     FLD35
*A*          ANAI    07740
*A*          MERGE   031
*A*          STA     FLD35         FORM LEFT ROTATE
*A*          LDB     1,X           FIELD SIZE
*A*          LDAE    FLMSK,B       BIT MASK FOR FIELD SIZE
*A*FLD35     LRLA    0             MOVE MASK INTO POSITION
*A*          CPA
*A*          LDB     FLD95
*A*          ANA     0,B           MASK OFF BITS FROM WORD
*A*          TAB
*A*          LDA     FLD91         NEW FIELD VALUE
*A*          XEC     FLD35         MOVE VALUE INTO POSITION
*A*          MERGE   031           UPDATE WORD
*A*          LDB     FLD95         RAM WORD ADDR
*A*          STA     0,B
*A*FLD39     EQU     *
*A*          LDA     FLD93         TERMINATING CHAR
*A*          SUBI    ','
*A*          JAZ     FLD40         IF MORE CHARS IN RECORD
*A*          JMP     EXC10         GET NEW DIRECTIVE
*A*FLD40     EQU     *
*A*          JMPM    FETCHA        FETCH NEXT CHAR
*A*          TAB
*A*          SUBI    ' '
*A*          JAZ     F114A         IF DISPLAY NEXT WORD
*A*          TBA
*A*          JMP     FLD05         CONTINUE DIRECTIVE PROCESSING
*A*FLD50A    EQU     *             SPECIAL PROCESSING FOR FS FIELD
*A*          LDBI    2             SET FOR FS FIELD ROTATE
```

```
*A*        JMP     FLD50B                                                F
*A*FLD50   EQU     *              SPECIAL PROCESSING FOR IM FIELD        F
*A*        INCR    02             SET FOR IM FIELD ROTATE                F
*A*FLD50B  EQU     *                                                     F
*A*        STB     FLD51          FIELD FLAG                             F
*A*        TBA                                                           F
*A*        ADDI    16             SET UP LLRL INST                       F
*A*        ADD     FLLRL                                                 F
*A*        STA     FLD56          ROTATE INTO A REG LSBS                 F
*A*        LDAI    16                                                    F
*A*        SUBI    0              SET UP RESTORE ROTATE                  F
*A*FLD51   BES     0                                                     F
*A*        ADD     FLLRL          LLRL INST                              F
*A*        STA     FLD57                                                 F
*A*        LDX     FLD95          CCS WORD SUB ADDR                      F
*A*        LDA     0,X            LSB PORTION                            F
*A*        DXR                    POINT TO NEXT HIGH ORDER BITS          F
*A*        LDB     0,X            MSB PORTION                            F
*A*FLD56   LLRL    0              POSITION FIELD INTO A REG LSBS         F
*A*        ANAI    0177760        MASK OUT FIELD                         F
*A*        ADD     FLD91          CHANGE VALUE                           F
*A*FLD57   LLRL    0              RETURN FIELDS TO PROPER POSITION       F
*A*        STB     0,X                                                   F
*A*        STA     1,X                                                   F
*A*        JMP     FLD39          CONTINUE FIELD PROCESSING              F
*A**                                                                     F
*A*FLLRL   LLRL    0              LLRL INSTRUCTION BASE VALUE            F
*A*FLD90   DATA    0              FIELD NAME                             F
*A*FLD91   DATA    0              NEW FIELD VALUE                        F
*A*FLD92   DATA    0              NAME TABLE POINTER                     F
*A*FLD93   DATA    0              TERM CHAR                              F
*A*FLD94   DATA    0                                                     F
*A*FLD95   DATA    0              ROM WORD SUBCOMP ADDR                  F
*A**                                                                     F
*A*FLD60   CALL    SIOUT,3,FLER1    OUTPUT FIELD NAME ERROR              F
*A*        JMP     EXC10          TRY AGAIN                              F
*A*FLD65   CALL    SIOUT,3,FLER2    OUTPUT FIELD RANGE ERROR             F
*A*        JMP     EXC10          TRY AGAIN                              F
*A*FLER1   DATA    '  MS16'                                             F
*A*FLER2   DATA    '  MS17'                                             F
*A**                                                                     F
*A**  FIELD NAME TABLE                                                   F
*A**       3 WORD ENTRY                                                  F
```

```
*A**                1=2 CHAR NAME                                          F
*A**                2=FIELD SIZE                                           F
*A**                3=LOW ORDER BIT POSITION                               F
*A**                                                                       F
*A*FLNAM   EQU      *                                                      F
*A*        DATA     'AA',4,0                                               F
*A*        DATA     'BB',4,4                                               F
*A*        DATA     'SH',3,8                                               F
*A*        DATA     'XF',2,11                                              F
*A*        DATA     'WF',1,13                                              F
*A*        DATA     'VF',1,14                                              F
*A*        DATA     'SC',1,15                                              F
*A*        DATA     'WR',1,16                                              F
*A*        DATA     'CF',2,17                                              F
*A*        DATA     'MF',1,19                                              F
*A*        DATA     'FF',4,20                                              F
*A*        DATA     'RF',3,24                                              F
*A*        DATA     'LA',2,27                                              F
*A*        DATA     'LB',2,29                                              F
*A*        DATA     'IM',4,31                                              F
*A*        DATA     'AB',2,35                                              F
*A*        DATA     'MR',1,37                                              F
*A*        DATA     'GF',4,38                                              F
*A*        DATA     'SF',2,42                                              F
*A*        DATA     'TF',2,44                                              F
*A*        DATA     'FS',4,46                                              F
*A*        DATA     'MT',1,50                                              F
*A*        DATA     'MS',4,51                                              F
*A*        DATA     'AF',5,55                                              F
*A*        DATA     'TS',4,60                                              F
*A*        DATA     0               END OF TABLE                           F
*A**                                                                       F
*A*FLMSK   EQU      *               BIT MASK TABLE                         F
*A*        DATA     0                                                      F
*A*        DATA     01                                                     F
*A*        DATA     03                                                     F
*A*        DATA     07                                                     F
*A*        DATA     017                                                    F
*A*        DATA     037                                                    F
   .REPLACE,5425,5426           MICSIM    NO SMR    04/21/75  KERNS        E.2
*D*        LDA      FIL1,1                                                 D
*D*TRC     LDB      CPAG            CURRENT PAGE                           D
*A*TRC     LDB      PPAG            CURRENT PAGE AT START OF MICRO         E.2
```

```
*A*         LDAE    TRACE,2                                                    E,2
*A*         JAZ     TRE             IF TRACE IS OFF FOR THE CURRENT PAGE       E,2
*A*         LDA     FIL1,1          CURRENT CCS ADDR                           E,2
   .INSERT,5432                     MICSIM    NO SMR    04/21/75  KERNS       E,2
*A*TRE      DECR    01              SET FOR NO TRACE                           E,2
*A*         JMP*    TRSETA          RETURN                                     E,2
   .REPLACE,5710                    MICSIM    SMR=      04/07/75  KERNS       E,2
*D*LD5      STAT    LD1,LD3,LD3,LD3,LD2                                        C,1
*A*LD5      STAT    LD1,LD6,LD6,LD6,LD5                                        E,2
   .REPLACE,5748                    MICSIM    NO SMR    11/27/74  KERNS       FP
*D*         LDX     3,X             SO ASSIGNMENT                               C
*A*         LDA     3,Y             SO ASSIGNMENT                              FP
```

```
   .REPLACE,28                    DSIGN     SMR-739              KERNS     E.1
*D*30     CALL $ZN
*A* 30     CALL $ZC                                                        E.1
```

```
   .INSERT,141                                                          F.1
*A*        TFT      VORTEX-3                                           F.1
*A*        TFF      VORTEX-4                                           F.1
*A*        GOTO     DAFPP                                              F.1
   .REPLACE,157
*D*        CALL     TF,DATP,DAT3,DAT3,DAT1
*A*        CALL     TF,DATP,DAT3,DAT3,DAT1X
*A*DAT3X   FLDD
*A*        DATA     DAP2M
*A*        IMP      DAT2X
*A*DAT1X   FLDD
*A*        DATA     DAP2
*A*DAT2X   FSTD
*A*        DATA     DATS
   .REPLACE,165,168
*D*        FLDD                                                         V&1
*D*        DATA     DAP2            PI/2                                V&1
*D*        FSTD                                                         V&1
*D*        DATA     DATS                                                V&1
   .REPLACE,233
*D*        CALL     TF,DATP,DAT1,DAT2,DAT2
*A*        CALL     TF,DATP,DAT3X,DAT2,DAT2
   .INSERT,242
*A*DAP2M   DATA     0201,0115570,073250,042050    -PI/2
   .INSERT,296                                                          F.1
*A*DAFPP   CONT                                                         F.1
*A*        TFT      VORTEX-1                                           F.1
*A*        TFF      VORTEX-2                                           F.1
*A*        GOTO     DASDET                                             F.1
*A*        DATA     V                                                  F.1
*A*************************                                            F.1
*A** SET FINAL CONDITIONS *                                             F.1
*A*************************                                            F.1
*A*        TZB                      INIT SIGN FLAG                     F.1
*A*        LDA      V+1                                                F.1
*A*        JAP      DAT010                                             F.1
*A*        IBR                      (SET NEGATIVE)                     F.1
*A*        CPA                      FORCE POSITIVE                     F.1
*A*        STA      V+1                                                F.1
*A*DAT010  FLDD                     COMPARE TO 1.0                     F.1
*A*        DATA     ONE                                                F.1
*A*        FSBD                                                        F.1
*A*        DATA     V                                                  F.1
```

```
*A*          FSTO                                                    F.1
*A*          DATA     TEMP                                           F.1
*A*          LDXI     PTOVR2         TNTT CONSTANT DIFFERENCE        F.1
*A*          LDA      TEMP+1         Y .GT. 1?                       F.1
*A*          TAP      DAT030         NO-  GO SET CONSTANTS           F.1
*A*          FLDD                    TNVERT X                        F.1
*A*          DATA     ONE                                            F.1
*A*          FDVD                                                    F.1
*A*          DATA     V                                              F.1
*A*          FSTO                                                    F.1
*A*          DATA     V                                              F.1
*A*          IDAI     ONE            TNTT SIGN OF RESULT             F.1
*A*          JBNZ     DAT020         (USE +)                         F.1
*A*DAT015 IDAI     MONE           (USE -)                         F.1
*A*DAT020 STA      DBTH21                                         F.1
*A*          STY      DBTH18                                         F.1
*A*********************************************************          F.1
*A** SELECT THE APPROPRIATE TAN-1(U) (BY EIGHTHS) *                   F.1
*A*********************************************************          F.1
*A*          LDA      V                                              F.1
*A*          TZR                                                     F.1
*A*          SUBI     0175                                           F.1
*A*          TAN      DBTH11                                         F.1
*A*          TBR                                                     F.1
*A*          TAZ      DBTH11                                         F.1
*A*          SUBI     0200-0175                                      F.1
*A*          LDR      V+1                                            F.1
*A*          ASRR     11                                             F.1
*A*          TAZ      DBTH09         NO SHIFT OF R                   F.1
*A*          TAP      DBTH01         FORCE EIGHT                     F.1
*A*          TAP                                                     F.1
*A*          TAZ      DBTH06         SHIFT R ONCE                    F.1
*A*          TAR                                                     F.1
*A*          TAZ      DBTH03         SHIFT R TWICE                   F.1
*A*          ASRR     1                                              F.1
*A*          JMP      DBTH09                                         F.1
*A*DBTH01 LDRI     R              USE R                           F.1
*A*          JMP      DBTH11                                         F.1
*A*DBTH03 ASRR     1                                              F.1
*A*DBTH06 ASRR     1                                              F.1
*A*DBTH09 THA                                                     F.1
*A*          TZR                                                     F.1
*A*          LASR     1                                              F.1
```

```
*A*        JBZ     D8TH10                                                E.1
*A*        TAR                                                           E.1
*A*D8TH10  TAB                                                           E.1
*A*D8TH11  LDAE    DTANU,2                                               E.1
*A*        STA     D8TH12          SET ARCTAN ADDR                       E.1
*A*        ASLB    1               FORCE V TO NEAREST EIGHTH             E.1
*A*        LDAE    V8TH,2                                                E.1
*A*        STA     U                                                     E.1
*A*        LDAE    V8TH+1,2                                              E.1
*A*        STA     U+1                                                   E.1
*A*************************                                              E.1
*A** TAN-1((U-V)/(1+U*V)) *                                              E.1
*A*************************                                              E.1
*A*        FLDD                                                          E.1
*A*        DATA    V                                                     E.1
*A*        FMUD                                                          E.1
*A*        DATA    U                                                     E.1
*A*        FADD                                                          E.1
*A*        DATA    ONE                                                   E.1
*A*        FSTD                                                          E.1
*A*        DATA    TEMP                                                  E.1
*A*        FLDD                                                          E.1
*A*        DATA    V                                                     E.1
*A*        FSBD                                                          E.1
*A*        DATA    U                                                     E.1
*A*        FDVD                                                          E.1
*A*        DATA    TEMP                                                  E.1
*A*        FSTD                                                          E.1
*A*        DATA    Z                                                     E.1
*A*******************                                                    E.1
*A** CALCULATE Z**2 *                                                    E.1
*A*******************                                                    E.1
*A*        FLDD                                                          E.1
*A*        DATA    Z                                                     E.1
*A*        FMUD                                                          E.1
*A*        DATA    Z                                                     E.1
*A*        FSTD                                                          E.1
*A*        DATA    R                                                     E.1
*A*********************                                                  E.1
*A** CALCULATE ARCTAN *                                                  E.1
*A*********************                                                  E.1
*A*        FLDD                                                          E.1
*A*        DATA    A4                                                    E.1
```

```
*A*          FMUD                                                    E,1
*A*          DATA    R                                               E,1
*A*          FADD                                                    E,1
*A*          DATA    A3                                              E,1
*A*          FMUD                                                    E,1
*A*          DATA    R                                               E,1
*A*          FADD                                                    E,1
*A*          DATA    A2                                              E,1
*A*          FMUD                                                    E,1
*A*          DATA    R                                               E,1
*A*          FADD                                                    E,1
*A*          DATA    A1                                              E,1
*A*          FMUD                                                    E,1
*A*          DATA    R                                               E,1
*A*          FADD                                                    E,1
*A*          DATA    A0                                              E,1
*A*          FMUD                                                    E,1
*A*          DATA    Z                                               E,1
*A*          FADD                                                    E,1
*A*DRTH12    RSS     1                                               E,1
*A*          FSBD                                                    E,1
*A*DRTH18    RSS     1                                               E,1
*A*          FMUD                                                    E,1
*A*DRTH21    DATA    1                                               E,1
*A*          FSTD                                                    E,1
*A*          DATA    AC                                              E,1
*A*          LDA                                                     E,1
*A*          LDB                                                     E,1
*A*          LDX                                                     E,1
*A*          JMP*    $DAT                                            E,1
*A*DAT030    LDXI    ZERO            NO CONSTANT ADDED               E,1
*A*          LDAI    ONE                                             E,1
*A*          JBZ     DAT020          NO SIGN CHANGE                  E,1
*A*          JMP     DAT015          USE -1 TO CONVERT SIGN          E,1
*A*DTANU     DATA    DT00,DT18,DT14,DT38,DT12,DT58,DT34,DT78,DT1     E,1
*A*DT1       DATA    0200,062207,073250,042055       .7853981633974483096D0  E,1
*A*DT78      DATA    0200,056002,047612,074106       .7188299996215245056D0  E,1
*A*DT34      DATA    0200,051136,017506,023237       .6435011087932843866D0  E,1
*A*DT58      DATA    0200,043600,013527,057377       .5585993153435624366D0  E,1
*A*DT12      DATA    0177,073261,047012,060673       .4636476090008081166D0  E,1
*A*DT38      DATA    0177,055730,031203,062337       .3587706702705722206D0  E,1
*A*DT14      DATA    0176,076555,065762,026201       .2449786631268641546D0  E,1
*A*DT18      DATA    0175,077526,072465,026057       .1243549945467614356D0  E,1
```

```
*A*ZERO    BSS     0                                                    E.1
*A*DT00    DATA    0,0,0,0                                  .0          E.1
*A*A0      DATA    0201,040000,0,0                     .99999999999999840D0  E.1
*A*A1      DATA    0177,0125252,025252,052305         -.33333333333128423 2D0  E.1
*A*A2      DATA    0176,063146,031441,012502           .19999999580115327 6D0  E.1
*A*A3      DATA    0176,0133333,013232,07433          -.14285413038870496 4D0  E.1
*A*A4      DATA    0175,070337,051326,036035           .11022814624822128 6D0  E.1
*A*VRTH    DATA    0,0                                 0                   E.1
*A*        DATA    0176,040000                         1/8                 E.1
*A*        DATA    0177,040000                         2/8                 E.1
*A*        DATA    0177,060000                         3/8                 E.1
*A*        DATA    0200,040000                         4/8                 E.1
*A*        DATA    0200,050000                         5/8                 E.1
*A*        DATA    0200,060000                         6/8                 E.1
*A*        DATA    0200,070000                         7/8                 E.1
*A*ONE     DATA    0201,040000,0,0                     1                   E.1
*A*MONE    DATA    0201,0137777,0,0                     -1.0               E.1
*A*TEMP    BSS     4                TEMPORARY STORAGE                      E.1
*A*PIOVR2  DATA    0201,062207,073250,042055            1.570796326794     E.1
*A*U       BSS     4                                                       E.1
*A*V       BSS     4                                                       E.1
*A*R       BSS     4                                                       E.1
*A*Z       BSS     4                                                       E.1
*A*DASOFT  CONT                                                            E.1
```

```
    .REPLACE,447,454                                                    E,1
*D*DEA2     DATA    0206            16-BIT 28
*D*         DATA    034000
*D*         DATA    0
*D*         DATA    0
*D*DEB1     DATA    0217            16-BIT 15,120
*D*         DATA    035420
*D*         DATA    0
*D*         DATA    0
*A*DEA2     DATA    0205,070000,0,0         16-BIT 28               E,1
*A*DEB1     DATA    0216,073040,0,0         16-BIT 15,120           E,1
```

```
    .INSERT,123                                                    E.1
*A*FLT     MAC                                                     E.1
*A*        DATA    0105425                                         E.1
*A*        DATA    P(1)                                            E.1
*A*        FMAC                                                    E.1
*A*        SPAC    1                                               E.1
    .INSERT,195                                                    E.1
*A*        TFT     VORTEX=3                                        E.1
*A*        TFF     VORTEX=4                                        E.1
*A*        GOTO    DERM1                                           E.1
    .INSERT,197                                                    E.1
*A*DERM1   CONT                                                    E.1
    .INSERT,225                                                    E.1
*A*        TFT     VORTEX=1                                        E.1
*A*        TFF     VORTEX=2                                        E.1
*A*        GOTO    DERM1                                           E.1
*A*        FLT     DLN+1           CREATE N=1/2                    E.1
*A*        FADD                                                    E.1
*A*        DATA    DHALF                                           E.1
*A*DERM1   CONT                                                    E.1
    .INSERT,227                                                    E.1
*A*        TFT     VORTEX=3                                        E.1
*A*        TFF     VORTEX=4                                        E.1
*A*        GOTO    DERM2                                           E.1
    .INSERT,243                                                    E.1
*A*DERM2   CONT                                                    E.1
    .INSERT,282                                                    E.1
*A*        TFT     VORTEX=1                                        E.1
*A*        TFF     VORTEX=2                                        E.1
*A*        GOTO    DERM1                                           E.1
*A** EVALUATE POLY (NO LOOP FOR FPP)                               E.1
*A*        FMUD                                                    E.1
*A*        DATA    DL7                                             E.1
*A*        FADD                                                    E.1
*A*        DATA    A5                                              E.1
*A*        FMUD                                                    E.1
*A*        DATA    DL7                                             E.1
*A*        FADD                                                    E.1
*A*        DATA    A4                                              E.1
*A*        FMUD                                                    E.1
*A*        DATA    DL7                                             E.1
*A*        FADD                                                    E.1
*A*        DATA    A3                                              E.1
```

```
*A*          FMUD                                                    E.1
*A*          DATA    DL7                                             E.1
*A*          FADD                                                    E.1
*A*          DATA    A2                                              E.1
*A*          FMUD                                                    E.1
*A*          DATA    DL7                                             E.1
*A*          FADD                                                    E.1
*A*          DATA    A1                                              E.1
*A*          FMUD                                                    E.1
*A*          DATA    DL7                                             E.1
*A*          FADD                                                    E.1
*A*          DATA    A0                                              E.1
*A*          GOTO    DFRM2                                           E.1
*A*DFRM1     CONT                                                    E.1
    .INSERT,302                                                      E.1
*A*DFRM2     CONT                                                    E.1
    .INSERT,373                                                      E.1
*A*A5        EQU     *                                               E.1
    .INSERT,377                                                      E.1
*A*A4        EQU     *                                               E.1
    .INSERT,381                                                      E.1
*A*A3        EQU     *                                               E.1
    .INSERT,385                                                      E.1
*A*A2        EQU     *                                               E.1
    .INSERT,389                                                      E.1
*A*A1        EQU     *                                               E.1
    .INSERT,393                                                      E.1
*A*A0        EQU     *                                               E.1
    .INSERT,397                                                      E.1
*A*DHALF     DATA    0200,040000,0,0                                 E.1
```

```
    .TNSERT,29                                                          E.1
*A*         NAME    ASC             WORDO                               E.1
*A*         NAME    $CAC                                                E.2
*A*         NAME    $CACT                                               E.2
*A*ASC      FOU     *                                                   E.1
    .TNSERT,34                                                          E.2
*A*$CAC     DATA    0,0                                                 E.2
*A*$CACT    DATA    0,0
```

```
   .REPLACE,192                                                          E.1
*D*        LDXE     *=4          FETCH ADDR OF ARGUMENT                  D.1
*A*        LDXE     *=5          FETCH ADDR OF ARG                       E.1
```

```
    .REPLACE,25                         EXP       SHR-741   12/06/74  NEWDORF  E.1
*D**                        DATA REAL ARGUMENT R                                 *
*A**                    DATA -- ADDRESS OF REAL ARG R                            E.1
    .REPLACE,173                                                                 E.1
*D*         IDXE    *-4             FETCH ADDR OF ARGUMENT                       D.1
*A*         IDXE    *-5             FETCH ADDR OF ARG                            E.1
```

```
    .INSERT,27                          XDCOMP    FORT G              MCMURRAY  E.2
*A*         NAME    $60                                                       E.2
    .INSERT,30                          XDCOMP    FORT G              MCMURRAY  E.2
*A*$60      EQU     XDCO                                                      E.2
```

```
   .INSERT,147                                                          E.1
*A** FLOATING TO FIX                                                    E.1
*A*FIX     MAC                                                          E.1
*A*        DATA    0105621         FPP FIX                              E.1
*A*        DATA    P(1)                                                 E.1
*A*        EMAC                                                         E.1
*A*        SPAC    1                                                    E.1
*A** FIX TO FLOATING                                                    E.1
*A*FLT     MAC                                                          E.1
*A*        DATA    0105425         FPP FLOAT                            E.1
*A*        DATA    P(1)                                                 E.1
*A*        EMAC                                                         E.1
*A*        SPAC    1                                                    E.1
   .REPLACE,180                                                         E.1
*D*        STA     $DC0            LOAD EXIT
   .REPLACE,193,194                                                     E.1
*D*DC01    LDAI    $DC0+0100000                                         V&1
*D*        STA     $DSI            MOVE RETURN
*A*        LDAI    $DC0+0100000                                         E.1
*A*DC01    STA     $DSI            MOVE RETURN                          E.1
   .INSERT,208                                                          E.1
*A*        TFT     VORTEX-3                                             E.1
*A*        TFF     VORTEX-4                                             E.1
*A*        GOTO    DSCFPP                                               E.1
   .INSERT,566                                                          E.1
*A*DSCFPP CONT                                                          E.1
*A*        TFT     VORTEX-1                                             E.1
*A*        TFF     VORTEX-2                                             E.1
*A*        GOTO    DSCFPP                                               E.1
*A*        STR     DSSR                                                 E.1
*A*        STX     DSSX                                                 E.1
*A*************************************************************        E.1
*A** COMPUTES SIN/COS BY FORCING X TO MODULAR PI/2 AND *                 E.1
*A** PERFORMING 7 ITERATIONS.                          *                 E.1
*A*************************************************************        E.1
*A**                                A=1 FOR COS, 0 FOR SIN               E.1
*A*DS1     FSTO                     SAVE ARGUMENT                        E.1
*A*        DATA    XVAL                                                 E.1
*A*        STA     FLAG             SAVE SIN FLAG                       E.1
*A*        LDA     XVAL+1           CONVERT NEGATIVE TO PLU             E.1
*A*        JAP     SC030                                                E.1
*A*        CPA                                                          E.1
*A*        STA     XVAL+1                                               E.1
```

```
*A*         LDA     FLAG            IS IT SIN                       E.1
*A*         OAR                                                     E.1
*A*         JAZ     SC030                                           E.1
*A*         INR     FLAG            YES, SET FLAG TO TWO            E.1
*A*         INR     FLAG                                            E.1
*A*****************************************                         E.1
*A*** INITIALLY FORCE X TO MODULAR PI/2 *                           E.1
*A*****************************************                         E.1
*A*SC030    FLDD                                                    E.1
*A*         DATA    XVAL                                            E.1
*A*         FDVD                    X/(PI/2)                        E.1
*A*         DATA    PIOVR2                                          E.1
*A*         FSTD                                                    E.1
*A*         DATA    XSAR                                            E.1
*A*         FLDD                                                    E.1
*A*         DATA    XSAR                                            E.1
*A*         FIX     MOD             FORCE INTEGER MODULAR           E.1
*A*         LDA     MOD                                             E.1
*A*         CPA                                                     E.1
*A*         TAR                                                     E.1
*A*         STA     NEWX                                            E.1
*A*         FLT     NEWX            SEPARATE MANTISSA AND EXP       E.1
*A*         FADD                                                    E.1
*A*         DATA    XSAR                                            E.1
*A*         FSTD                                                    E.1
*A*         DATA    NEWX                                            E.1
*A*********************                                            E.1
*A*** SELECT OCTANT *                                               E.1
*A*********************                                            E.1
*A*         LDA     MOD             ONLY LEAST TWO BITS IMP         E.1
*A*         ADD     FLAG                                            E.1
*A*         STA     FLAG                                            E.1
*A*         INCR    1                                               E.1
*A*         STA     MOD                                             E.1
*A*         LDA     NEWX                                            E.1
*A*         SUB     H200            IS NEW X .GE. 1/2               E.1
*A*         JAN     SC080                                           E.1
*A*****************************************************            E.1
*A*** INVERT ARGUMENT AND FUNCTION IF REQUESTED *                    E.1
*A*****************************************************            E.1
*A*         TZA                     YES, INVERT ARGUMENT            E.1
*A*         STA     MOD                                             E.1
*A*         FLDD                                                    E.1
```

```
*A*        DATA    DONE                                                  E,1
*A*        FSBD                                                          E,1
*A*        DATA    NEWX                                                  E,1
*A*        FSTD                                                          E,1
*A*        DATA    NEWX                                                  E,1
*A**                          FIRST CALCULATE X**2                       E,1
*A*SC080   FLDD                                                          E,1
*A*        DATA    NEWX                                                  E,1
*A*        FMUD                                                          E,1
*A*        DATA    NEWX                                                  E,1
*A*        FSTD                                                          E,1
*A*        DATA    XSQR                                                  E,1
*A*        INCR    1                                                     E,1
*A*        ANA     FLAG                                                  E,1
*A*        SUB     MOD                                                   E,1
*A*        STA     MOD                                                   E,1
*A*        JANZ    SC090                                                 E,1
*A*********************************                                       E,1
*A** PERFORM COSINE ITERATIONS *                                          E,1
*A*********************************                                       E,1
*A*        FLDD                                                          E,1
*A*        DATA    XSQR                                                  E,1
*A*        FMUD                                                          E,1
*A*        DATA    COS7                                                  E,1
*A*        FADD                                                          E,1
*A*        DATA    COS6                                                  E,1
*A*        FMUD                                                          E,1
*A*        DATA    XSQR                                                  E,1
*A*        FADD                                                          E,1
*A*        DATA    COS5                                                  E,1
*A*        FMUD                                                          E,1
*A*        DATA    XSQR                                                  E,1
*A*        FADD                                                          E,1
*A*        DATA    COS4                                                  E,1
*A*        FMUD                                                          E,1
*A*        DATA    XSQR                                                  E,1
*A*        FADD                                                          E,1
*A*        DATA    COS3                                                  E,1
*A*        FMUD                                                          E,1
*A*        DATA    XSQR                                                  E,1
*A*        FADD                                                          E,1
*A*        DATA    COS2                                                  E,1
*A*        FMUD                                                          E,1
```

```
*A*        DATA    XSQR                                                E.1
*A*        FADD                                                        E.1
*A*        DATA    COS1                                                E.1
*A*        JMP     SC100                                               E.1
*A*****************************                                        E.1
*A** PERFORM SIN ITERATIONS *                                          E.1
*A*****************************                                        E.1
*A*SC090   FLDD                                                        E.1
*A*        DATA    XSQR                                                E.1
*A*        FMUD                                                        E.1
*A*        DATA    SIN7                                                E.1
*A*        FADD                                                        E.1
*A*        DATA    SIN6                                                E.1
*A*        FMUD                                                        E.1
*A*        DATA    XSQR                                                E.1
*A*        FADD                                                        E.1
*A*        DATA    SIN5                                                E.1
*A*        FMUD                                                        E.1
*A*        DATA    XSQR                                                E.1
*A*        FADD                                                        E.1
*A*        DATA    SIN4                                                E.1
*A*        FMUD                                                        E.1
*A*        DATA    XSQR                                                E.1
*A*        FADD                                                        E.1
*A*        DATA    SIN3                                                E.1
*A*        FMUD                                                        E.1
*A*        DATA    XSQR                                                E.1
*A*        FADD                                                        E.1
*A*        DATA    SIN2                                                E.1
*A*        FMUD                                                        E.1
*A*        DATA    XSQR                                                E.1
*A*        FADD                                                        E.1
*A*        DATA    SIN1                                                E.1
*A*SC100   LDA     MOD                                                 E.1
*A*        JAZ     SC105                                               E.1
*A*        FMUD                    MULTIPLY BY MANTISSA IF SIN         E.1
*A*        DATA    NEWX                                                E.1
*A*SC105   FSTD                                                        E.1
*A*        DATA    AC                                                  E.1
*A*        LDXI    AC                                                  E.1
*A*        LDA     FLAG            INVERT SIGN IF REQUIRED             E.1
*A*        ANAI    2                                                   E.1
*A*        TAB                                                         E.1
```

```
*A*         LDA     1,1                                                 E,1
*A*         XBNZ    SC110                                               E,1
*A*         STA     1,1                                                 E,1
*A*         LDA     DSSA            RESTORE A                           E,1
*A*         LDB     DSSB            RESTORE B                           E,1
*A*         LDX     DSSX            RESTORE X                           E,1
*A*         JMP*    $DSI            EXIT                                E,1
*A*SC110    CPA                                                         E,1
*A*DSSA     BSS     1               INTITAL A                           E,1
*A*DSSB     BSS     1               INTITAL B                           E,1
*A*DSSX     BSS     1               INTITAL X                           E,1
*A*FLAG     BSS     1               1=INVERT RESULT SIGN                E,1
*A*MOD      BSS     1                                                   E,1
*A*XVAL     EQU     *                                                   E,1
*A*NEWX     BSS     4               NEW X VALUE                         E,1
*A*XSQR     BSS     4               X**2                                F,1
*A*H200     DATA    0200                                                F,1
*A*STN7     DATA    0150,074374,071032,050631                           F,1
*A*STN6     DATA    0156,0103477,07072,073571                           E,1
*A*STN5     DATA    0164,052036,02773,021770                            E,1
*A*STN4     DATA    0171,0131513,054634,027136                          E,1
*A*STN3     DATA    0175,050632,074316,072607                           E,1
*A*STN2     DATA    0200,0126521,074714,022676                          E,1
*A*PTOVR2   EQU     *                                                   E,1
*A*STN1     DATA    0201,052207,073250,042055                           E,1
*A*COS7     DATA    0153,076366,076021,06037                            E,1
*A*COS6     DATA    0161,0113115,034076,042163                          E,1
*A*COS5     DATA    0166,074175,017554,05015                            E,1
*A*COS4     DATA    0173,0125213,074373,037731                          E,1
*A*COS3     DATA    0177,040360,037020,015407                           E,1
*A*COS2     DATA    0201,0130413,074623,037105                          E,1
*A*DONE     EQU     *                                                   E,1
*A*COS1     DATA    0201,040000,0,0                                     E,1
*A*DSCEPP   CONT                                                        E,1
```

```
    .INSERT,30
*A*INC      SET     3
*A*         IFF     VORTEX-1
*A*INC      SET     INC+1
*A*         IFF     VORTEX-3
*A*INC      SET     INC+1
    .REPLACE,36
*D*         LDX     $DIT+4
*A*         LDX     $DIT+INC
```

```
   .REPLACE,5                    FPPWCS                          FRITCHOFF E.2
*D**                                        RELEASED   08/02/74
*A**                                                                      E.2
   .INSERT,18                    FPPWCS                          PARKER    E.2
*A*AONES  EQU      2                                                      E.2
   .REPLACE,67                   FPPWCS                          PARKER    E.2
*D*OLZE   EQU      7
   .INSERT,72                    FPPWCS                          PARKER    E.2
*A*ORL7   EQU      7                                                      E.2
   .INSERT,103                   FPPWCS                          PARKER    E.2
*A*ROVFL  EQU      4                                                      E.2
   .INSERT,108                   FPPWCS                          PARKER    E.2
*A*SOVFL  EQU      2                                                      E.2
   .INSERT,117                   FPPWCS                          PARKER    E.2
*A*S$OVF$ EQU      8                                                      E.2
   .REPLACE,234                  FPPWCS                          FRITCHOFF E.2
*D*                C16(R7),11(LIT),MKFFFC,14(ADD),13(POUT)
*A*                C16(RD),11(LIT),MKFFFC,14(ADD),13(POUT)                E.2
   .REPLACE,247,253              FPPWCS                          PARKER    E.2
*D**
*D**                ENTRY FOR $DO WITH INCREMENT = 1
*D**                   SET RF = 1, ENTER INTO DO LOOP
*D**
*D*        ORG     X'17
*D*$DOIN1 GEN      /N(VALUE),12(A$SPEC),22(AZERO),14(INCA),16(CRY1),
*D*                C17(GPROUT),24(RF),10(WAITMD),6(SPEC)
   .REPLACE,297                  FPPWCS                          FRITCHOFF E.2
*D*                C24(R7),12(A$SPEC),22(AONE)
*A*                C24(R9),12(A$SPEC),22(AONE)                            E.2
   .REPLACE,299                  FPPWCS                          FRITCHOFF E.2
*D*FORTS4 GEN      /N(FORTS7),10(IF$P),12(A$GPR),24(R7),14(INCA),
*A*FORTS4 GEN      /N(FORTS7),10(IF$P),12(A$GPR),24(R9),14(INCA),         E.2
   .REPLACE,302                  FPPWCS                          FRITCHOFF E.2
*D*FORTS1 GEN      /*,10(OF$ALU),12(A$GPR),24(R7),14(INCA),16(CRY1),
*A*FORTS1 GEN      /*,10(OF$ALU),12(A$GPR),24(R9),14(INCA),16(CRY1),     E.2
   .REPLACE,317                  FPPWCS                          FRITCHOFF E.2
*D*                C12(A$GPR),24(R7)
*A*                C12(A$GPR),24(R9)                                      E.2
   .REPLACE,420                  FPPWCS                          FRITCHOFF E.2
*D*FORTS7 GEN      /P(SS3M),10(IF$P),12(A$GPR),24(R7),14(TRNA),
*A*FORTS7 GEN      /P(SS3M),10(IF$P),12(A$GPR),24(R9),14(TRNA),           E.2
   .REPLACE,514                  FPPWCS                          FRITCHOFF E.2
*D*                C11(B$SPEC),17(GPROUT),24(R7),        PTR IN 'P', TEST FOR
```

```
*A*              C11(B$SPEC),17(GPROUT),24(RD),     PTR IN 'P', TEST FORE,2
   ,REPLACE,589                    FPPWCS                          FRITCHOFF E,2
*D*MCTOS  GEN     /N(MCTOSX),10(OS$ALU),6(MEMC),24(R7),13(POUT)
*A*MCTOS  GEN     /N(MCTOSX),10(OS$ALU),6(MEMC),24(RD),13(POUT)          E,2
   ,REPLACE,650                    FPPWCS                          PARKER    E,2
*D*MRS20  GEN     /N(MRS21),10(OFSP),6(MEMC),11(B$SPEC),23(OLZF),
*A*MRS20  GEN     /N(MRS21),10(OFSP),6(MEMC),11(B$SPEC),23(ORLZ),        E,2
   ,REPLACE,696,698                FPPWCS                          FRITCHOFF E,2
*D**      RESTORE P FROM R7 AND RETURN
*D*MRS24  GEN     /N(SS3M),7(PJMPS),10(IFSALU),6(MEMCS),12(ASGPR),24(R7),
*D*              C14(INCA),18(CRY1),13(POUT)
*A**      RESTORE P FROM R9 AND RETURN                                   E,2
*A*MRS24  GEN     /P(SS2M),10(PJMP),12(ASGPR),24(R9),14(TRNA),15(LOG),   E,2
*A*              C13(POUT)                                              E,2
   ,REPLACE,798                    FPPWCS                          FRITCHOFF E,2
*D*MRS    GEN     /N(MRS1),12(ASP),14(TRNA),15(LOG),17(GPROUT),24(R7)
*A**      SAVE P IN R9                                                   E,2
*A*MRS    GEN     /N(MRS1),12(ASP),14(TRNA),15(LOG),17(GPROUT),24(R9)    E,2
   ,INSERT,800                     FPPWCS                          PARKER    E,2
*A*       EJEC                                                           E,2
*A*PAGE   EQU     X'200          PAGE NUMBER                             E,2
*A**                                                                     E,2
*A**      BCS ENTRY POINTS                                               E,2
*A**                                                                     E,2
*A**                                                                     E,2
*A**      ENTRY POINT FOR STACK-JUMPS AND JUMP IF >0.                    E,2
*A**      SAMPLES A REG FOR USE BY CONDITIONAL JUMP.                     E,2
*A**                                                                     E,2
*A*       ORG     X'15                                                   E,2
*A*       GEN     /F(STKJ1),FSA,2(1),12(ASGPR),24(R0),14(TRNA),15(LOG),  E,2
*A*              CTF0,SF0,7(S$ALU)                                      E,2
*A**                                                                     E,2
*A**      BCS ENTRY POINTS FOR RELATIONAL/LOGICAL ROUTINES               E,2
*A**      FLAG QUOS SET OR RESET TO FLAG WHICH ENTRY WAS TAKEN.          E,2
*A**                                                                     E,2
*A*       ORG     X'14                                                   E,2
*A*TFR1   GEN     /N(RFL1),12(ASSPEC),22(AZERO),14(TRNA),15(LOG),        E,2
*A*              CWF1,SC0                                               E,2
*A*       ORG     X'16                                                   E,2
*A*JTR1   GEN     /N(RFL1),12(ASSPEC),22(AONES),14(TRNA),15(LOG),        E,2
*A*              CWF1,SC0                                               E,2
*A**                                                                     E,2
*A**      BCS ENTRY - CHOOSE BETWEEN $001 AND MATH/SORT/INDEX.           E,2
```

```
*A**                                                                    E.2
*A*       ORG     X'17                                                  E.2
*A*       GEN     /F(SDOTN1),FSB,2(1)                                   E.2
*A**                                                                    E.2
*A**      RCS ENTRY POINT - DOUBLE WORD INTEGER ADD AND SUBTRACT        E.2
*A**      BEGIN FETCH OF SECOND OPERAND                                 E.2
*A**                                                                    E.2
*A*       ORG     X'1C                                                  E.2
*A*       GEN     /N(IAR1),10(OFSMTR),6(MEMC)                           E.2
*A*       FJFC                                                          E.2
*A**                                                                    E.2
*A**      MAIN BODY OF MICRO ROUTINES                                   E.2
*A**                                                                    E.2
*A*       ORG     X'F6                                                  E.2
*A**                                                                    E.2
*A**      FIELD SELECT ENTRY POINT FOR STACK JUMPS                      E.2
*A***     (THIS MICRO MUST BE ON AN EVEN MICRO ADDR)                    E.2
*A**                                                                    E.2
*A**                                                                    E.2
*A**      START FETCH OF STACK POINTER                                  E.2
*A**                                                                    E.2
*A*STKJ1  GEN      /N(STKJ2),10(OFSALU),6(MEMC),12(ASGPR),24(R1),       E.2
*A*                C14(TRNA),15(LUG)                                   E.2
*A**                                                                    E.2
*A**      JUMP IF A REGISTER GREATER THAN ZERO                          E.2
*A**                                                                    E.2
*A**      CALLING SEQUENCE:                                             E.2
*A**                                                                    E.2
*A**      DATA 0105125                                                  E.2
*A**      DATA (JUMP ADDR IF A>0)                                       E.2
*A**                                                                    E.2
*A***                                                                   E.2
*A***     TEST IF R0 WAS NEGATIVE; IF SO, A NOT >0 SO JUMP TO EXIT      E.2
*A***     (THIS MICRO MUST FOLLOW STKJ1)                                E.2
*A***                                                                   E.2
*A*JAG1   GEN      /T(JAG2,JAG6),5(FT),7(ALUS)                          E.2
*A**                                                                    E.2
*A**      TEST IF R0 WAS ZERO; IF SO, A NOT >0 SO JUMP TO EXIT          E.2
*A**                                                                    E.2
*A*JAG2   GEN      /T(JAG3,JAG6),5(FT),7(ALUZ)                          E.2
*A***                                                                   E.2
*A***     A REG IS >0 SO FETCH INSTRUCTION AT JUMP ADDR;                E.2
*A***     IF ADDR WAS IND, FETCH EFFECTIVE ADDR;                        E.2
```

```
*A***     (THIS MICRO MUST BE ON AN EVEN LOCATION)                                E.2
*A***                                                                             E.2
*A*       ORG     *+1                                                             E.2
*A*JAG3   GEN     /N(JAG4),10(TFSMIR),6(MEMC)                                     E.2
*A***                                                                             E.2
*A***     TEST IF ADDR WAS INDIRECT; IF SO, BACK TO FETCH AGAIN                   E.2
*A***     ALSO, SET NEW P VALUE.                                                  E.2
*A*       ORG     *-2                                                             E.2
*A*JAG4   GEN     /T(JAG5,JAG3),5(FT),7(MIRS),                                    E.2
*A*               C11(BRSPEC),23(MIR),14(TRNB),15(LOG),13(POUT)                   E.2
*A***                                                                             E.2
*A***     CONTINUE RESTART OF PIPELINE AT JUMP ADDR;                              E.2
*A***     EXIT TO ROM.                                                            E.2
*A***                                                                             E.2
*A*       ORG     *+1                                                             E.2
*A*JAG5   GEN     /P(SS3M),7(PJMPS),TF0,13(INCP),10(TFSP),6(MEMCS)                E.2
*A***                                                                             E.2
*A***     EXIT TO NEXT INSTRUCTION IN SEQUENCE; NO JUMP; A NOT > 0.               E.2
*A***     (THIS MICRO MUST BE EVEN)                                               E.2
*A***                                                                             E.2
*A*JAG6   GEN     /P(SS2M),7(PJMPS),TF0,13(INCP),10(TFSP),6(MEMCS)                E.2
*A*       FJEC                                                                    E.2
*A***                                                                             E.2
*A***     REENTRANT SUBROUTINE CALL/RETURN                                        E.2
*A***                                                                             E.2
*A***     RETURN ADDRESS IS SAVED ON STACK                                        E.2
*A***                                                                             E.2
*A***     R1 (B REG) CONTAINS ADDRESS OF STACK POINTER                            E.2
*A***                                                                             E.2
*A***     SAMPLE MICRO FLOW:                                                      E.2
*A***         DIR PSHJMP : STKJ1,STKJ2,STKJ3,PSHJ1,PSHJ2,PSHJ3                    E.2
*A***         IND PSHJMP : STKJ1,...,STKJ3,PSHJ1,PSHJ2,PSHJ4,PSHJ5,               E.2
*A***                      PSHJ6,PSHJ3                                            E.2
*A***         POPJMP     : STKJ1,STKJ2,STKJ3,POPJ1,POPJ2,POPJ3                    E.2
*A***                                                                             E.2
*A***     CALLING SEQUENCE:                                                       E.2
*A***                                                                             E.2
*A***         PSHJMP: DATA  0105025                                               E.2
*A***                 DATA  (JUMP ADDR)                                           E.2
*A***                                                                             E.2
*A***         POPJMP: DATA  0105065                                               E.2
*A***                                                                             E.2
*A***                                                                             E.2
```

```
*A**       SAVE CONTENTS OF LOC P+1 IN RF;                                    E.2
*A**       ALSO SAMPLE ITS SIGN AND INCR P REG.                               E.2
*A**                                                                          E.2
*A**STKJ2  GEN     /N(STKJ3),11(B$SPEC),23(MIR),14(TRNB),15(LOG),             E.2
*A*              C17(GPROUT),24(RF),7(S$ALU),13(INCP)                        E.2
*A**                                                                          E.2
*A**       INITIATE STORE OF NEW STACK POINTER;                               E.2
*A**       ALSO FIELD SELECT TO DECODE IF PSHJMP OR POPJMP.                   E.2
*A**                                                                          E.2
*A**STKJ3  GEN     /F(PSHJ1),FS0,2(1),10(DS$ALU),6(MEMC),                     E.2
*A*              C12(A$GPR),24(R1),14(TRNA),15(LOG)                          E.2
*A***                                                                         E.2
*A***      PROVIDE UPDATED STACK POINTER FOR STORE;                           E.2
*A***      INITIATE STORE OF RETURN ADDRESS ON STACK.                         E.2
*A***      (THIS MICRO ADDR MUST BE EVEN AND POPJ1 MUST FOLLOW PSHJ1)         E.2
*A***                                                                         E.2
*A*        ORG     *+1                                                        E.2
*A*PSHJ1   GEN     /N(PSHJ2),10(DS$ALU),6(MEMC),11(B$SPEC),23(MIR),           E.2
*A*              C14(DECR),12(A$SPEC),22(ADNFS)                              E.2
*A**                                                                          E.2
*A**       PROVIDE UPDATED STACK POINTER FOR STORE;                           E.2
*A**       INITIATE FETCH OF EXIT ADDRESS FROM STACK.                         E.2
*A**                                                                          E.2
*A*POPJ1   GEN     /N(POPJ2),10(DF$MIR),6(MEMC),11(B$SPEC),23(MIR),           E.2
*A*              C14(INCR),16(CRY1),12(A$SPEC),22(AZERO)                     E.2
*A**                                                                          E.2
*A**       WAIT FOR EXIT ADDRESS TO ARRIVE FROM MEMORY;                       E.2
*A**       RESTART PIPELINE AT EXIT ADDRESS.                                  E.2
*A**                                                                          E.2
*A*        ORG     *-3                                                        E.2
*A*POPJ2   GEN     /N(POPJ3),10(IF$MIR),6(MEMC)                               E.2
*A**                                                                          E.2
*A**       CONTINUE PIPELINE STARTUP;                                         E.2
*A**       EXIT BACK TO ROM TO DECODE NEXT INSTRUCTION.                       E.2
*A**                                                                          E.2
*A*        ORG     *+2                                                        E.2
*A*POPJ3   GEN     /P(SS3M),7(PJMPS),6(MEMCS),10(IF$ALU),11(B$SPEC),          E.2
*A*              C23(MIR),14(INCR),16(CRY1),12(A$SPEC),22(AZERO),            E.2
*A*              C13(POUT)                                                   E.2
*A**                                                                          E.2
*A**       PROVIDE RETURN ADDRESS FOR STORE ON STACK;                         E.2
*A**       CHECK IF JUMP ADDRESS IS INDIRECT;(NOTE, SIGN OF JUMP ADDR         E.2
*A**       WAS SAMPLED BACK AT STKJ2.)                                        E.2
```

```
*A**                                                                    E.2
*A*PSHJ2  GEN     /T(PSHJ3,PSHJ4),5(FT),7(ALUS),12(ASP),                E.2
*A*             C14(TRNA),15(LOG),6(SPEC),10(WAITMD)                    E.2
*A**                                                                    E.2
*A**      SET P TO JUMP ADDRESS;                                        E.2
*A**      RESTART PIPELINE; EXIT BACK TO ROM.                           E.2
*A**                                                                    E.2
*A*PSHJ3  GEN     /P(SS2M),7(PJMPS),6(MEMCS),10(IFSALU),13(POUT),       E.2
*A*             C12(ASGPR),24(RF),14(TRNA),15(LOG)                      E.2
*A***                                                                   E.2
*A***     FETCH EFFECTIVE JUMP ADDRESS USING INDIRECT POINTER;          E.2
*A***     (THIS MICRO ADDR MUST BE EVEN)                                E.2
*A***                                                                   E.2
*A*       ORG     **1                                                   E.2
*A*PSHJ4  GEN     /N(PSHJ5),10(DFSALU),6(MEMC),12(ASGPR),24(RF),        F.2
*A*             C14(TRNA),15(LOG)                                       E.2
*A**                                                                    E.2
*A**      WAIT FOR EFFECTIVE ADDRESS TO ARRIVE;                         E.2
*A**                                                                    E.2
*A*       ORG     *-2                                                   E.2
*A*PSHJ5  GEN     /N(PSHJ6),6(SPEC),10(WAITMD)                          E.2
*A**                                                                    E.2
*A**      CHECK IF ANOTHER LEVEL OF INDIRECT                            E.2
*A**                                                                    E.2
*A*       ORG     **1                                                   E.2
*A*PSHJ6  GEN     /T(PSHJ3,PSHJ4),5(FT),7(MIRS),11(BSSPEC),23(MIR),     E.2
*A*             C14(TRNB),15(LOG),17(GPROUT),24(RF)                     E.2
*A*       FJFC                                                          F.2
*A**                                                                    E.2
*A**      RELATIONAL/LOGICAL CONVERSION ROUTINES                        E.2
*A**                                                                    E.2
*A**      TWO SINGLE PRECISION FLOATING POINT NUMBERS ARE COMPARED. THE E.2
*A**      FOLLOWING OPTIONS ARE ALLOWED BY THESE ROUTINES:              F.2
*A**              1) JUMP IF SPECIFIED RELATION MET; (BCS X'14).        E.2
*A**              2) TEST FOR SPECIFIED RELATION; SET A TO ONES IF      E.2
*A**                 RELATION MET OR TO ZERO IF NOT; (BCS X'16).        E.2
*A**              3) THREE WAY BRANCH ON LT,EQ,AND GT;                  E.2
*A**                 (BCS X'14 OR X'16).                                E.2
*A**                                                                    F.2
*A**      PARAMETER FIELD OF BCS SPECIFIES RELATION AS FOLLOWS:         E.2
*A**              0 = NE                                                E.2
*A**              1 = EQ                                                E.2
*A**              2 = LT                                                E.2
```

```
*A**                3 - GE                                                  E.2
*A**                4 - 3 WAY BRANCH                                        F.2
*A**                5 - 3 WAY BRANCH                                        E.2
*A**                6 - LE                                                  E.2
*A**                7 - GT                                                  F.2
*A**                                                                        E.2
*A**      REGISTER USAGE:                                                   E.2
*A**                R8 - ADDR OF FIRST OPERAND                              E.2
*A**                R9 - FIRST WORD OF FIRST OPERAND                        F.2
*A**                RA - SECOND WORD OF FIRST OPERAND                       E.2
*A**                RE - DIFFERENCE BETWEEN OPERANDS; ALSO SET TO FLAG RFLE.2
*A**                RF - ADDR OF SECOND OPERAND                             F.2
*A**                MIR- FIRST/SECOND WORD OF SECOND OPERAND                E.2
*A**                OPR- DIFFERENCE BETWEEN OPERAND SECOND WORDS.           F.2
*A**                OUOS - IF ONE, THEN JUMP IF RELATION MET; SET AT ENTRYE.2
*A**                                                                        F.2
*A**      CAUTION: THIS ROUTINE MAY NOT BE USED TO COMPARE DOUBLE-WORD      F.2
*A**                FIXED POINT NUMBERS SINCE FORMATS OF NEGATIVE NUMBERS   E.2
*A**                ARE NOT CONSISTENT.                                     E.2
*A**                                                                        E.2
*A**                                                                        E.2
*A**      BEGIN FETCH OF WORD ONE, OPERAND ONE (OR IND ADDR)                F.2
*A**                                                                        F.2
*A*RFL1   GEN       /N(RFL2),10(OFSMIR),6(MEMC)                             E.2
*A**                                                                        E.2
*A**      IF ADDRESS WAS INDIRECT, LOOP BACK TO FETCH FINAL ADDR;           F.2
*A**      SAVE FINAL ADDR;                                                  E.2
*A**                                                                        E.2
*A*RFL2   GEN       /T(RFL1,REL3),5(TT),7(MIRS),11(BSSPEC),23(MIR),         F.2
*A*                C14(TRNR),15(LOG),17(GPROUT),24(R8)                     F.2
*A**                                                                        E.2
*A***     INITIATE FETCH OF ADDR OF OPERAND TWO                             E.2
*A***     (THIS MICRO ADDR MUST BE EVEN)                                    F.2
*A***                                                                       F.2
*A*RFL3   GEN       /N(RFL4),13(INCP),10(OFSP),6(MEMC)                      E.2
*A**                                                                        F.2
*A**      SAVE WORD ONE, OPERAND ONE                                        F.2
*A**                                                                        E.2
*A*RFL4   GEN       /N(RFL5),11(BSSPEC),23(MIR),14(TRNR),15(LOG),           F.2
*A*                C17(GPROUT),24(R9)                                      F.2
*A**                                                                        F.2
*A**      BEGIN FETCH OF WORD ONE, OPERAND TWO (OR IND ADDR).               E.2
*A**                                                                        F.2
```

```
*A*REL5   GEN     /N(REL6),10(OPSMTR),6(MEMC)                          E.2
*A**                                                                   E.2
*A**      IF ADDR WAS INDIRECT, LOOP BACK TO FETCH FINAL ADDR;         E.2
*A**      SAVE FINAL ADDR;                                             E.2
*A**                                                                   E.2
*A*REL6   GEN     /T(REL5,REL7),5(TT),7(MIRS),11(BSSPEC),23(MIR),      E.2
*A*             C14(TRNB),15(LOG),17(GPROUT),24(RF)                  E.2
*A***                                                                  E.2
*A***     INITIATE FETCH OF SECOND WORD, OPERAND ONE;                  E.2
*A***     (THIS MICRO ADDR MUST BE EVEN).                              E.2
*A***                                                                  E.2
*A*REL7   GEN     /N(REL8),10(OPSALU),6(MEMC),12(ASGPR),24(R8),        E.2
*A*             C14(INCA),16(CRY1),7(ROVFL)                          E.2
*A**                                                                   E.2
*A**      COMPARE FIRST TWO WORDS OF OPERANDS                          E.2
*A**                                                                   E.2
*A*REL8   GEN     /N(REL9),12(ASGPR),24(R9),11(BSSPEC),23(MIR),        E.2
*A*             C14(SUB),16(CRY1),7(SSOVFS+SSALU),TF0,SF0            E.2
*A**                                                                   E.2
*A**      IF OVERFLOW OCCURED, JUMP TO MAKE LT/GT DECISION             E.2
*A**                                                                   E.2
*A*REL9   GEN     /T(REL10,REL20),5(FT),7(OVFL)                        E.2
*A**                                                                   E.2
*A**      IF SIGN SET, THEN JUMP TO LT STATE                           E.2
*A**                                                                   E.2
*A*REL10  GEN     /T(REL11,RLT),5(FT),7(ALUS)                          E.2
*A**                                                                   E.2
*A**      IF FIRST WORDS OF OPERANDS ARE EQUAL, BEGIN FETCH OF OP2 SECONE.2
*A**      WORD; ELSE JUMP TO GT STATE.                                 E.2
*A**                                                                   E.2
*A*REL11  GEN     /T(REL12,RGT),5(TT),7(ALUZ),10(OPSALU),6(TESTT),     E.2
*A*             C12(ASGPR),24(RF),14(INCA),16(CRY1)                  E.2
*A**                                                                   E.2
*A**      SAVE SECOND WORD OF OPERAND ONE                              E.2
*A**                                                                   E.2
*A*REL12  GEN     /N(REL13),11(BSSPEC),23(MIR),14(TRNB),15(LOG),       E.2
*A*             C17(GPROUT),24(RA)                                   E.2
*A**                                                                   E.2
*A**      WAIT FOR SECOND WORD OF OPERAND TWO                          E.2
*A**                                                                   E.2
*A*REL13  GEN     /N(REL14),6(SPEC),10(WAITMD)                         E.2
*A**                                                                   E.2
*A**      COMPARE SECOND WORDS OF OPERANDS;                            E.2
```

```
*A**      NOTE: OVERFLOW SHOULD NOT BE POSSIBLE DUE TO SNGL                    E.2
*A**      PRECISION FP FORMAT.                                                 E.2
*A**                                                                           E.2
*A*RFL14  GEN      /N(RFL15),12(ASGPR),24(RA),11(RSSPEC),23(MIR),             E.2
*A*              C14(SUB),16(CRY1),7(SSOVFS+SSALU),TFO,SFO,13(OPROUT)         E.2
*A**                                                                          E.2
*A**      IF WORDS EQUAL, JUMP TO EQ STATE                                     E.2
*A**                                                                           E.2
*A*RFL15  GEN      /T(REQ,RFL16),5(TT),7(ALUZ)                                E.2
*A***                                                                          E.2
*A***     TOGGLE SIGN IF OPERANDS (FIRST WORD) ARE NEGATIVE                    E.2
*A***     (THIS MICRO ADDR MUST BE EVEN)                                       E.2
*A***                                                                          E.2
*A*       ORG      **1              PUT ON EVEN WORD                          E.2
*A*RFL16  GEN      /N(RFL17),12(ASGPR),24(R9),11(RSSPEC),23(OPR),             E.2
*A*              C14(EOR),15(LOG),TFO,SFO,7(SSALU)                            E.2
*A**                                                                          E.2
*A**      IF SIGN SET, JUMP TO LT STATE; ELSE, GT STATE.                       E.2
*A**                                                                           E.2
*A*RFL17  GEN      /T(RLT,RGT),5(TT),7(ALUS)                                  E.2
*A***                                                                          E.2
*A***     MAKE LT/GT DECISION AFTER OVERFLOW                                   E.2
*A***     (THIS MICRO ADDR MUST BE EVEN)                                       E.2
*A***                                                                          E.2
*A*RFL20  GEN      /T(RGT,RLT),5(TT),7(ALUS)                                  E.2
*A***                                                                          E.2
*A***     SIGNAL LT,EQ, OR GT BY LOADING RF WITH -1,0,OR 1.                    E.2
*A***     BRANCH TO RELATIONAL CHECK USING PARAMETER FIELD OF RCS              E.2
*A***     (THIS MICRO ADDR MUST BE EVEN)                                       E.2
*A***                                                                          E.2
*A*       ORG      **1                                                        E.2
*A*RLT    GEN      /F(TJNF),FS9,2(7),12(ASSPEC),22(AZERO),                    E.2
*A*              C14(DECA),17(GPROUT),24(RE),TFO,SFO,7(SSALU),13(INCP)        E.2
*A*RFO    GEN      /F(TJNF),FS9,2(7),12(ASSPEC),22(AZERO),                    E.2
*A*              C14(TRNA),15(LOG),17(GPROUT),24(RF),TFO,SFO,                 E.2
*A*              C7(SSALU),13(INCP)                                           E.2
*A*RGT    GEN      /F(TJNE),FS9,2(7),12(ASSPEC),22(AZERO),                    E.2
*A*              C14(INCA),16(CRY1),17(GPROUT),24(RE),TFO,SFO,                E.2
*A*              C7(SSALU),13(INCP)                                           E.2
*A**                                                                          E.2
*A**      RELATIONAL TESTS                                                     E.2
*A**      THESE MICROS TEST FOR ONE OF THE 6 RELATIONS OR                      E.2
*A**      BEGIN PROCESSING THE THREE WAY BRANCH (AIF).                         E.2
```

```
*A**      THESE MICROS MUST BE ON AN 8 WORD BOUNDARY.                          E.2
*A***                                                                          E.2
*A*       ORG     **1             PUT ON EVEN WORD                             E.2
*A*TJNE   GEN      /T(RMET,RNOT),5(FT),7(ALUZ)                                 E.2
*A*TJEQ   GEN      /T(RMET,RNOT),5(TT),7(ALUZ)                                 E.2
*A*TJLT   GEN      /T(RMET,RNOT),5(TT),7(ALUS)                                 E.2
*A*TJGE   GEN      /T(RMET,RNOT),5(FT),7(ALUS)                                 E.2
*A*ATF1   GEN      /N(ATF2),10(DFSALU),6(MEMC),12(ASP),11(BSGPR),23(RE),       E.2
*A*              C14(ADD),16(CRY1)    *** FETCH 3-WAY JUMP ADDR ***           E.2
*A*       GEN      /N(ATF2),10(DFSALU),6(MEMC),12(ASP),11(BSGPR),23(RE),       E.2
*A*              C14(ADD),16(CRY1)    *** FETCH 3-WAY JUMP ADDR ***           E.2
*A*TJLE   GEN      /N(TJNE),12(ASGPR),24(RE),14(DECA),7(SSALU),TF0,SF0         E.2
*A*TJGT   GEN      /N(TJEQ),12(ASGPR),24(RE),14(DECA),7(SSALU),TF0,SF0         E.2
*A***                                                                          E.2
*A**      FLAG RELATION TRUE; TEST QUOS TO DECIDE IF JUMP WANTED.              E.2
*A***                                                                          E.2
*A*       ORG     **1                                                         E.2
*A*RMET   GEN      /T(TFR2,JIR2),5(FT),7(QUOS),12(ASSPEC),22(ADNES),           E.2
*A*              C14(TRNA),15(LOG),17(GPROUT),24(RO)                          E.2
*A****                                                                         E.2
*A****    FLAG RELATION FALSE; TEST QUOS TO DECIDE IF JUMP WANTED.             E.2
*A****    THIS MICRO ADDR MUST BE EVEN.                                        E.2
*A*       ORG     **2                                                         E.2
*A*RNOT   GEN      /T(TFR2,JIR2),5(FT),7(QUOS),12(ASSPEC),22(AZERO),           E.2
*A*              C14(TRNA),15(LOG),17(GPROUT),24(RO)                          E.2
*A***                                                                          E.2
*A***     NO JUMP- RETURN LOGICAL VALUE IN RO;                                 E.2
*A***     EXIT TO ROM.                                                         E.2
*A***                                                                          E.2
*A*       ORG     **2                                                         E.2
*A*TFR2   GEN      /P(SS2M),7(PJMPS),6(MEMCS),TF0,10(IFSP)                     E.2
*A****                                                                         E.2
*A****    JUMP IF RELATION MET;                                                E.2
*A****    ADVANCE P TO SKIP JUMP ADDR OTHERWISE.                               E.2
*A****    THIS MICRO ADDR MUST BE EVEN.                                        E.2
*A****                                                                         E.2
*A*       ORG     **2                                                         E.2
*A*JIR2   GEN      /T(TFR2,AIF1),5(FT),7(GPRS),12(ASSPEC),22(ADNES),           E.2
*A*              C14(DFCA),17(GPROUT),24(RE),13(INCP)                         E.2
*A****                                                                         E.2
*A****    FETCH NEXT INSTRUCTION (OR IND ADDR).                                E.2
*A****    THIS MICRO ADDR MUST BE EVEN                                         E.2
*A*       ORG     **1                                                         E.2
```

```
*A*ATF2   GEN     /N(ATF3),10(TFSMIR),6(MEMC)                          E,2
*A**                                                                   E,2
*A**      TF INDIRECT ADDR, LOOP BACK TO FETCH EFFECTIVE ADDR;         E,2
*A**      SAVE EFFECTIVE ADDR IN P.                                    E,2
*A**                                                                   E,2
*A*ATF3   GEN     /T(AIF4,AIF2),5(FT),7(MIRS),11(BSSPEC),23(MIR),      E,2
*A*             C14(TRNB),15(LOG),13(POUT)                           E,2
*A**                                                                   E,2
*A**      CONTINUE RESTART OF PIPELINE;                                E,2
*A**      EXIT TO ROM.                                                 E,2
*A**                                                                   E,2
*A*ATF4   GEN     /P(SS3M),7(PJMPS),6(MEMCS),TFO,13(INCP),10(IFSP)     E,2
*A*       EJEC                                                        E,2
*A**                                                                   E,2
*A**      TWO WORD INTEGER MATH ROUTINES                               E,2
*A**                                                                   E,2
*A**      CALLING SEQUENCE:                                            E,2
*A**              (OPERAND 1 IN A/B REGISTER)                          E,2
*A**              BCS                                                  E,2
*A**              DATA  (ADDR OF OPERAND 2)                            E,2
*A**                                                                   E,2
*A**      BCS VALUES;                                                  E,2
*A**              IAD = 0105334                                        E,2
*A**              ISB = 0105374                                        E,2
*A**              IMV = 0105027                                        E,2
*A**              IDV = 0105067                                        E,2
*A**                                                                   E,2
*A*********************                                               E,2
*A***                                                                  E,2
*A***     IF OPERAND ADDRESS IS INDIRECT, LOOP BACK TO RESOLVE IT;     E,2
*A***     SAVE EFFECTIVE ADDRESS IN RC.                                E,2
*A***     (THIS MICRO ADDR MUST BE EVEN)                               E,2
*A***                                                                  E,2
*A*       ORG     *+1             LEAVE SPACE FOR IAR4                 E,2
*A*IAR1   GEN     /T(IAR3,IAR2),5(FT),7(MIRS),11(BSSPEC),23(MIR),      E,2
*A*             C14(TRNB),15(LOG),17(GPROUT),24(RC)                  E,2
*A*       ORG     *+1             LEAVE SPACE FOR SQTN1                E,2
*A***                                                                  E,2
*A***     FETCH INDIRECT ADDR                                          E,2
*A***     (THIS MICRO ADDR MUST BE EVEN)                               E,2
*A***                                                                  E,2
*A*IAR2   GEN     /N(IAR1),10(OFSMIR),6(MEMC)                          E,2
```

```
*A***                                                                  E.2
*A***      INITIATE FETCH OF SECOND OPERAND WORD; CLEAR OVERFLOW FLAG.  E.2
*A***                                                                  E.2
*A*IAR3    GEN     /N(IAR4),10(OPSALU),6(MEMC),12(ASGPR),24(RC),      E.2
*A*              C14(INCA),16(CRY1),7(ROVFL)                          E.2
*A***                                                                  E.2
*A***      SAVE FIRST WORD OF OPERAND 2                                E.2
*A***                                                                  E.2
*A*IADRG1  EQU     *             SAVE POSITION                        E.2
*A*        ORG     IAR1-1        RELOCATE IAR4                        E.2
*A*IAR4    GEN     /N(IAR5),11(BSSPEC),23(MIR),                       E.2
*A*              C14(TRNB),15(LOG),17(GPROUT),24(RC)                  E.2
*A*        ORG     IADRG1        RESTORE POSITION                     E.2
*A***                                                                  E.2
*A***      BEGIN FETCH OF NEXT INSTRUCTION TO RESTART PIPELINE AND     E.2
*A***      FORCE COMPLETION OF FETCH OF SECOND WORD OF OPERAND 2.       E.2
*A***                                                                  E.2
*A*IAR5    GEN     /N(IAR6),10(TESP),6(MEMC),13(INCP)                 E.2
*A***                                                                  E.2
*A***      SAVE SECOND WORD OF OPERAND 2;                              E.2
*A***      USE FIELD SELECT ON BCS TO BRANCH TO ADD/SUBTRACT OR        E.2
*A***      MULTIPLY/DIVIDE ROUTINES.                                   E.2
*A***                                                                  E.2
*A*IAR6    GEN     /F(IMD1),FSA,2(1),11(BSSPEC),23(MIR),              E.2
*A*              C14(TRNB),15(LOG),17(GPROUT),24(RD)                  E.2
*A*        FJEC                                                        E.2
*A***                                                                  E.2
*A***      MULTIPLY-DIVIDE COMMON MICROS                               E.2
*A***                                                                  E.2
*A***                                                                  E.2
*A***      INITIALIZES SIGN FLAG (RA AND QUOS);                        E.2
*A***      TESTS SIGN OF OP1 AND IF NEG, JUMPS TO COMPLEMENT IT.       E.2
*A***      (THIS MICRO ADDR MUST BE EVEN).                             E.2
*A***                                                                  E.2
*A*IMD1    GEN     /T(IMD2,IMD3),5(TT),7(GPRS),12(ASSPEC),22(AZERO),  E.2
*A*              C14(TRNA),15(LOG),17(GPROUT),24(RA),WF1,SCO          E.2
*A*        ORG     *+1           LEAVE SPACE FOR IAS1                 E.2
*A***                                                                  E.2
*A***      MAKE OP1 POSITIVE   ; TOGGLE SIGN FLAG .                    E.2
*A***                                                                  E.2
*A*IMD2    GMSK    /P(PAGE+ICMP),10(STACK),16(PUSH),                  E.2
*A*              C15(PAGE+IMD3),LB3,TF0,SF0                           E.2
*A***                                                                  E.2
```

```
*A***     TRANSFER MS(OP1) ;                                                    E.2
*A***     (THIS MICRO ADDR MUST BE EVEN)                                        E.2
*A***                                                                           E.2
*A**      ORG      **+1              ALLIGN EVEN LOCATIONS                      E.2
*A*IMD3   GEN       /N(IMD4),11(BSGPR),23(R0),14(TRNB),15(LOG),                E.2
*A*                C17(GPROUT),24(R8)                                          E.2
*A***                                                                          E.2
*A***     TRANSFER LS(OP1) ;                                                    E.2
*A***                                                                          E.2
*A*IMD4   GEN       /N(IMD5),11(BSGPR),23(R1),14(TRNB),15(LOG),                E.2
*A*                C17(GPROUT),24(R9)                                          E.2
*A***                                                                          E.2
*A***     SETUP MS(OP2) FOR POTENTIAL COMPLEMENT.                               E.2
*A***                                                                          E.2
*A*IMD5   GEN       /N(IMD6),11(BSGPR),23(RC),14(TRNB),15(LOG),                E.2
*A*                C17(GPROUT),24(R0)                                          E.2
*A***                                                                          E.2
*A***     SETUP LS(OP2);                                                        E.2
*A***     IF OP2 NOT NEGATIVE, SKIP COMPLEMENT.                                 E.2
*A***                                                                          E.2
*A*IMD6   GEN       /T(IMD8,IMD7),5(FT),7(GPRS),11(BSGPR),23(RD),              E.2
*A*                C14(TRNB),15(LOG),17(GPROUT),24(R1)                         E.2
*A****                                                                         E.2
*A****    MAKE OP2 POSITIVE; TOGGLE SIGN FLAG.                                  E.2
*A****    (THIS MICRO ADDR MUST BE EVEN)                                        E.2
*A****                                                                         E.2
*A*IMD7   GMSK      /P(PAGE+ICMP),10(STACK),16(PUSH),15(PAGE+IMD8),            E.2
*A*                CLB3,TF0,SF0                                                E.2
*A***                                                                          E.2
*A***     TRANSFER MS(OP2)                                                      E.2
*A***                                                                          E.2
*A*IMD8   GEN       /N(IMD9),11(BSGPR),23(R0),14(TRNB),15(LOG),                E.2
*A*                C17(GPROUT),24(RB)                                          E.2
*A***                                                                          E.2
*A***     CLEAR R0 INPREPARATION FOR FIRST MULTIPLY                             E.2
*A***                                                                          E.2
*A*IMD9   GEN       /N(IMD10),12(ASPEC),22(AZERO),14(TRNA),15(LOG),            E.2
*A*                C17(GPROUT),24(R0)                                          E.2
*A***                                                                          E.2
*A***     TRANSFER LS(OP2);                                                     E.2
*A***     USE FIELD SELECT TO CHOOSE BETWEEN MULTIPLY AND DIVIDE.               E.2
*A***                                                                          E.2
*A*IMD10  GEN       /F(IMU1),FS9,2(1),11(BSGPR),23(R1),14(TRNB),15(LOG),       E.2
```

```
*A*                C17(GPROUT),24(RC)                                    E.2
*A*        EJEC                                                          E.2
*A**                                                                     E.2
*A**       MICROS UNIQUE TO MULTIPLY                                     E.2
*A**                                                                     E.2
*A***                                                                    E.2
*A***      SETUP LS(OP1)FOR FIRST MULTIPLY;                              E.2
*A***      (THIS MICRO ADDR MUST BE EVEN).                               E.2
*A***                                                                    E.2
*A*IMU1    GEN      /N(IMU3),11(B$GPR),23(R9),14(TRNB),15(LOG),          E.2
*A*                C17(GPROUT),24(R1)                                    E.2
*A*        ORG      *+1              ALLOW SPACE FOR TOV1                E.2
*A**                                                                     E.2
*A**       MULTIPLY LS(OP1) AND LS(OP2)                                  E.2
*A**                                                                     E.2
*A*IMU3    GMSK     /P(PAGE+MUL),10(STACK),16(PUSH),15(PAGE+IMU4),       E.2
*A*                CLB3,TF0,SF0                                          E.2
*A**                                                                     E.2
*A**       SAVE LS(LSXLS)                                                E.2
*A**                                                                     E.2
*A*IMU4    GEN      /N(IMU5),11(B$GPR),23(R1),14(TRNB),15(LOG),          E.2
*A*                C17(GPROUT),24(RD)                                    E.2
*A**                                                                     E.2
*A**       SETUP MS(OP1) FOR NEXT MULTIPLY                               E.2
*A**                                                                     E.2
*A*IMU5    GEN      /N(IMU6),11(B$GPR),23(R8),14(TRNB),15(LOG),          E.2
*A*                C17(GPROUT),24(R1)                                    E.2
*A**                                                                     E.2
*A**       MULTIPLY MS(OP1) AND LS(OP2); (PLUS MS(LSXLS)).               E.2
*A**                                                                     E.2
*A*IMU6    GMSK     /P(PAGE+MUL),10(STACK),16(PUSH),15(PAGE+IMU7),       E.2
*A*                CLB3,TF0,SF0                                          E.2
*A**                                                                     E.2
*A**       SAMPLE MOST SIGNIFICANT WORD OF PRODUCT, MS(MSXLS).           E.2
*A**                                                                     E.2
*A*IMU7    GEN      /N(IMU8),12(A$GPR),24(R0),14(TRNA),15(LOG),          E.2
*A*                CTF0,SF0,7(S$ALU)                                     E.2
*A**                                                                     E.2
*A**       SAVE LS(MS*LS); BRANCH TO SET OVERFLOW IF                     E.2
*A**       MS(MS*LS) WAS NONZERO.                                        E.2
*A**                                                                     E.2
*A*IMU8    GEN      /T(IMU9,IMU10),5(FT),7(ALU7),11(B$GPR),23(R1),       E.2
*A*                C14(TRNB),15(LOG),17(GPROUT),24(RA)                   E.2
```

```
*A**                                                                    E.2
*A**      SET OVERFLOW; CLEAR R0, (UNUSED MEM REQ DUE TO SOVFL FIELDS).  E.2
*A**                                                                    E.2
*A*IMU0   GEN    /N(IMU10),12(ASSPEC),22(AZERO),14(TRNA),15(LOG),       E.2
*A*             C17(GPROUT),24(R0),7(SOVFL),10(TFSP),6(MEMC)           E.2
*A***                                                                   E.2
*A***     SETUP MS(OP2) FOR MULTIPLY                                    E.2
*A***     (THIS MICRO ADDR MUST BE EVEN).                               E.2
*A***                                                                   E.2
*A*       ORG    *+1             PUT ON EVEN LOCATION                   E.2
*A*IMU10  GEN    /N(IMU11),11(BSGPR),23(RB),14(TRNB),15(LOG),           E.2
*A*             C17(GPROUT),24(RC)                                     E.2
*A**                                                                    E.2
*A**      SETUP LS(OP1) FOR MULTIPLY                                    E.2
*A**                                                                    E.2
*A*IMU11  GEN    /N(IMU12),11(BSGPR),23(R0),14(TRNB),15(LOG),           E.2
*A*             C17(GPROUT),24(R1)                                     E.2
*A**                                                                    E.2
*A**      MULTIPLY LS(OP1) AND MS(OP2)                                  E.2
*A**                                                                    E.2
*A*IMU12  GMSK   /P(PAGE+MUL),10(STACK),18(PUSH),15(PAGE+IMU13),        E.2
*A*             CLB3,TF0,SF0                                           E.2
*A**                                                                    E.2
*A**      SAMPLE MOST SIGNIFICANT WORD OF PRODUCT, MS(LSXMS).           E.2
*A**                                                                    E.2
*A*IMU13  GEN    /N(IMU14),12(ASGPR),24(R0),14(TRNA),15(LOG),           E.2
*A*             CTF0,SF0,7(SSALU)                                      E.2
*A**                                                                    E.2
*A**      SAVE LS(LS*MS); BRANCH TO SET OVERFLOW IF                     E.2
*A**      MS(LS*MS) WAS NONZERO.                                        E.2
*A**                                                                    E.2
*A*IMU14  GEN    /T(IMU15,IMU16),5(FT),7(ALUZ),11(BSGPR),23(R1),        E.2
*A*             C14(TRNB),15(LOG),17(GPROUT),24(R0)                    E.2
*A**                                                                    E.2
*A**      SET OVERFLOW;                                                 E.2
*A**      CAUTION: THIS WILL RESULT IN OVERRIDE IF MEMORY ACTIVE.       E.2
*A**                                                                    E.2
*A*IMU15  GMSK   /N(IMU16),7(SOVFL),SF1,TF0,10(TFSOVR)                  E.2
*A***                                                                   E.2
*A***     GET MOST SIGNIFICANT WORD OF RESULT BY                        E.2
*A***     LS(LSXMS)+(LS(MSXLS)+MS(LSXLS)).                              E.2
*A***     (THIS MICRO ADDR MUST BE EVEN).                               E.2
*A***                                                                   E.2
```

```
*A*IMU16  GEN     /N(IMU17),12(ASGPR),24(RO),11(BSGPR),23(RA),          E.2
*A*               C14(ADD),17(GPROUT),7(SSOVFS)                         E.2
*A**                                                                    E.2
*A**      SETUP LEAST SIG WORD OF RESULT, LS(LSXLS),                    E.2
*A**                                                                    E.2
*A*IMU17  GEN     /N(IMU18),11(BSGPR),23(RO),14(TRNB),15(LOG),          E.2
*A*               C17(GPROUT),24(R1)                                    E.2
*A**                                                                    E.2
*A**      SAMPLE MS(OP2) AS PART OF FINAL OVERFLOW CHECK,               E.2
*A**                                                                    E.2
*A*IMU18  GEN     /N(IMU19),12(ASGPR),24(RB),14(TRNA),15(LOG),          E.2
*A*               C7(SSALU)                                             E.2
*A**                                                                    E.2
*A**      IF MS(OP2) WAS ZERO, COULDN'T CAUSE OVERFLOW SO JUMP TO EXIT; E.2
*A**      ELSE, CHECK MS(OP1),                                          E.2
*A**                                                                    E.2
*A*IMU19  GEN     /T(IMU23,IMU20),5(TT),7(ALUZ)                         E.2
*A**                                                                    E.2
*A**      CHECK MS(OP1)                                                 E.2
*A***     (THIS MICRO ADDR MUST BE EVEN)                                E.2
*A**                                                                    E.2
*A*IMU20  GEN     /N(IMU21),12(ASGPR),24(RB),14(TRNA),15(LOG),          E.2
*A*               C7(SSALU)                                             E.2
*A**                                                                    E.2
*A**      IF MS(OP1) IS ALSO NON-ZERO, JUMP TO SET OVERFLOW,            E.2
*A**                                                                    E.2
*A*IMU21  GEN     /T(IMU23,IMU22),5(TT),7(ALUZ)                         E.2
*A**                                                                    E.2
*A**      SET OVERFLOW; CAUTION; MEMORY OVERRIDE IF MEMORY ACTIVE       E.2
*A***     (THIS MICRO ADDR MUST BE EVEN)                                E.2
*A**                                                                    E.2
*A*IMU22  GEN     /N(IMU23),7(SOVFL),SF1,TF0,10(TFSOVR)                 E.2
*A**                                                                    E.2
*A**      TEST SIGN FLAG (QUOS) TO SEE IF RESULT SHOULD BE NEGATIVE,    E.2
*A**                                                                    E.2
*A*IMU23  GEN     /T(IMU25,IMU24),5(FT),7(QUOS)                         E.2
*A**                                                                    E.2
*A**      MAKE RESULT NEGATIVE                                          E.2
*A***     (THIS MICRO ADDR MUST BE EVEN)                                E.2
*A**                                                                    E.2
*A*IMU24  GMSK     /P(PAGE+TCMP),10(STACK),16(PUSH),                     E.2
*A*               C15(PAGE+IMU25),LB3,TF0,SF0                           E.2
*A**                                                                    E.2
```

```
*A**      EXIT BACK TO ROM                                                    E.2
*A**                                                                          E.2
*A*IMU25  GEN    /P(SS2M),10(PJMP),SF0,TF0                                   E.2
*A*       EJEC                                                               E.2
*A**                                                                          E.2
*A**      UNIQUE DIVIDE MICROS                                                E.2
*A**                                                                          E.2
*A*DVORG1 EQU    *                                                           E.2
*A**                                                                          E.2
*A**      SAMPLE MS(OP2) TO SEE IF SHOULD DIVIDE BY IT.                       E.2
*A**                                                                          E.2
*A*       ORG    IMU1+1          PLACE IN FIELD SELECT TABLE                 E.2
*A*IDV1   GEN    /N(IDV2),12(ASGPR),24(RB),14(TRNA),15(LOG),                 E.2
*A*             C7(SSAL1)                                                   E.2
*A*       ORG    DVORG1          RESTORE ORG ADDR                            E.2
*A**                                                                          E.2
*A**      CLEAR R01 IF MS(OP2) WAS NOT ZERO, JUMP TO DIVIDE BY IT.            E.2
*A**      ELSE, CONTINUE IN SEQUENCE TO DIVIDE BY 1S.                         E.2
*A**                                                                          E.2
*A*IDV2   GEN    /T(IDV3,IDV20),5(TT),7(ALU2),12(ASSPEC),22(AZERO),          E.2
*A*             C14(TRNA),15(LOG),17(GPROUT),24(R0)                         E.2
*A**                                                                          E.2
*A**      CHECK IF LS(OP2) IS ZERO                                            E.2
*A**                                                                          E.2
*A*IDV3   GEN    /N(IDV4),12(ASGPR),24(RC),14(TRNA),15(LOG),                 E.2
*A*             C7(SSAL1)                                                   E.2
*A*       ORG    *+1             LEAVE ROOM FOR IDV20                        E.2
*A**                                                                          E.2
*A**      SETUP MS(OP1) FOR DIVIDE;                                           E.2
*A**      IF LS(OP2) WAS ALSO ZERO, JUMP TO OVERFLOW EXIT.                    E.2
*A**                                                                          E.2
*A*IDV4   GEN    /T(IDVOF,IDV5),5(TT),7(ALU7),11(BSGPR),23(R8),              E.2
*A*             C14(TRNB),15(LOG),17(GPROUT),24(R1)                         E.2
*A**                                                                          E.2
*A**      DIVIDE MS(OP1) BY LS(OP2)                                           E.2
*A**                                                                          E.2
*A*IDV5   GMSK   /P(PAGE+DIV),10(STACK),18(PUSH),                            E.2
*A*             C15(PAGE+IDV6),LB3,TF0,SF0                                  E.2
*A**                                                                          E.2
*A**      SAVE QUOTIENT AS MOST SIG WORD OF RESULT.                           E.2
*A**                                                                          E.2
*A*IDV6   GEN    /N(IDV7),11(BSGPR),23(R1),14(TRNB),15(LOG),                 E.2
*A*             C17(GPROUT),24(RD)                                          E.2
```

```
*A**                                                                    E.2
*A**      SETUP LS(OP1) FOR DIVIDE                                      E.2
*A**                                                                    E.2
*A*IDV7   GEN     /N(IDV8),11(BSGPR),23(R9),14(TRNB),15(LOG),          E.2
*A*              C17(GPROUT),24(R1)                                    E.2
*A**                                                                    E.2
*A**      DIVIDE REMAINDER + LS(OP1) BY LS(OP2).                        E.2
*A**                                                                    E.2
*A*IDV8   GMSK    /P(PAGE+DIV),10(STACK),16(PUSH),                     E.2
*A*              C15(PAGE+IDV9),LB3,TF0,SF0                            E.2
*A**                                                                    E.2
*A**      SETUP MOST STG WORD OF RESULT IN R0 FOR RETURN.               E.2
*A**      LEAST SIG WORD IS IN R1 FROM PREVIOUS DIVISION.               E.2
*A**                                                                    E.2
*A*IDV9   GEN     /N(IDV10),11(BSGPR),23(R0),14(TRNB),15(LOG),         E.2
*A*              C17(GPROUT),24(R0)                                    E.2
*A**                                                                    E.2
*A**      SAMPLE STGN FLAG TO SEE IF RESULT SHOULD BE NEGATIVE.         E.2
*A**                                                                    E.2
*A*IDV10  GEN     /N(IDV11),12(ASGPR),24(RA),14(TRNA),15(LOG),         E.2
*A*              C7(SSALU)                                             E.2
*A**                                                                    E.2
*A**      IF RESULT SHOULD BE POSITIVE, JUMP TO EXIT.                   E.2
*A**                                                                    E.2
*A*IDV11  GEN     /T(IDV12,IDV13),5(TT),7(ALUS)                        E.2
*A**                                                                    E.2
*A**      MAKE RESULT NEGATIVE                                          E.2
*A**                                                                    E.2
*A*IDV12  GMSK    /P(PAGE+ICMP),10(STACK),16(PUSH),                    E.2
*A*              C15(PAGE+IDV13),LB3,TF0,SF0                           E.2
*A**                                                                    E.2
*A**      EXIT TO ROM.                                                  E.2
*A**                                                                    E.2
*A*IDV13  GEN     /P(SS2M),10(PJMP),SF0,TF0                            E.2
*A*       EJEC                                                         E.2
*A***                                                                   E.2
*A***     OVERFLOW EXIT = ATTEMPTED DIVISION BY ZERO;                   E.2
*A**      SET OVERFLOW; CAUTION: MEM OVERRIDE IF MEM ACTIVE             E.2
*A**      RESTORE MS(OP1)                                               E.2
*A***     (THIS MICRO ADDR MUST BE EVEN).                               E.2
*A***                                                                   E.2
*A*IDVOF  GEN     /N(IDVOF3),7(SOVFL),SF1,TF0,10(IFSOVR),              E.2
*A*              C11(BSGPR),23(RA),14(TRNB),15(LOG),17(GPROUT),24(R0)  E.2
```

```
*A**                                                                    E.2
*A**      RESTORE LS(OP1) ; JUMP TO CHECK SIGN THEN EXIT.               E.2
*A**                                                                    E.2
*A*IDVOF3 GEN     /N(IDV10),11(BSGPR),23(R9),14(TRNB),15(LOG),          E.2
*A*             C17(GPROUT),24(R1)                                     E.2
*A*       EJEC                                                         E.2
*A**                                                                    E.2
*A**      SETUP MS(OP1) FOR DIVISION                                    E.2
*A**                                                                    E.2
*A*DVORG2 EQU     *                                                    E.2
*A*       ORG     IDV3+1          ADJUST ORG TO ALLOW COND ADDR,       E.2
*A*IDV20  GEN     /N(IDV21),11(BSGPR),23(R8),14(TRNB),15(LOG),         E.2
*A*             C17(GPROUT),24(R1)                                     E.2
*A*       ORG     DVORG2          RESTORE ORG ADDR                     E.2
*A**                                                                    E.2
*A**      SAVE LS(OP2) FOR LATER USE                                    E.2
*A**                                                                    E.2
*A*IDV21  GEN     /N(IDV22),11(BSGPR),23(RC),14(TRNB),15(LOG),          E.2
*A*             C17(GPROUT),24(RD)                                     E.2
*A**                                                                    E.2
*A**      SETUP MS(OP2) FOR DIVISION                                    E.2
*A**                                                                    E.2
*A*IDV22  GEN     /N(IDV23),11(BSGPR),23(RB),14(TRNB),15(LOG),          E.2
*A*             C17(GPROUT),24(RC)                                     E.2
*A**                                                                    E.2
*A**      DIVIDE MS(OP1) BY MS(OP2) TO GET TRIAL QUOTIENT               E.2
*A**                                                                    E.2
*A*IDV23  GMSK    /P(PAGE+DIV),10(STACK),16(PUSH),                      E.2
*A*             C15(PAGE+IDV24),LR3,TF0,SF0                            E.2
*A**                                                                    E.2
*A**      SAVE REMAINDER                                                E.2
*A**                                                                    E.2
*A*IDV24  GEN     /N(IDV25),11(BSGPR),23(R0),14(TRNB),15(LOG),          E.2
*A*             C17(GPROUT),24(R8)                                     E.2
*A**                                                                    E.2
*A**      SETUP QUOTIENT FOR MULTIPLY CHECK.                            E.2
*A**                                                                    E.2
*A*IDV25  GEN     /N(IDV26),11(BSGPR),23(R1),14(TRNB),15(LOG),          E.2
*A*             C17(GPROUT),24(RC)                                     E.2
*A**                                                                    E.2
*A**      SETUP LS(OP1) FOR MULTIPLY                                    E.2
*A**                                                                    E.2
*A*IDV26  GEN     /N(IDV27),12(ASSPEC),22(AZERO),14(TRNA),15(LOG),      E.2
```

```
*A*                 C17(GPROUT),24(R0)                                    E.2
*A*IDV27  GEN        /N(IDV28),11(B$GPR),23(RD),14(TRNB),15(LOG),         E.2
*A*                 C17(GPROUT),24(R1)                                    E.2
*A**                                                                      E.2
*A**      MULTIPLY TRIAL QUOTIENT BY LS(OP1)                              E.2
*A**                                                                      E.2
*A*IDV28  GMSK       /P(PAGE+MUL),10(STACK),16(PUSH),                     E.2
*A*                 C15(PAGE+IDV29),LB3,TF0,SF0                           E.2
*A**                                                                      E.2
*A**      COMPARE MOST SIG WORD OF PRODUCT TO REMAINDER FROM THE          E.2
*A**      DIVISION;                                                       E.2
*A**                                                                      E.2
*A*IDV29  GEN        /N(IDV30),12(A$GPR),24(R8),11(B$GPR),23(R0),         E.2
*A*                 C14(SUB),16(CRY1),7(S$ALU)                            E.2
*A**                                                                      E.2
*A**      IF NOT EQUAL, SKIP SECOND WORD COMPARE                          E.2
*A**                                                                      E.2
*A*IDV30  GEN        /T(IDV31,IDV32),5(TT),7(ALUZ)                        E.2
*A**                                                                      E.2
*A**      COMPARE LEAST SIG WORD OF PRODUCT TO LS(OP1)                    E.2
*A**                                                                      E.2
*A*IDV31  GEN        /N(IDV32),12(A$GPR),24(R9),11(B$GPR),23(R1),         E.2
*A*                 C14(SUB),16(CRY1),7(S$ALU)                            E.2
*A**                                                                      E.2
*A**      GREATER? IF PRODUCT OF TRIAL QUOTIENT AND LS(OP2) GREATER       E.2
*A**      THAN REMAINDER+LS(OP1), THEN TRIAL QUOTIENT ONE TOO LARGE;      E.2
*A**      ELSE TRIAL QUOTIENT IS ACCURATE.                                E.2
*A**                                                                      E.2
*A*IDV32  GEN        /T(IDV33A,IDV33B),5(FT),7(ALUS)                      E.2
*A**                                                                      E.2
*A**      RETURN TRIAL QUOTIENT AS RESULT                                 E.2
*A**                                                                      E.2
*A*IDV33A GEN        /N(IDV34),11(B$GPR),23(RC),14(TRNB),15(LOG),         E.2
*A*                 C17(GPROUT),24(R1)                                    E.2
*A***                                                                     E.2
*A***     RETURN TRIAL QUOTIENT -1 AS RESULT                              E.2
*A***     (THIS MICRO ADDR MUST BE EVEN).                                 E.2
*A***                                                                     E.2
*A*IDV33B GEN        /N(IDV34),11(B$GPR),23(RC),14(DECB),12(A$SPEC),      E.2
*A*                 C22(ADNFS),17(GPROUT),24(R1)                          E.2
*A**                                                                      E.2
*A**      CLEAR MOST SIG WORD OF RESULT; EXIT TO SIGN CHECKS.             E.2
*A**                                                                      E.2
```

```
*A*IDV34  GEN    /N(IDV10),12(A$SPEC),22(AZERO),14(TRNA),15(LOG),          E.2
*A*              C17(GPROUT),24(RO)                                        E.2
*A*       EJEC                                                             E.2
*A**                                                                       E.2
*A**      FIRST MICRO UNIQUE TO DOUBLE ADD/SUBTRACT ;                      E.2
*A**      SHIFTS LS(OP2) LEFT IN PREPARATION FOR ADD/SUB;                  E.2
*A**      USES FIELD SELECT TO CHOOSE ADD OR SUBTRACT.                     E.2
*A**                                                                       E.2
*A*IAORG2 EQU    *              SAVE POSITION                             E.2
*A*       ORG    IMD1+1         PLACE MICRO IN FSEL TBL                   E.2
*A*IAS1   GEN    /F(IAS2),FS9,2(1),12(A$GPRL),24(RO),                      E.2
*A*              C14(TRNA),15(LOG),17(GPROUT)                              E.2
*A*       ORG    IAORG2         RESTORE POSITION                          E.2
*A***                                                                      E.2
*A***     ADD LS WORDS OF OPERANDS ; SAMPLE CARRY                          E.2
*A***     (THIS MICRO ADDR MUST BE EVEN).                                  E.2
*A***                                                                      E.2
*A*IAS2   GEN    /N(IAS4),12(A$GPRL),24(R1),11(B$GPR),23(RO),              E.2
*A*              C14(ADD),17(GPROUT),7(S$ALU)                              E.2
*A**                                                                       E.2
*A**      SUBTRACT LS WORDS OF OPERANDS; SAMPLE CARRY                      E.2
*A**                                                                       E.2
*A*IAS3   GEN    /N(IAS4),12(A$GPRL),24(R1),11(B$GPR),23(RO),              E.2
*A*              C14(SUB),16(CRY1),17(GPROUT),7(S$ALU)                     E.2
*A**                                                                       E.2
*A**      SHIFT LS RESULT RIGHT ;                                          E.2
*A**      FIELD SELECT JUMP TO ADD OR SUBTRACT MS WORDS.                   E.2
*A**                                                                       E.2
*A*       ORG    *+2            RELOCATE IAS4                             E.2
*A*IAS4   GEN    /F(IAS5),FS9,2(1),12(A$GPRR),24(R1),SH4,                  E.2
*A*              C14(TRNA),15(LOG),17(GPROUT)                              E.2
*A***                                                                      E.2
*A***     ADD MS WORDS; (WITH STORED CARRY); SAMPLE OVERFLOW AND ALU;       E.2
*A***     RETURN TO ROM.                                                   E.2
*A***     (THIS MICRO ADDR MUST BE EVEN)                                   E.2
*A***                                                                      E.2
*A*       ORG    *-3            BACK UP TO EVEN PATR                      E.2
*A*IAS5   GEN    /P(SS2M),SF0,TF0,10(PJMP),GFA,12(A$GPR),24(RO),           E.2
*A*              C11(B$GPR),23(RC),14(ADD),CF1,17(GPROUT)                  E.2
*A**                                                                       E.2
*A**      SUBTRACT MS WORDS, WITH STORED CARRY; SAMPLE OVERFLOW AND ALU.   E.2
*A**      RETURN TO ROM.                                                   E.2
*A**                                                                       E.2
```

```
*A*IAS6   GEN     /P(SS2M),SF0,TF0,10(PJMP),GFA,12(ASGPR1,24(R0),        F.2
*A*              C11(BSGPR),23(RC),14(SUB),CF1,17(GPROUT)               E.2
*A*       ORG     *+1             SKIP PAST RELOCATED IAS4               E.2
*A*       FJFC                                                           E.2
*A**                                                                     F.2
*A**      COMPLEMENT DOUBLE INTEGER SUBROUTINE                           F.2
*A**                                                                     F.2
*A**      R0,R1 CONTAINS INTEGER                                         E.2
*A**                                                                     E.2
*A**      RA IS ONES COMPLEMENTED TO ACT AS SIGN FLAG.                   E.2
*A**      OUOS RECORDS SIGN OF RA.                                       E.2
*A**                                                                     E.2
*A**********                                                            E.2
*A**                                                                     E.2
*A**      CHECK LEAST SIG WORD                                           E.2
*A**                                                                     E.2
*A*ICMP    GEN     /N(ICMP1),12(ASGPR),24(R1),14(TRNA),15(LOG),7(SSALU)  F.2
*A**                                                                     E.2
*A**      IF LS ZERO, JUMP TO TWO'S COMPL MS)                            E.2
*A**                                                                     E.2
*A*ICMP1   GEN     /T(ICMP2,ICMP9),5(FT),7(ALUZ)                        E.2
*A**                                                                     E.2
*A**      TWO'S COMPLEMENT LS                                            E.2
*A**                                                                     E.2
*A*ICMP2   GEN     /N(ICMP3),11(BSGPR),23(R1),14(TCB),16(CRY1),          E.2
*A*              C12(ASSPEC),22(AZERO),17(GPROUT),24(R1)                E.2
*A**                                                                     E.2
*A**      CLEAR SIGN OF LS                                               E.2
*A**                                                                     E.2
*A*ICMP3   GMSK    /N(ICMP4),12(ASGPR),16(R1),11(LIT),15(X'8000),        E.2
*A*              C14(AND),13(OPROUT)                                    E.2
*A*ICMP4   GEN     /N(ICMP5),11(BSSPEC),23(OPR),14(TRNB),15(LOG),        E.2
*A*              C17(GPROUT),24(R1)                                     E.2
*A**                                                                     E.2
*A**      ONES COMPLEMENT MS                                             E.2
*A**                                                                     E.2
*A*ICMP5   GEN     /N(ICMP6),12(ASGPR),24(R0),                           E.2
*A*              C14(NOTA),15(LOG),17(GPROUT)                           E.2
*A**                                                                     E.2
*A**      ONES COMPLEMENT SIGN FLAG RA) SET OUOS)                        E.2
*A**                                                                     E.2
*A*ICMP6   GEN     /N(ICMP7),12(ASGPR),24(RA),14(NOTA),15(LOG),          E.2
*A*              C17(GPROUT),WF1,SC0                                    E.2
```

```
*A**                                                                    E.2
*A**        EXIT TO CALLER                                              E.2
*A**                                                                    E.2
*A*ICMP7    GEN     10(STACK),24(POPJMP),L83,TF0,SF0,BR0                E.2
*A**                                                                    E.2
*A**        DECREMENT MS TO GET TWOS COMPL                              E.2
*A**                                                                    E.2
*A**        ORG     *+1             PUT ON EVEN LOCATION                E.2
*A*ICMP9    GEN     /N(ICMP5),12(ASGPR),24(R0),14(DECA),                E.2
*A*             C17(GPROUT)                                             E.2
*A*         FJEC                                                        E.2
*A**                                                                    E.2
*A**        SQUARE ROOT AND INDEXING COMMON MICROS                      E.2
*A**                                                                    E.2
*A**                                                                    E.2
*A**        CHECK FOR SIGN BIT ON INDEX ADDR;                           E.2
*A**        IF SET, LOOP ON TOP2 UNTIL REACH END OF IND ADDR CHAIN.     E.2
*A**        ALSO SAVE ADDR OF OPERAND                                   E.2
*A**        THEN FIELD SELECT TO EITHER SQUARE ROOT OR INDEX            E.2
*A**                                                                    E.2
*A*         ORG     TAR1+1          PLACE MICRO IN FIELD SELECT TABLE   E.2
*A*SQIN1    GEN     /S(SQ1,SQIN2),5(FT),7(MIRS),FS9,2(1),               E.2
*A*             C11(BSSPEC),23(MIR),14(TRNB),15(LOG),17(GPROUT),24(RB)  E.2
*A**                                                                    E.2
*A**        FOLLOW INDIRECT ADDRESS CHAIN                               E.2
*A**                                                                    E.2
*A*         ORG     X'182           PLACE TO ALLOW TEST ADDRESSING      E.2
*A*SQIN2    GEN     /N(SQIN1),6(MEMC),10(DF$MIR)                        E.2
*A*         FJEC                                                        E.2
*A**                                                                    E.2
*A**        SINGLE PRECISION FLOATING POINT SQUARE ROOT                 E.2
*A**                                                                    E.2
*A**        USES TABLE LOOKUP TO GET INITIAL GUESS, THEN PERFORMS       E.2
*A**        TWO ITERATIONS THROUGH  Y2= 1/2(X/Y1 +Y1).                  E.2
*A**                                                                    E.2
*A**        CALLING SEQUENCE:                                           E.2
*A**                                                                    E.2
*A**                DATA    0105127                                     E.2
*A**                DATA    ADDR OF OPERAND                             E.2
*A**                                                                    E.2
*A**        RESULT RETURNED IN AR REG.                                  E.2
*A**                                                                    E.2
*A**        REGISTER USAGE:                                             E.2
```

```
*A**                                                                    E.2
*A**            R0,R1 - TEMPORARY VALUES; FINAL RESULT                  E.2
*A**            R8,R9 - LEFT JUSTIFIED OPERAND (WITHOUT EXPONENT),      E.2
*A**            RA    - EXPONENT                                        E.2
*A**            RC    - Y1, INTERMEDIATE RESULT                         E.2
*A**            OPR   - TEMP, STORAGE;(BYTE SWAPPING USED FOR SHIFT 8)  E.2
*A**                                                                    E.2
*A**                                                                    E.2
*A**      START FETCH OF SECOND WORD OF OPERAND; CLEAR OVERFLOW,        E.2
*A***     THIS MICRO MUST BE ON A 16 WORD BOUNDARY TO ALLOW 'S'         E.2
*A**                                                                    E.2
*A*       ORG     Y'180         PLACE ON 16 WORD BLOCK                  E.2
*A*S01    GEN     /N(S02),10(OFSALU),6(MEMC),12(ASGPR),24(R8),          E.2
*A*             C14(INCA),16(CRY1),7(ROVFL)                            E.2
*A*       ORG     *+2           LEAVE ROOM FOR IOP3 AND SOIN2           E.2
*A**                                                                    E.2
*A**      FIRST OPERAND WORD TO R8; SAMPLE                              E.2
*A**                                                                    E.2
*A*S02    GEN     /N(S03),11(BSSPEC),23(MIR),14(TRNB),15(LOG),          E.2
*A*             C17(GPROUT),24(R8),7(SSALU)                            E.2
*A**                                                                    E.2
*A**      RESTART PIPELINE FOR EXIT IF NEGATIVE ; CLEAR R0,             E.2
*A**                                                                    E.2
*A*S03    GEN     /T(S04,SOOVF),5(FT),7(ALUS),10(IFSP),6(MEMC),         E.2
*A*             C13(INCP),12(ASSPEC),22(AZERO),14(TRNA),15(LOG),       E.2
*A*             C17(GPROUT),24(R0)                                     E.2
*A**                                                                    E.2
*A**      SAVE SECOND OPERAND WORD IN R9                                E.2
*A**                                                                    E.2
*A*S04    GEN     /N(S05),11(BSSPEC),23(MIR),14(TRNB),15(LOG),          E.2
*A*             C17(GPROUT),24(R9)                                     E.2
*A**                                                                    E.2
*A**      OVERFLOW EXIT = CAN'T TAKE ROOT OF NEGATIVE; CLEAR R0,R1,     E.2
*A**                                                                    E.2
*A*SOOVF  GEN     /N(SOOVF1),12(ASSPEC),22(AZERO),14(TRNA),15(LOG),     E.2
*A*             C17(GPROUT),24(R0),10(IFSP),6(MEMC),7(SOVFL)           E.2
*A*SOOVF1 GEN     /P(SS2M),10(PJMP),SF0,TF0,12(ASSPEC),22(AZERO),       E.2
*A*             C14(TRNA),15(LOG),17(GPROUT),24(R1)                    E.2
*A**                                                                    E.2
*A**      ADD 0201 TO EXPONENT TO OBTAIN CORRECT BIAS AFTER SHIFT,      E.2
*A**                                                                    E.2
*A*S05    GMSK    /N(S06),12(ASGPR),16(R8),11(LIT),15(X'8F7F),          E.2
*A*             C14(ADD),13(OPROUT)                                    E.2
```

```
*A**                                                                    E.2
*A**      SAMPLE ADJUSTED EXPONENT EVEN OR ODD                          E.2
*A**                                                                    E.2
*A*S06    GEN     /N(S07),11(BSSPEC),23(ORSE),14(TRNB),15(LOG),7(SSALU) E.2
*A**                                                                    E.2
*A**      TRANSFER EXPONENT (OPR LEFT BYTE),TO R4,                      E.2
*A**                                                                    E.2
*A*S07    GEN     /N(S08),11(BSSPEC),23(OLSE),      EXP TO RGHT BYTE    E.2
*A*             C14(TRNB),15(LOG),13(OPROUT)                           E.2
*A*S08    GEN     /N(S09),11(BSSPEC),23(ORLZ),      EXP TO LEFT BYTE    E.2
*A*             C14(TRNB),15(LOG),17(GPROUT),24(R4)                    E.2
*A**                                                                    E.2
*A**      TRANSFER REST OF FIRST OP WORD TO R8                          E.2
*A**                                                                    E.2
*A*S09    GMSK    /N(S010),12(ASGPR),16(R8),11(LIT),15(X'FF80'),        E.2
*A*             C14(AND),13(OPROUT)                                    E.2
*A*S010   GEN     /N(S011),11(BSSPEC),23(ORLZ),14(TRNB),15(LOG),        E.2
*A*             C17(GPROUT),24(R8)                                     E.2
*A**                                                                    E.2
*A**      SETUP SECOND OP WORD FOR COMBINATION WITH FIRST               E.2
*A**                                                                    E.2
*A*S011   GEN     /N(S012),12(ASGPRL),24(R9),14(TRNA),15(LOG),          E.2
*A*             C13(OPROUT)                                            E.2
*A**                                                                    E.2
*A**      SHIFT BY 8 (WITH BYTE SWOP) AND PUT RESULT BACK IN R9         E.2
*A**                                                                    E.2
*A*S012   GEN     /N(S013),11(BSSPEC),23(ORLZ),14(TRNB),15(LOG),        E.2
*A*             C17(GPROUT),24(R9)                                     E.2
*A**                                                                    E.2
*A**      RIGHT JUSTIFY LEFT BYTE TO STMULATE A BIT SHIFT INTO NEXT WORDE.2
*A**                                                                    E.2
*A*S013   GEN     /N(S014),11(BSSPEC),23(OLSE),                         E.2
*A*             C14(TRNB),15(LOG),13(OPROUT)                           E.2
*A**                                                                    E.2
*A**      COMBINE FIRST OP WORD AND TOP OF SECOND OP WORD;              E.2
*A**      IF EXPONENT WAS ODD,  JUMP TO SKIP RIGHT SHIFT.               E.2
*A**                                                                    E.2
*A*S014   GEN     /T(S017,S015),5(TT),7(ALUS),12(ASGPR),24(R8),         E.2
*A*             C11(BSSPEC),23(ORZF),14(OR),17(GPROUT)                 E.2
*A***                                                                   E.2
*A***     SHIFT COMBINED FIRST OP WORD RIGHT ONE                        E.2
*A***     (THIS MICRO ADDR MUST BE EVEN)                                E.2
*A***                                                                   E.2
```

```
*A*S015   GEN     /N(S016),12(ASGPRR),SH4,24(R8),14(TRNA),15(LOG),          E.2
*A*             C17(GPROUT)                                                E.2
*A**                                                                       E.2
*A**      SHIFT SECOND WORD OF COMBINED OP RIGHT ONE;                      E.2
*A**      SET BIT 15 FROM COPY OF FIRST WORD IN OPR                        E.2
*A**                                                                       E.2
*A*S016   GEN     /N(S017),12(ASGPRR),24(R9),SH3,14(TRNA),15(LOG),          E.2
*A*             C17(GPROUT)                                                E.2
*A**                                                                       E.2
*A**      TRANSFER FIRST OP WORD TO IBR FOR FIELD SELECTION CHOICE         E.2
*A**      OF INITIAL Y1 VALUES.                                            E.2
*A**                                                                       E.2
*A*S017   GEN     /N(S018),12(ASGPRR),24(R8),14(TRNA),15(LOG),              E.2
*A*             CSF0,IM4                                                   E.2
*A*S018   GEN     /N(S019),12(ASGPRR),24(R8),14(TRNA),15(LOG)               E.2
*A**                                                                       E.2
*A**      CLEAR SIGN BIT OF SECOND OP WORD BY RIGHT SHIFT;                 E.2
*A**      TRANSFER FIRST OP WORD ON TO I.                                  E.2
*A**                                                                       E.2
*A*S019   GEN     /N(S020),12(ASGPRR),SH4,24(R9),14(TRNA),15(LOG),          E.2
*A*             C17(GPROUT),TF0,SF0,7(IBR$I)                               E.2
*A**                                                                       F.2
*A**      SETUP FIRST OP WORD FOR MULTIPLY;                                F.2
*A**      USE FIELD SELECT TO CHOOSE 'K' VALUE TO MULTIPLY.                F.2
*A**                                                                       E.2
*A*S020   GEN     /F(S021A),FSP,2(7),11(BSGPR),23(R8),                     F.2
*A*             C14(TRNB),15(LOG),17(GPROUT),24(R1)                        E.2
*A***                                                                      E.2
*A***     CHOOSE INITIAL VALUE FOR MULTIPLY;                               E.2
*A***     THIS TABLE MUST BE ON AN EVEN 8 WORD BOUNDARY.                   E.2
*A***                                                                      E.2
*A*       ORG     X'198                                                    E.2
*A*S021A  GEN     /N(SOOVF)      OVERFLOW - NOT NORMALIZED OP              E.2
*A*S021B  GEN     /N(SOOVF)      OVERFLOW - NOT NORMALIZED OP              F.2
*A*S021C  GMSK    /N(S022),11(LIT),15(X'8CED),      T(V6-2)   071422        E.2
*A*             C14(TRNB),13(OPROUT)                                       E.2
*A*S021D  GMSK    /N(S022),11(LIT),15(X'9EFD),      2(V)-V6)  060402        E.2
*A*             C14(TRNB),13(OPROUT)                                       E.2
*A*S021E  GMSK    /N(S022),11(LIT),15(X'AEA1),      2(V3-V2)  050536        E.2
*A*             C14(TRNB),13(OPROUT)                                       E.2
*A*S021F  GMSK    /N(S022),11(LIT),15(X'AEA1),                050536        E.2
*A*             C14(TRNB),13(OPROUT)                                       F.2
*A*S021G  GMSK    /N(S022),11(LIT),15(X'BB87),      2(2-V3)   042230        E.2
```

```
*A*                C14(TRNB),13(OPROUT)                                    E.2
*A*S021H  GMSK      /N(S022),11(LIT),15(X'BB67),                  042230    E.2
*A*                C14(TRNB),13(OPROUT)                                    E.2
*A**                                                                       E.2
*A**       SETUP CONSTANT VALUE FOR MULTIPLY.                              E.2
*A**                                                                       E.2
*A*S022    GEN      /N(S023),11(B$SPEC),23(OPR),14(TRNB),15(LOG),          E.2
*A*                C17(GPROUT),24(RC)                                      E.2
*A**                                                                       E.2
*A**       MULTIPLY CONSTANT BY FIRST WORD OF OPERAND                      E.2
*A**                                                                       E.2
*A*S023    GMSK     /P(PAGE+MUL),10(STACK),16(PUSH),                       E.2
*A*                C15(PAGE+S024),LB3,TF0,SF0                              E.2
*A**                                                                       E.2
*A**       FIELD SELECT TO CHOOSE SECOND CONSTANT TO ADD TO RESULT         E.2
*A**                                                                       E.2
*A*S024    GEN      /F(S025A),FSF,2(7)                                     E.2
*A***                                                                      E.2
*A***      CHOOSE CONSTANT AND ADD INTO PRODUCT                            E.2
*A***      THIS TABLE MUST BE ON AN EVEN 8 WORD BOUNDARY                   E.2
*A***                                                                      E.2
*A*        ORG      Y'1AB                                                  F.2
*A*S025A   GEN      /N(SOOVF)      OVERFLOW - BAD OP                       E.2
*A*S025B   GEN      /N(SOOVF)      OVERFLOW - BAD OP                       E.2
*A*S025C   GMSK     /N(S026),12(A$GPR),16(R0),11(LIT),14(ADD),13(OPROUT),  F.2
*A*                C15(D'156225)  (26-7V6)132                    021552    E.2
*A*S025D   GMSK     /N(S026),12(A$GPR),16(R0),11(LIT),14(ADD),13(OPROUT),  E.2
*A*                C15(D'152742)  (26V2-15V6)/32                 025035    F.2
*A*S025E   GMSK     /N(S026),12(A$GPR),16(R0),11(LIT),14(ADD),13(OPROUT),  E.2
*A*                C15(D'146752)  (13V2-7V3)/16                  031025    E.2
*A*S025F   GMSK     /N(S026),12(A$GPR),16(R0),11(LIT),14(ADD),13(OPROUT),  E.2
*A*                C15(D'146752)                                 031025    E.2
*A*S025G   GMSK     /N(S026),12(A$GPR),16(R0),11(LIT),14(ADD),13(OPROUT),  E.2
*A*                C15(D'142160)  1-2(2-V3)                      035617    E.2
*A*S025H   GMSK     /N(S026),12(A$GPR),16(R0),11(LIT),14(ADD),13(OPROUT),  E.2
*A*                C15(D'142226)  1-2(2-V3)(1-2**(-15))          035551    E.2
*A**                                                                       F.2
*A**       TRANSFER SUM TO RC FOR DIVIDE;                                  E.2
*A**       SAMPLE FOR OVERFLOW,(IF ALU SIGN),                              E.2
*A**                                                                       F.2
*A*S026    GEN      /N(S027),11(B$SPEC),23(OPR),14(TRNB),15(LOG),          E.2
*A*                C17(GPROUT),24(RC),7(S$ALU)                             E.2
*A**                                                                       E.2
```

```
*A**       SETUP FIRST OP WORD FOR DIVIDE;                                    E.2
*A**                                                                          E.2
*A*S027    GEN     /N(S028),11(B$GPR),23(R8),14(TRNB),15(LOG),                E.2
*A*                C17(GPROUT),24(R0)                                         E.2
*A**                                                                          E.2
*A**       SETUP SECOND OP WORD FOR DIVIDE;                                   E.2
*A**       IF GOT OVERFLOW (ALUS) IN LAST SUM, TAKE QUICK EXIT                E.2
*A**                                                                          E.2
*A*S028    GEN     /T(S029,S038),5(FT),7(ALUS),11(B$GPR),23(R9),              E.2
*A*                C14(TRNB),15(LOG),17(GPROUT),24(R1)                        E.2
*A**                                                                          E.2
*A**       DIVIDE OP BY Y1                                                    E.2
*A**                                                                          E.2
*A*S029    GMSK    /P(PAGE+DIV),10(STACK),16(PUSH),                           E.2
*A*                C15(PAGE+S030),LB3,TF0,SF0                                 E.2
*A**                                                                          E.2
*A**       Y2 = (X/Y1)+Y1                                                     E.2
*A**                                                                          E.2
*A*S030    GEN     /N(S031),12(A$GPR),24(RC),11(B$GPR),23(R1),                E.2
*A*                C14(ADD),17(GPROUT)                                        E.2
*A**                                                                          E.2
*A**       Y2 = 1/2 (X/Y1 + Y1)                                               E.2
*A**                                                                          E.2
*A*S031    GEN     /N(S032),12(A$GPRR),SH4,24(RC),14(TRNA),15(LOG),           E.2
*A*                C17(GPROUT)                                                E.2
*A**                                                                          E.2
*A**       SETUP  OPERAND FOR SECOND DIVIDE                                   E.2
*A**                                                                          E.2
*A*S032    GEN     /N(S033),11(B$GPR),23(R8),14(TRNB),15(LOG),                E.2
*A*                C17(GPROUT),24(R0)                                         E.2
*A*S033    GEN     /N(S034),11(B$GPR),23(R9),14(TRNB),15(LOG),                E.2
*A*                C17(GPROUT),24(R1)                                         E.2
*A**                                                                          E.2
*A**       DIVIDE OP BY Y2 TO GET MS RESULT WORD                              E.2
*A**                                                                          E.2
*A*S034    GMSK    /P(PAGE+DIV),10(STACK),16(PUSH),                           E.2
*A*                C15(PAGE+S035),LB3,TF0,SF0                                 E.2
*A**                                                                          E.2
*A**       SAVE MS RESULT WORD                                                E.2
*A**                                                                          E.2
*A*S035    GEN     /N(S036),11(B$GPR),23(R1),14(TRNP),15(LOG),                E.2
*A*                C17(GPROUT),24(RB)                                         E.2
*A**                                                                          E.2
```

```
*A**      DIVIDE AGAIN BY Y2 TO GET LS RESULT WORD                          E.2
*A**                                                                        E.2
*A*S036   GEN    /N(S037),12(A$SPEC),22(AZERO),14(TRNA),15(LOG),           E.2
*A*             C17(GPROUT),24(R1)                                         E.2
*A*S037   GMSK   /P(PAGE+DIV),10(STACK),18(PUSH),                          E.2
*A*             C15(PAGE+S038),LB3,TF0,SF0                                 E.2
*A***                                                                       E.2
*A***     TRANSFER LS RESULT WORD TO OPR FOR SHIFTING,                      E.2
*A***     ( THIS MICRO ADDR MUST BE EVEN ).                                 E.2
*A***                                                                       E.2
*A*S038   GEN    /N(S039),12(A$GPR),24(R1),14(TRNA),15(LOG),13(OPROUT)      E.2
*A**                                                                        E.2
*A**      SETUP LEFT BYTE OF LS FOR COMBINATION WITH MS RESULT WORD         E.2
*A**                                                                        E.2
*A*S039   GEN    /N(S040),11(B$SPEC),23(OLSF),14(TRNB),15(LOG),            E.2
*A*             C17(GPROUT),24(R1)                                         E.2
*A**                                                                        E.2
*A**      GET MS RESULT BY SUM OF Y2 AND FIRST QUOTIENT,(QP/Y2),            E.2
*A**                                                                        E.2
*A*S040   GEN    /N(S041),12(A$GPR),24(RC),11(B$GPR),23(RR),               E.2
*A*             C14(ADD),17(GPROUT),13(OPROUT)                             E.2
*A**                                                                        E.2
*A**      INCLUSIVE OR LS AND RIGHT BYTE MS TO SIMULATE SHIFT 8,            E.2
*A**                                                                        E.2
*A*S041   GEN    /N(S042),12(A$GPRL),24(R1),11(B$SPEC),23(ORLZ),           E.2
*A*             C14(OR),17(GPROUT)                                         E.2
*A**                                                                        E.2
*A**      RIGHT SHIFT MS BY 8,(THROUGH BYTE SWAP),                          E.2
*A**                                                                        E.2
*A*S042   GEN    /N(S043),11(B$SPEC),23(OLSF),14(TRNB),15(LOG),            E.2
*A*             C13(OPROUT)                                                E.2
*A**                                                                        E.2
*A**      DOUBLE SHIFT OPR AND R1 TO COMPLETE 9 PLACE SHIFT                 E.2
*A**                                                                        E.2
*A*S043   GEN    /N(S044),12(A$GPRR),SH3,24(R1),14(TRNA),15(LOG),          E.2
*A*             C17(GPROUT),18(SHFTOP),20(RGHT),XF2                        E.2
*A**                                                                        E.2
*A**      CLEAR SIGN BIT OF LS RESULT WORD; RESTART PIPELINE,               E.2
*A**      ALLIGN MS FOR COMBINATION WITH EXPONENT .                         E.2
*A**                                                                        E.2
*A*S044   GEN    /N(S045),12(A$GPRR),SH4,24(R1),14(TRNA),15(LOG),          E.2
*A*             C17(GPROUT),10(TF$P),6(MEMC),18(SHFTOP),20(LFT)            E.2
*A**                                                                        E.2
```

```
*A**        COMBINE MS RESULT WITH EXPONENT                                  E.2
*A**                                                                         E.2
*A**S045    GEN     /N(S046),12(ASGPR),24(RA),11(BSSPEC),23(OR7F),           E.2
*A**            C14(OP),17(GPROUT)                                           E.2
*A**                                                                         E.2
*A**        CLEAR SIGN BIT                                                   E.2
*A**                                                                         E.2
*A**S046    GEN     /N(S047),12(ASGPR),SH4,24(RA),14(TRNA),15(LOG),          E.2
*A**            C17(GPROUT)                                                  E.2
*A**                                                                         E.2
*A**        SETUP RESULT AND EXIT BACK TO ROM                                E.2
*A**                                                                         E.2
*A**S047    GEN     /P(SS2M),10(PJMP),SF0,TF0,11(BSGPR),23(RA),              E.2
*A**            C14(TRNB),15(LOG),17(GPROUT),24(R0)                          E.2
*A**        EJFC                                                             E.2
*A**                                                                         E.2
*A**        INDEX AND PERFORM OPERATION                                      E.2
*A**                                                                         E.2
*A**        COMPUTES EFFECTIVE MEMORY ADDRESS FROM INDEX AND BASE;           E.2
*A**        STORES IT AS OPERAND ADDRESS OF FOLLOWING INSTRUCTION.           E.2
*A**                                                                         E.2
*A**        CALLING SEQUENCE:                                                E.2
*A**                                                                         E.2
*A**            BCS   TOP1                                                   E.2
*A**            DATA : INDEX ADDR                                            E.2
*A**            DATA : ARRAY BASE ADDR                                       E.2
*A**             -   : FOLLOWING INSTR , FIRST WORD                          E.2
*A**             -   : SECOND INSTR WORD (REPLACED BY EFFEC ADDR)            E.2
*A**                                                                         E.2
*A**    NOTE:   BOTH INDEX ADDRESS AND ARRAY BASE ADDRESS MAY                E.2
*A**            BE INDIRECT.                                                 E.2
*A**                                                                         E.2
*A**    REG USAGE:                                                           E.2
*A**            RA - INDEX VALUE STORAGE                                     E.2
*A**            OPR - EFFECTIVE ADDR STORAGE                                 E.2
*A**            RB - COPY OF OPR ON EXIT                                     E.2
*A**                                                                         E.2
*A**                                                                         E.2
*A**                                                                         E.2
*A**                                                                         E.2
*A**    WAIT FOR COMPLETION OF INDEX VALUE FETCH;                            E.2
*A**    START FETCH OF ARRAY BASE ADDRESS;                                   E.2
*A**                                                                         E.2
```

```
*A*IORG1  EQU   *              SAVE POSITION                          E.2
*A*       ORG   SQ1+1          PLACE MICRO IN FIELD SEL TABLE         E.2
*A*IOP3   GEN   /N(IOP4),6(MEMC),10(OFSP),13(INCP)                    E.2
*A**                                                                  E.2
*A**      SAVE INDEX VALUE; ADVANCE P TO FOLLOWING INSTR.             E.2
*A**                                                                  E.2
*A*       ORG   X'179                                                 E.2
*A*IOP4   GEN   /*,11(BSSPEC),23(MIR),14(DECB),12(ASSPEC),22(ADDFS),  E.2
*A*           C17(GPROUT),24(RA),13(INCP)                             E.2
*A**                                                                  E.2
*A**      WAIT FOR COMPLETION OF BASE ADDRESS FETCH                   E.2
*A**                                                                  E.2
*A*IOP5   GEN   /*,6(SPEC),10(WAITMD)                                 E.2
*A**                                                                  E.2
*A**      TEST IF BASE ADDRESS HAS INDIRECT BIT SET;                  E.2
*A**      IF SO, USE IT AS INDIRECT ADDRESS AND LOOP TILL END OF CHAIN. E.2
*A**      SET EFFECTIVE ADDR = BASE ADDR +(2*INDEX)                   E.2
*A**                                                                  E.2
*A*       GEN   /T(IOP5,IOP7),5(TT),7(MIRS),6(TESTT),10(OFSMIR),      E.2
*A*           C12(ASGPR),24(RA),11(BSSPEC),23(MIR),14(ADD),           E.2
*A*           C13(OPROUT)                                             E.2
*A**                                                                  E.2
*A**      START STORE OF EFFECTIVE ADDRESS INTO SECOND WORD           E.2
*A**      OF FOLLOWING INSTRUCTION.                                   E.2
*A**                                                                  E.2
*A*IOP7   GEN   /*,6(MEMC),10(OSSALU),12(ASP),14(INCA),16(CRY1)      E.2
*A**                                                                  E.2
*A**      SETUP EFFECTIVE ADDR FOR STORAGE;                           E.2
*A**      WAIT FOR STORE TO COMPLETE;                                 E.2
*A**      START FETCH OF FOLLOWING INSTR;                             E.2
*A**      EXIT THROUGH ROM STANDARD STATE                             E.2
*A**                                                                  E.2
*A*       GEN   /P(SS2M),7(PJMPS),6(MEMCS),TF0,10(TFSP),              E.2
*A*           C11(BSSPEC),23(OPR),14(TRNB),15(LOG),17(GPROUT),24(RB)  E.2
*A*       ORG   IORG1          RESTORE LOCATION                       E.2
*A*       FJFC                                                        E.2
*A**                                                                  E.2
*A**      FOLLOWING IS FIELD SELECT TABLE USED BY BCS X'17;           E.2
*A**      CHOOSES BETWEEN SDD PROCESSING AND MATH/SQRT/INDEX;         E.2
*A**                                                                  E.2
*A**                                                                  E.2
*A**      ENTRY FOR SDD WITH INCREMENT = 1                            E.2
*A**      SET RF = 1, ENTER INTO DO LOOP.                             E.2
```

```
*A**                                                                    E.2
*A*$DOIN1 GEN     /N(VALUE),12(ASSPEC),22(AZERO),14(INCA),16(CRY1),     E.2
*A*               C17(GPROUT),24(RF),10(WAITMD),6(SPEC)                 E.2
*A**                                                                    E.2
*A**      FIRST MICRO - MATH/SORT/INDEX.                                E.2
*A**      BEGIN FETCH OF SECOND OPERAND                                 E.2
*A**      FIELD SELECT TO CHOOSE MULT/DIVIDE OR OTHERS                  E.2
*A**                                                                    E.2
*A*       GEN     /F(IAR1),10(OFSMTR),6(MEMC),FSA,2(1)                 E.2
```

```
   .REPLACE,150,151                V$LPD24   NO SMR    04/03/75  KERNS      E.2
*D*D321    DATA     0321           SINGLE STEP COMMAND
*D*SMK2    DATA     0301           FOR PAPER FEED COMMAND,
*A*D321    DATA     012            PAPER FEED COMMAND                        E.2
*A*SMK2    DATA     0300           BIT FOR REQ WD, ALSO TO INDICATE A        E.2
```

```
   .REPLACE,505
*D*        JAZ      OASN7        DUMMY
*A*        JANZ     OASN31       IF NOT DUMMY
*A*        STAE     ONSR98       SET LUN = 0
*A*        JMP      OASN7
```

```
    .REPLACE,667                    V$OABORT  SMR-572   03/13/75  SLIPSON   E.2
*D*         ANAI    0177400
*A*         ANA     LHW                                                     E.2
    .REPLACE,706                    V$OABORT  SMR-572   03/13/75  SLIPSON   E.2
*D*UNUMG    SUBI    020             0260
*A*UNUMG    SUB     B54             =020    0260                            E.2
```

```
   ,REPLACE,410                    V$OTIME  SMR-572  03/13/75  SLIPSON  E.2
*D*      SUBI    13               RETURN A=BINARY
*A*      SUB     ODAT4+1          RETURN A=BINARY                     E.2
   ,REPLACE,414                    V$OTIME  SMR-572  03/13/75  SLIPSON  E.2
*D*      SUBI    19               CHECK IF LESS THAN 31 DAYS
*A*      SUB     ODAT5+1          CHECK IF LESS THAN 31 DAYS          E.2
   ,REPLACE,416                    V$OTIME  SMR-572  03/13/75  SLIPSON  E.2
*D*      LDBI    01
*A*      LDB     ONE                                                  E.2
   ,REPLACE,425                    V$OTIME  SMR-572  03/13/75  SLIPSON  E.2
*D*      SUBI    13               CHECK TO SEE IF IN BOUNDS FOR MONTHS
*A*UDAT4 SUBI    13               CHECK TO SEE IF IN BOUNDS FOR MONTHS E.2
   ,REPLACE,432                    V$OTIME  SMR-572  03/13/75  SLIPSON  E.2
*D*      SUBI    19
*A*ODAT5 SUBI    19                                                   E.2
   ,REPLACE,486,487                V$OTIME  SMR-572  03/13/75  SLIPSON  E.2
*D*      JXZ     ODEV6            ALL DONE
*D*      JMP     ODEV4-1          DO NEXT
*A*      JXNZ    ODEV4-1          DO NEXT IF NOT DONE                 E.2
   ,REPLACE,573,574                V$OTIME  SMR-572  03/13/75  SLIPSON  E.2
*D*      TBA
*D*      IAR
*A*      INCR    021              TBA IAR                             E.2
   ,REPLACE,589                    V$OTIME  SMR-572  03/13/75  SLIPSON  E.2
*D*      ANAI    0177400
*A*      ANA     LHW                                                  E.2
   ,REPLACE,661                    V$OTIME  SMR-572  03/13/75  SLIPSON  E.2
*D*      DIVI    10               A=REMAIDER, B= QUOTIENT
*A*      DIV     TEN              A=REMAINDER, B=QOUTIENT             E.2
   ,REPLACE,768                    V$OTIME  SMR-572  03/13/75  SLIPSON  E.2
*D*ONUMG SUBI    020              0260
*A*ONUMG SUB     BS4              0260   (BS4=020)                    E.2
   ,REPLACE,770                    V$OTIME  SMR-572  03/13/75  SLIPSON  E.2
*D*      SUBI    10
*A*      SUB     TEN                                                  E.2
   ,REPLACE,782,783                V$OTIME  SMR-572  03/13/75  SLIPSON  E.2
*D*      JXZ     ONUMB            ALL DIGITS CONVERTED TO BCD
*D*      JMP     ONUMA
*A*      JXNZ    ONUMA            JUMP IF ALL DIGITS NOT CONVERTED TO BCD E.2
```

```
    .INSERT,716                     MIUTIL    NO SMR    02/03/75  KERNS     F
*A*E114A  EQU     *                                                         F
    .REPLACE,2328                   MIUTIL    NO SMR    01/31/75  KERNS     F
*D*       JMP     *+4
*A*       STA     INHA                                                      F
*A*       SUBI    '='                                                       F
*A*       JAZ     INH6      IF ZERO BEFORE STORE FIELD VALUE                F
*A*       SUBI    '+'-'='                                                   F
*A*       JAZ     INH7      IF DO NOT ZERO BEFORE STORE FIELD VALUE         F
*A*       LDA     INHA                                                      F
*A*       JMP     INH4      CONTINUE PROCESSING                             F
    .INSERT,2331                    MIUTIL    NO SMR    01/31/75  KERNS     F
*A*INH4   EQU     *                                                         F
    .INSERT,2369                    MIUTIL    NO SMR    01/31/75  KERNS     F
*A*INH6   LDBI    MBUF      ROM WORD ADDR                                   F
*A*       TZA                                                               F
*A*       STA     0,2       ZERO                                            F
*A*       STA     1,2       ENTIRE                                          F
*A*       STA     2,2       ROM                                             F
*A*       STA     3,2       WORD                                            F
*A*INH7   JMP     FLDCG     PROCESS FIELD CHANGE DIRECTIVE                  F
    .INSERT,2379                    MIUTIL    NO SMR    01/31/75  KERNS     F
*A*       EJEC                                                              F
*A**      THIS IS A SUBROUTINE TO HANDLE THE CHANGE CENTRAL CONTROL         F
*A**      STORE BY FIELD SPECIFICATION                                      F
*A**                                                                        F
*A**      CALLING SEQUENCE                                                  F
*A**      JMP     FLDCG                                                     F
*A**                                                                        F
*A**      RETURNS TO SIMULATOR EXEC FOR PROCESSING OF NEXT DIRECTIVE        F
*A**                                                                        F
*A*FLDCG  EQU     *                                                         F
*A*       JMPM    FETCHA                                                    F
*A*FLD05  LRLA    8                                                         F
*A*       STA     FLD90     SAVE LEFT BYTE                                  F
*A*       JMPM    FETCHA                                                    F
*A*       ERA     FLD90     FORM 2 CHAR NAME                                F
*A*       STA     FLD90                                                     F
*A*       LDXI    FLNAM     FIELD NAME TABLE                                F
*A*FLD10  LDA     0,1       FIELD NAME                                      F
*A*       JAZ     FLD60     IF END OF TABLE                                 F
*A*       ERA     FLD90                                                     F
*A*       JAZ     FLD20     IF MATCH FOUND                                  F
```

```
*A*          IXR                                                    F
*A*          IXR                                                    F
*A*          IXR                                                    F
*A*          JMP     FLD10       CONTINUE SCAN THROUGH TABLE        F
*A*FLD20     EQU     *                                              F
*A*          STX     FLD92       SAVE TABLE PTR                     F
*A*          JMPM    INA         FETCH CHANGE VALUE                 F
*A*          DATA    'N'                                            F
*A*          STX     FLD93       SAVE TERM CHAR                     F
*A*          STA     FLD91       SAVE VALUE                         F
*A*          LDX     FLD92       FIELD TABLE PTR                    F
*A*          LDB     1,X         FIELD SIZE                         F
*A*          LDA     FLD25                                          F
*A*          ANAI    07740                                          F
*A*          MERGE   031         FORM RIGHT SHIFT                   F
*A*          STA     FLD25                                          F
*A*          LDA     FLD91       VALUE                              F
*A*FLD25     LSRA    0                                              F
*A*          JAZ     FLD30       IF VALUE WITHIN BOUNDS             F
*A*          JMP     FLD65       RANGE ERROR                        F
*A*FLD30     LDA     2,X         LOW ORDER BIT POSITION             F
*A*          LSRA    4                                              F
*A*          CPA                                                    F
*A*          ANAI    3                                              F
*A*          ADDI    MBUF        FETCH 16 BIT WORD ADDR             F
*A*          STA     FLD95                                          F
*A*          LDA     2,X                                            F
*A*          TAB                                                    F
*A*          SUBI    31          CHECK IF IM FIELD                  F
*A*          JAZ     FLD50       YES, SPECIAL PROCESSING            F
*A*          SUBI    15          CHECK IF FS FIELD                  F
*A*          JAZ     FLD50A      YES, SPECIAL PROCESSING            F
*A*          TBA                 NEITHER IM NOR FD, RESTORE VALUE   F
*A*          ANAI    017         WORD DISPLACEMENT                  F
*A*          TAB                                                    F
*A*          LDA     FLD35                                          F
*A*          ANAI    07740                                          F
*A*          MERGE   031                                            F
*A*          STA     FLD35       FORM LEFT ROTATE                   F
*A*          LDB     1,X         FIELD SIZE                         F
*A*          LDAE    FLMSK,B     BIT MASK FOR FIELD SIZE            F
*A*FLD35     LRLA    0           MOVE MASK INTO POSITION            F
*A*          CPA                                                    F
```

```
*A*          LDB     FLD95                                          F
*A*          ANA     0,B             MASK OFF BITS FROM WORD        F
*A*          TAB                                                    F
*A*          LDA     FLD91           NEW FIELD VALUE                F
*A*          XEC     FLD35           MOVE VALUE INTO POSITION       F
*A*          MERGE   031             UPDATE WORD                    F
*A*          LDB     FLD95           ROM WORD ADDR                  F
*A*          STA     0,B                                            F
*A*FLD39     EQU     *                                              F
*A*          CALL    WCS             OUTPUT WORD TO WCS             F
*A*          LDA     FLD93           TERMINATING CHAR               F
*A*          SUBI    ','                                            F
*A*          JAZ     FLD40           IF MORE CHARS IN RECORD        F
*A*          JMP     EXC10           GET NEW DIRECTIVE              F
*A*FLD40     EQU     *                                              F
*A*          JMPM    FETCHA          FETCH NEXT CHAR                F
*A*          TAB                                                    F
*A*          SUBI    ' '                                            F
*A*          JAZ     E114A           IF DISPLAY NEXT WORD           F
*A*          TBA                                                    F
*A*          JMP     FLD05           CONTINUE DIRECTIVE PROCESSING  F
*A*FLD50A    EQU     *               SPECIAL PROCESSING FOR FS FIELD F
*A*          LDBI    2               SET FOR FS FIELD ROTATE        F
*A*          JMP     FLD50B                                         F
*A*FLD50     EQU     *               SPECIAL PROCESSING FOR IM FIELD F
*A*          INCR    02              SET FOR IM FIELD ROTATE        F
*A*FLD50B    EQU     *                                              F
*A*          STB     FLD51           FIELD FLAG                     F
*A*          TBA                                                    F
*A*          ADDI    16              SET UP LLRL INST               F
*A*          ADD     FLLRL                                          F
*A*          STA     FLD56           ROTATE INTO A REG LSBS         F
*A*          LDAI    16                                             F
*A*          SUBI    0               SET UP RESTORE ROTATE          F
*A*FLD51     BES     0                                              F
*A*          ADD     FLLRL           LLRL INST                      F
*A*          STA     FLD57                                          F
*A*          LDX     FLD95           CCS WORD SUB ADDR              F
*A*          LDA     0,X             LSB PORTION                    F
*A*          DXR                     POINT TO NEXT HIGH ORDER BITS  F
*A*          LDB     0,X             MSB PORTION                    F
*A*FLD56     LLRL    0               POSITION FIELD INTO A REG LSBS F
*A*          ANAI    0177760         MASK OUT FIELD                 F
```

```
*A*         ADD     FLD91           CHANGE VALUE                                F
*A*FLD57    LLRL    0               RETURN FIELDS TO PROPER POSITION            F
*A*         STB     0,X                                                         F
*A*         STA     1,X                                                         F
*A*         JMP     FLD39           CONTINUE FIELD PROCESSING                   F
*A**                                                                            F
*A*FLLRL    LLRL    0               LLRL INSTRUCTION BASE VALUE                 F
*A*FLD90    DATA    0               FIELD NAME                                  F
*A*FLD91    DATA    0               NEW FIELD VALUE                             F
*A*FLD92    DATA    0               NAME TABLE POINTER                          F
*A*FLD93    DATA    0               TERM CHAR                                   F
*A*FLD94    DATA    0                                                           F
*A*FLD95    DATA    0               ROM WORD SUBCOMP ADDR                       F
*A**                                                                            F
*A*FLD60    CALL    SIOUT,3,FLER1     OUTPUT FIELD NAME ERROR                   F
*A*         JMP     EXC10           TRY AGAIN                                   F
*A*FLD65    CALL    SIOUT,3,FLER2     OUTPUT FIELD RANGE ERROR                  F
*A*         JMP     EXC10           TRY AGAIN                                   F
*A*FLER1    DATA    '  MU16'                                                    F
*A*FLER2    DATA    '  MU17'                                                    F
*A**                                                                            F
*A**  FIELD NAME TABLE                                                         F
*A**        3 WORD ENTRY                                                        F
*A**                1=2 CHAR NAME                                               F
*A**                2=FIELD SIZE                                                F
*A**                3=LOW ORDER BIT POSITION                                    F
*A**                                                                            F
*A*FLNAM    EQU     *                                                           F
*A*         DATA    'AA',4,0                                                    F
*A*         DATA    'BB',4,4                                                    F
*A*         DATA    'SH',3,8                                                    F
*A*         DATA    'XF',2,11                                                   F
*A*         DATA    'WF',1,13                                                   F
*A*         DATA    'VF',1,14                                                   F
*A*         DATA    'SC',1,15                                                   F
*A*         DATA    'WR',1,16                                                   F
*A*         DATA    'CF',2,17                                                   F
*A*         DATA    'MF',1,19                                                   F
*A*         DATA    'FF',4,20                                                   F
*A*         DATA    'RF',3,24                                                   F
*A*         DATA    'LA',2,27                                                   F
*A*         DATA    'LB',2,29                                                   F
*A*         DATA    'IM',4,31                                                   F
```

```
*A*         DATA    'AB',2,35                                          F
*A*         DATA    'MR',1,37                                          F
*A*         DATA    'GF',4,38                                          F
*A*         DATA    'SF',2,42                                          F
*A*         DATA    'TF',2,44                                          F
*A*         DATA    'FS',4,46                                          F
*A*         DATA    'MT',1,50                                          F
*A*         DATA    'MS',4,51                                          F
*A*         DATA    'AF',5,55                                          F
*A*         DATA    'TS',4,60                                          F
*A*         DATA    0               END OF TABLE                       F
*A**                                                                   F
*A*FLMSK    EQU     *               BIT MASK TABLE                     F
*A*         DATA    0                                                  F
*A*         DATA    01                                                 F
*A*         DATA    03                                                 F
*A*         DATA    07                                                 F
*A*         DATA    017                                                F
*A*         DATA    037                                                F
```

```
   .INSERT,661                    V$JCP     SMR-499   09/20/74  NEWDORF   E.1
*A*JCPRE  LDA     V$JCB          CALCULATE CHAR ADDRESS OF LAST + 1      E.1
*A*       ADDI    40             CHAR IN JC BUF                          E.1
*A*       LRLA    1              AND STORE IT                            E.1
*A*       STA     BUFLIM         INTO BUFLIM.                             E.1
*A*       LDB     CHPTR          SAVE CHPTR IN SAVECP.                   E.1
*A*       STB     SAVECP                                                 E.1
*A*       IAR                                                            E.1
*A*       ERA     CHPTR          IS CHPTR AT END OF BUFFER?            E.1
*A*       JAZ     JCLAB5         IF YES, JUMP TO END OF ROUTINE.         E.1
*A*       LDA     PERPTR         WHEN PERPTR IS ZERO, WE SCAN FOR "."    E.1
*A*       JAZ     JCLAB1         IF NOT, WE ALREADY KNOW WHERE "." IS.   E.1
*A*       STA     CHPTR                                                  E.1
*A*       TZA                                                            E.1
*A*       STA     PERPTR         REINITIALIZE PERPTR.                    E.1
*A*JCLAB1 JSR     JCGTCH,X       IS NEXT CHAR A PERIOD?                  E.1
*A*       SUB     N256                                                   E.1
*A*       JANZ    JCLAB4         IF NOT, JUMP                            E.1
*A*       LDA     BUFLIM         CALCULATE LAST WORD ADDRESS OF          E.1
*A*       LSRA    1              JCB AND STORE                           E.1
*A*       DAR                                                            E.1
*A*       STA     BUFLIM         IT BACK INTO BUFLIM                     E.1
*A*       LDX     N12024         TWO SPACES.                              E.1
*A*       LDB     CHPTR                                                  E.1
*A*       DBR                                                            E.1
*A*       TBA                                                             E.1
*A*       LSRB    1              COMPUTE WORD ADDRESS.                    E.1
*A*       BT      RA0+B0,JCLAB2  IS PERIOD IN LEFT BYTE?                  E.1
*A*       LDA     0,B            NO.                                     E.1
*A*       ANA     LHW            REPLACE PERIOD BY                        E.1
*A*       ORAI    0240           A BLANK.                                 E.1
*A*       STA     0,B            STORE BACK INTO JCB.                    E.1
*A*       JMP     JCLAB3                                                  E.1
*A*JCLAB2 STX     0,B            STORE BLANKS INTO JCB.                  E.1
*A*JCLAB3 TBA                                                            E.1
*A*       SUB     BUFLIM         ARE WE AT END OF BUFFER?                E.1
*A*       JAZ     JCLAB5         IF YES, JUMP TO JCLAB5.                 E.1
*A*       IBR                    IF NOT, INCREMENT B AND LOOP.           E.1
*A*       JMP     JCLAB2                                                 E.1
*A*JCLAB4 LDA     CHPTR          THIS CHAR NOT A PERIOD.                 E.1
*A*       SUB     BUFLIM                                                 E.1
*A*       JANZ    JCLAB1         LOOK AT NEXT CHAR.                      E.1
*A*JCLAB5 LDA     SAVECP         RESTORE CHPTR.                          E.1
```

```
*A*         STA     CHPTR                                                   E.1
   .REPLACE,662                 V$JCP     SMR-499   09/20/74  NEWDORF    E.1
*D*JCPRE    LDB     JCPRB
*A*         LDB     JCPRB                                                   E.1
   .INSERT,664                  V$JCP     SMR-499   09/20/74  NEWDORF    E.1
*A*BUFLIM   DATA    0           STORES END OF JCB.                          E.1
*A*SAVECP   DATA    0           SAVES CHPTR WHILE WE LOOK FOR PERIOD        E.1
   .INSERT,687                  V$JCP     NO SMR    02/25/75  KERNS      E.2
*A*         LDA     FIVE        LO                                          E.2
*A*         JSR     JCLNDV,X    GET DST ENTRY ADDR                          E.2
*A*         TAB                                                             E.2
*A*         LDA     1,B         DEVICE NAME                                 E.2
*A*         LSRA    8                                                       E.2
*A*         SKE     N304,7,010  IS LO AN RMD                                E.2
*A*         JMP     JCP1        IF NOT AN RMD                               E.2
*A*         LDB     JCFCBS+3    LOFCB ADDR                                  E.2
*A*         LDA     3,B                                                     E.2
*A*         DAR                                                             E.2
*A*         JAN     JCP1        IF FILE NOT OPEN, DO NOT CLOSE              E.2
*A*         STB     *+5         STORE FCB ADDR IN CALL                      E.2
*A*         CLOSE   LOFCB,LO,0,1 CLOSE 'LO' FILE WITH WAIT/UPDATE           E.2
*A*JCP1     EQU     *                                                       E.2
*A*         LDA     SEVEN       80                                          E.2
*A*         JSR     JCLNDV,X    GET DST ENTRY ADDR                          E.2
*A*         TAB                                                             E.2
*A*         LDA     1,B         DEVICE NAME                                 E.2
*A*         LSRA    8                                                       E.2
*A*         SKE     N304,7,010  IS 80 AN RMD                                E.2
*A*         JMP     JCP2        IF NOT AN RMD                               E.2
*A*         LDB     JCFCBS+5    80FCB ADDR                                  E.2
*A*         LDA     3,B                                                     E.2
*A*         DAR                                                             E.2
*A*         JAN     JCP2        IF FILE NOT OPEN                            E.2
*A*         STB     *+5         STORE FCB ADDR IN CALL                      E.2
*A*         CLOSE   80FCB,7,0,1 CLOSE '80' FILE WITH WAIT/UPDATE            E.2
*A*JCP2     EQU     *                                                       E.2
   .REPLACE,896                 V$JCP     SMR-805   02/24/75  NEWDORF    E.2
*D*         SUB     NINE        NUMBER
*A*         SUB     TEN         NUMBER                                      E.2
   .REPLACE,898                 V$JCP     SMR-805   02/24/75  NEWDORF    E.2
*D*         SUB     EIGHT
*A*         SUB     SEVEN                                                   E.2
   .INSERT,942                  V$JCP     NO SMR              KERNS      E.2
```

```
*A**  THE A REG MUST BE ZERO AT THE TIME OF THE SCHED REQ -REQUIRED BY  E.2
*A**  VSORT-                                                            E.2
   .REPLACE,1286                 V$JCP    SMR-754   02/26/75  KERNS    E.2
*D*       LDAE    JCFCBS,B      PICKUP ADDRESS OF APPROPRIATE
*A*       LDAE    JCFCBS-2,B    PICKUP ADDR OF APPROPRIATE              E.2
   .INSERT,1776                  V$JCP    SMR-499   09/20/74  NEWDORF  E.1
*A*       LDA     CHPTR         SAVE POSITION OF                        E.1
*A*       DAR                   PERIOD IN JC BUFFER                     E.1
*A*       STA     PERPTR        IN PERPTR.                              E.1
   .INSERT,1785                  V$JCP    SMR-499   09/20/74  NEWDORF  E.1
*A*PERPTR DATA    0             SAVES CHPTR TO PERIOD IN JCB.           E.1
```

```
   .REPLACE,1024                    VSFMAIN   SMR-732   11/08/74  NEWDORF  E.1
*D*        JMP      FMCRX
*A*        JMP      FMCRX2                                                   E.1
   .REPLACE,1189                    VSFMAIN   SMR-732   11/08/74  NEWDORF  E.1
*D*        LDA      FMCREF     LOAD A WITH ERROR FLAG AT EXIT
*A*FMCRX2  LDA      FMCREF     LOAD A WITH ERROR FLAG                        E.1
   .REPLACE,1261                    VSFMAIN   SMR-732   11/08/74  NEWDORF  E.1
*D*        JMP      FMDEX
*A*        JMP      FMDEX2                                                   E.1
   .REPLACE,1423                    VSFMAIN   SMR-732   11/08/74  NEWDORF  E.1
*D*        JSR      FMLF,X     LINE FEED LO
*A*FMDEX2  JSR      FMLF,X     LINE FEED LO                                  E.1
   .REPLACE,1432                    FMAIN     SMR-732   05/14/75  KERNS     E.2
*D*        JMP      FMDEX
*A*        JMP      FMDEX2                                                   E.2
   .REPLACE,1582                    VSFMAIN   SMR-732   11/08/74  NEWDORF  E.1
*D*        JMP      FMENX
*A*        JMP      FMENX2                                                   E.1
   .REPLACE,1670                    VSFMAIN   SMR-732   11/08/74  NEWDORF  E.1
*D*        LDA      FMENEF     LOAD A WITH ERROR FLAG
*A*FMENX2  LDA      FMENEF     LOAD A WITH ERROR FLAG                        E.1
   .REPLACE,2401                    VSFMAIN   SMR-732   11/08/74  NEWDORF  E.1
*D*        JMP      FMLIX
*A*        JMP      FMLIX2                                                   E.1
   .REPLACE,2518                    VSFMAIN   SMR-732   11/08/74  NEWDORF  E.1
*D*        JSR      FMLF,X     LINE FEED LO
*A*FMLIX2  JSR      FMLF,X     LINE FEED ON LO                        E.1
   .INSERT,2835                     VSFMAIN   NO SMR    02/25/75  KERNS     E.2
*A*        LDAI     4096                                                     E.2
*A*        STA      FMN1                                                     E.2
   .INSERT,2843                     VSFMAIN   NO SMR    02/25/75  KERNS     E.2
*A*        LDAI     3277                                                     E.2
*A*        STA      FMN1                                                     E.2
   .INSERT,2854                     VSFMAIN   NO SMR    02/25/75  KERNS     E.2
*A*FMN1    BES      0                                                        E.2
   .REPLACE,3153                    VSFMAIN   SMR-732   11/08/74  NEWDORF  E.1
*D*        JMP      FMREX
*A*        JMP      FMREX2                                                   E.1
   .REPLACE,3209                    VSFMAIN   SMR-732   11/08/74  NEWDORF  E.1
*D*        LDA      FMREEF     LOAD A WITH ERROR FLAG AT EXIT
*A*FMREX2  LDA      FMREEF     LOAD A WITH ERROR FLAG                        E.1
```

```
   ,INSERT,887                          V$IOUTIL  SMR-692   09/27/74  KERNS     E,1
*A*        LDA     WKMAC+3                                                      E,1
*A*        LRLA    4                                                            E,1
*A*        ANA     BM3         ISOLATE MODE                                     E,1
*A*        DAR                                                                  E,1
*A*        JANZ    IUCPS       IF NOT ASCII MODE                                E,1
   ,REPLACE,2012,2023                   V$IOUTIL  NO SMR    09/27/74  KERNS     E,1
*D*V$DVTP  STB     DVTPA
*D*        ADD     V$LUT1
*D*        TAB
*D*        LDA     0,B         GET CURRENT DST ENTRY NO,
*D*        ANA     RHW
*D*        DAR                 CONVERT
*D*        STA     DVTPB       TO
*D*        ASLA    1           DISPLACEMENT
*D*        ADD     DVTPB       IN TABLE
*D*        ADD     V$DSTB      ADD TABLE BASE
*D*        TAB
*D*        LDA     1,B         GET DEVICE NAME
*A*V$DVTP  STB     DVTPA                                                        E,1
*A*        STX     DVTPX       SAVE RETURN                                      E,1
*A*        TAB                                                                  E,1
*A*        TZX                                                                  E,1
*A*        SUBE    DVLUN,X                                                      E,1
*A*        JAN     DVTP5       IF IN LUT 1                                      E,1
*A*        IXR                                                                  E,1
*A*        SUBE    DVLUN,X                                                      E,1
*A*        JAN     DVTP5       IF IN LUT 2                                      E,1
*A*        IXR                                                                  E,1
*A*DVTP5   EQU     *           LUN                                              E,1
*A*        ADDE    DVLUN,X     ADJUST FOR LUT ENTRY                             E,1
*A*        LDBE    DVLUT,X                                                      E,1
*A*        ADD     0,B         BASE ADDR OF LUT                                 E,1
*A*        TAB                                                                  E,1
*A*        LDA     0,B         CURRENT DST ENTRY NU,                            E,1
*A*        ANA     RHW                                                          E,1
*A*        DAR                                                                  E,1
*A*        STA     DVTPB                                                        E,1
*A*        ASLA    1                                                            E,1
*A*        ADD     DVTPB       TABLE DISPLACEMENT                               E,1
*A*        ADD     V$DSTB      ADD TABLE BASE                                   E,1
*A*        TAB                                                                  E,1
*A*        LDA     1,B         DEVICE NAME                                      E,1
```

```
*A*         SUB     DVTPSP                                                  E.1
*A*         JANZ    DVTP10          IF NOT A SPOOL UNIT                     E.1
*A*         LDA     0,B             DEVICE NUMBER (BITS 12-4)               E.1
*A*         LSRA    10              POSITION MSB OF UNIT NUMBER             E.1
*A*         ANA     BM7                                                     E.2
*A*         ADD     V$LUT3          SPOOL LUT BASE                          E.1
*A*         INCR    012             ADJUST PAST LUT SIZE WORD               E.2
*A*         LDA     0,B             CURRENT DST ENTRY NO.                   E.1
*A*         ANA     RHW                                                     E.1
*A*         DAR                                                             E.1
*A*         STA     DVTPB                                                   E.1
*A*         ASLA    1                                                       E.1
*A*         ADD     DVTPB           TABLE DISPLACEMENT                      E.1
*A*         ADD     V$DSTB          ADD TABLE BASE                          E.1
*A*         TAB                                                             E.1
*A*DVTP10   LDA     1,B             DEVICE NAME                             E.1
   .INSERT,2028                     V$IOUTIL  NO SMR    09/27/74  KERNS    E.1
*A*         LDX     DVTPX           RETURN ADDR                             E.1
   .INSERT,2033                     V$IOUTIL  NO SMR    09/27/74  KERNS    E.1
*A*DVTPSP  DATA    'SP'                                                    E.1
*A*DVTPC   DATA    0                                                       E.1
*A*DVTPX   DATA    0                                                       E.1
*A*DVLUN   DATA    101             LUN ADJUSTMENT TABLE                    E.1
*A*         DATA    79                                                      E.1
*A*         DATA    0                                                       E.1
*A*DVLUT   DATA    V$LUT1          LUT TABLE BASE ADDR                     E.1
*A*         DATA    V$LUT2                                                  E.1
*A*         DATA    V$LUT3                                                  E.1
```

```
    .INSERT,243                     VSORT1          10/17/74  MENG      E.1
*A*         LDA     4,B             GET SHORT SEQ COUNT                 E.1
*A*         ADD     2,B             PLUS DUMMYS                         E.1
*A*         JAZ     MG8             JUMP IF ZERO                        E.1
    .INSERT,269                     VSORT1          10/17/74  MENG      E.1
*A*         LDA     4,B             GET SHORT SEQ COUNT                 E.1
*A*         ADD     2,B             PLUS DUMMYS                         E.1
*A*         JAZ     MG12            JUMP IF ZERO                        E.1
    .REPLACE,422                    VSORT1          06/25/75  MENG      F
*D*MGND     CALL    GFCLS       CLOSE OUTPUT FILE
*A*MGND     LDXI    SRTC                                                F
*A*         LDA     SWCH,X          GET SWITCH                          F
*A*         BT      RA1+9,FMOM      FINISH UP MERGE OMIT FILE           F
*A*MGNO     EQU     *                                                   F
    .REPLACE,524                    VSORT1          10/15/74  MENG      E.1
*D*         JMP  MR13           JUMP IF NOT EOS
*A*         JAP     MR13            JUMP IF .GT. 0                      E.1
    .INSERT,605                     VSORT1          06/27/75  MENG      F
*A*         LDA     SWCH,X          GET SWITCH                          F
*A*         BT      RA1+9,*+4       SKIP WRITE IF FILE OF TAGS          F
    .INSERT,613                     VSORT1          06/29/74  MENG      E.1
*A*         LDAE    MR13+2          GET RETURN CODE                     E.1
*A*         JAN     MMEG            INVALID END OF FILE                 E.1
    .REPLACE,682                    VSORT1          07/09/74  MENG      E.1
*D*         SUB     BR15
*A*         IAR                                                         E.1
    .INSERT,859                     VSORT1          06/20/75  MENG      F
*A*         LDA     SWCH,X          GET SWITCH                          F
*A*         BT      RA1+9,MOMT      BUILD FILE OF TAGS                  F
    .INSERT,871                     VSORT1          06/26/75  MENG      F
*A*         EJEC                                                        F
*A**                                                                    F
*A**                                                                    F
*A** MOMT - BUILD LIST OF TAGS ON THE OUTPUT FILE                       F
*A**                                                                    F
*A**                                                                    F
*A*MOMT     EQU     *                                                   F
*A*         LDBI    FCBO                                                F
*A*         LDA     1,B             OUTPUT ADDRESS                      F
*A*         ADD     CWC             PLUS CURRENT OUTPUT WORD            F
*A*         STA     MOMTA                                               F
*A*         LDAE*   MRRI            RECORD NUMBER                       F
*A*         STAE    000                                                 F
```

```
*A*MOMTA  EQU   *-1                                                    F
*A*       INR   CWC          BUMP CURRENT WORD                         F
*A*       LDA   CWC                                                    F
*A*       SUB   Q,8          TEST FOR END                              F
*A*       JAZ   MOMT1                                                  F
*A*MOMTX  INRE  MRRI         STEP RETURN                               F
*A*       JMP*  MRRI         -EXIT-                                    F
*A*MOMT1  CALL  MRCB         WRITE TAGS                                F
*A*       TZA                ZERO                                      F
*A*       STA   CWC          WORD COUNTER                              F
*A*       JMP   MOMTX        -EXIT-                                    F
*A**                                                                   F
*A** FINISH UP FILE OF TAGS                                            F
*A**                                                                   F
*A*FMOM   EQU   *                                                      F
*A*       LDA   CWC          RECORD OUT?                               F
*A*       JAZ   FMOM1        -YES-                                     F
*A*       CALL  MRRI,0       PUT TAG OF ZERO                           F
*A*       JMP   FMOM         LOOP                                      F
*A*FMOM1  CALL  OFCLS        CLOSE OUTPUT FILE WITH FILE MARK          F
*A*       JMP   MGNO                                                   F
*A**                                                                   F
*A*CWC    DATA  0            CURRENT WORD COUNTER                      F
   .INSERT,896               VSORT1          06/29/74  MENG           E,1
*A*MMEG   LDAI  '07'         MAKE ST07                                 E,1
*A*       STA   MER3+2                                                 E,1
   .REPLACE,953              VSORT1          06/29/74  MENG           E,1
*D*MRER7  JMP*  MRDB
*A*MRER7  LDAE  MRD2+2                                                 E,1
*A*       STAE  MA2+2                                                  E,1
*A*       JMP   MMEG                                                   E,1
   .INSERT,1113              VSORT1          10/21/74  MENG           E,1
*A*       JAZ   SCMA         JUMP IF NO DUMMYS                         E,1
   .INSERT,1157              VSORT1          10/18/74  MENG           E,1
*A*SCMA   WRITE SE04,5,0,1                                             E,1
*A*       EXIT                                                         E,1
*A*SE04   DCB   3,ST04,0                                               E,1
*A*ST04   DATA  '  ST04'                                               E,1
   .REPLACE,1234             VSORT1   SMR844 05/19/75  MENG           F
*D*       DATA  (CTLS+7*4) PTR TO CTL FLD 4 CMP PARM      SET BY ASMB
*A*       DATA  (CTLS+7*3)   PTR TO CTL FLD 4 CMP PARM                 F
   .INSERT,1296              VSORT1          06/13/75  MENG           F
*A**                BIT 9 ON, SKIP FINAL MERGE                        F
```

```
   .REPLACE,1435,1438              VSORT1                06/25/75  MENG      F
*D*CJP2    JMP   C0S0          TARGET BYTE 0,SOURCE BYTE 0
*D*        JMP   C0S1          TARGET BYTE 0,SOURCE BYTE 1
*D*        JMP   C1S0          TARGET BYTE 1,SOURCE BYTE 0
*D*        JMP   C1S1          TARGET BYTE 1,SOURCE BYTE 1
*A*CJP2    JMP      C0S0A          TARGET BYTE 0, SOURCE BYTE 0              F
*A*        JMP      C0S1A          TARGET BYTE 0, SOURCE BYTE 1              F
*A*        JMP      C1S0A          TARGET BYTE 1, SOURCE BYTE 0              F
*A*        JMP      C1S1A          TARGET BYTE 1, SOURCE BYTE 1              F
   .INSERT,1556                    VSORT1                06/25/75  MENG      F
*A**       ALGEBRAIC COMPARE                                                 F
*A*C0S0A   LDAE*    TW,X           GET TARGET WORD                           F
*A*        ERAE*    SW,X           EOR SOURCE WORD                           F
*A*        JAP      C0S0           SAME SIGN                                 F
*A*TWSIN   LDAE*    TW,X           GET TARGET WORD AGAIN                     F
*A*        JAP      CRTL                                                     F
*A*        JMP      CRTG                                                     F
*A*C0S1A   LDAE*    SW,X           GET SOURCE WORD                           F
*A*        LRLA     8              RIGHT BYTE                                F
*A*        ERAE*    TW,X           EOR TARGET WORD                           F
*A*        JAP      C0S1           SAME SIGN                                 F
*A*        JMP      TWSIN                                                    F
*A*C1S0A   LDAE*    TW,X           GET TARGET WORD                           F
*A*        LRLA     8              RIGHT BYTE                                F
*A*        ERAE*    SW,X           EOR SOURCE WORD                           F
*A*        JAP      C1S0           SAME SIGN                                 F
*A*        LDAE*    SW,X           GET SOURCE WORD                           F
*A*        JAP      CRTG                                                     F
*A*        JMP      CRTL                                                     F
*A*C1S1A   LDAE*    TW,X           GET TARGET WORD                           F
*A*        ERAE*    SW,X           EOR SOURCE WORD                           F
*A*        LRLA     8              RIGHT BYTES                               F
*A*        JAP      C1S1           SAME SIGN                                 F
*A*        LDAE*    TW,X           GET TARGET WORD                           F
*A*        LRLA     8              RIGHT BYTE                                F
*A*        JAP      CRTL                                                     F
*A*        JMP      CRTG                                                     F
```

```
   .INSERT,141                    VSORT2                10/15/74  MENG      E.1
*A*        EXT      GOFCB                                                    E.1
*A*        EXT      SIFCB                                                    E.1
*A*        EXT      WRT5                                                     F
   .REPLACE,267                   VSORT2                07/09/74  MENG      E.1
*D*        LDB      BR15         INITIALIZE FOR HIGH VALUES
*A*        DECR     2              INITIALIZE FOR HIGH VALUES                 E.1
   .REPLACE,269                   VSORT2                07/09/74  MENG      E.1
*D*        LDB      NEG
   .REPLACE,271                   VSORT2    SMR821      05/21/75  MENG      F
*D*        LDB      BS15         INITIALIZE FOR LOW VALUES
*A*        TZB                     INITIALIZE FOR LOW VALUES                 F
   .INSERT,305                    VSORT2                07/03/74  MENG      E.1
*A*        JMPM     SSEQ           SORT AND WRITE LAST SEQUENCE              E.1
   .INSERT,306                    VSORT2                10/24/74  MENG      E.1
*A*EOF1    LDA      SBEG,X         PREPARE TO PAD ENTIRE BUFFER              E.1
*A*        STA      SCUR,X                                                   E.1
*A*        LDBI     XSF1                                                     E.1
*A*        LDA      1,B            SEQUENCES ON THE SHORT FILE               E.1
*A*        LDBI     XSF2            PLUS                                     F
*A*        ADD      1,B            SEQUENCES ON THE LONG FILE                F
*A*        SUB      SFTB,X         MINUS SEQUENCES REQUIRED                  F
*A*        JANZ     SROT+6         ADD MORE SEQUENCES IF REQUIRED            E.1
   .REPLACE,312,313               VSORT2                07/03/74  MENG      E.1
*D*       JMPM  SSEQ          SORT AND WRITE LAST SEQUENCE
*D*       LDXI  SRTC
   .REPLACE,314                   VSORT2                10/24/74  MENG      E.1
*D*EOF1    LDA  SWCH,X        GET SWITCH
*A*        LDA      SWCH,X         GET SWITCH                                E.1
   .REPLACE,657,658               VSORT2                07/08/74  MENG      E.1
*D*SS11    LDA      SWK2,X
*D*        STA      SSLO         MAKE SWK2 THE LOW RECORD
*A*        JMP      SS11           EXCHANGE REQUIRED                         E.1
   .REPLACE,663                   VSORT2                07/08/74  MENG      E.1
*D*        JAP  SS12          YES, GET READY FOR NEXT PASS
*A*        JAP      SS13           YES, GET READY FOR NEXT PASS              E.1
   .REPLACE,665,668               VSORT2                07/08/74  MENG      E.1
*D*SS12    LDA      SSLO         GET PTR TO LOW RECORD
*D*        SUB      SWK1,X       SUB PTR TO SWK1
*D*        JAZ      SS13         IF SAME, DON'T EXCHANGE
*D*        LDA      SCUR,X       GET RECORD WORK AREA
*A*SS11    LDA      SND1,X         GET RECORD WORK AREA                      E.1
   .REPLACE,681                   VSORT2                07/08/74  MENG      E.1
```

```
*D*         LDA     SSLO          GET SOURCE RECORD
*A*         LDA     SWK2,X          GET SOURCE RECORD                          E,1
   ,REPLACE,690                     VSORT2              07/08/74  MENG      E,1
*D*         LDA     SSLO          GET SSLO RECORD AS TARGET
*A*         LDA     SWK2,X          GET TARGET RECORD                          E,1
   ,INSERT,700                      VSORT2              07/08/74  MENG      E,1
*A*         JMP     SS6                                                        E,1
   ,REPLACE,736                     VSORT2              10/16/74  MENG      E,1
*D*         READ    SICB,SI,0,1 READ A SORT CONTROL RECORD
*A*         CALL    FORMOD          CHECK FOR FOREGROUND MODIFICATIONS         F
*A*OSLF     OPEN    SIFCB,SI,0,1                                               F
*A*SINO     LDAE    SICB            RECORD LENGTH                              F
*A*         STA     LINK+4                                                     E,1
*A*         LDAE    SICB+1          BUFFER LOCATION                            E,1
*A*         STA     LINK+3                                                     E,1
*A*LINK     LDLINK  SI,0,0                                                     E,1
*A*NSCR     READ    SIFCB,SI,0,1    READ A SORT CONTROL RECORD                 E,1
   ,REPLACE,742                     VSORT2              10/16/74  MENG      E,1
*D*         LDAE    SINI+6        GET NUMBER OF WORDS READ
*A*         LDAE    RSCR+5          GET NUMBER OF WORDS READ                   E,1
   ,REPLACE,754                     VSORT2              10/16/74  MENG      E,1
*D*         ADDE SINI+6           PLUS NUMBER OF WORDS READ
*A*         ADDE    RSCR+5          PLUS NUMBER OF WORDS READ                  E,1
   ,REPLACE,774                     VSORT2              05/19/75  MENG      F
*D*         JANZ SINI+1           JUMP TO READ IF LIMIT NOT REACHED
*A*         JANZ    SINO            JUMP TO READ IF LIMIT NOT REACHED          F
   ,REPLACE,932                     VSORT2              06/27/75  MENG      F
*D*SI5      LDBE ITB7             GET INEXIT
*A*SI5      LDXI    SRTC            GET SORT CONTROL TABLE                     F
*A*         LDA     SWCH,X          GET SWITCH                                 F
*A*         BT      RA0+9,SI5A      MERGE OMIT OFF                             F
*A*         BT      RA0+4,ST08E     MERGE OMIT, BUT NO TAG SORT                F
*A*SI5A     LDBE    ITB7            GET INEXIT                                 F
   ,REPLACE,964                     VSORT2              10/15/74  MENG      E,1
*D*         LDA     MBEG          GET START OF MERGE WORK AREA
*A*         LDAE    MBEG            GET START OF MERGE WORK AREA               E,1
   ,INSERT,1044                     VSORT2              06/27/75  MENG      F
*A*ST08E    CALL    SCER,ST08,0     PRINT ERROR MESSAGE                        F
*A*         JMP     SI5A            CONTINUE                                   F
   ,INSERT,1075                     VSORT2              06/27/75  MENG      F
*A*ST08     DATA    '08  '                                                     F
   ,REPLACE,1541                    VSORT2              10/15/74  MENG      E,1
*D*         STB     BUFL          SAVE I/O BUFFER SIZE (WORDS)
```

```
*A*         STBE    BUFL        SAVE I/O BUFFER SIZE (WORDS)                E.1
   .REPLACE,1543               VSORT2              10/15/74  MENG          E.1
*D*         STB     BUFLB   SAVE I/O BUFFER SIZE (BYTES)
*A*         STBE    BUFLB       SAVE I/O BUFFER SIZE (BYTES)                E.1
   .REPLACE,1548               VSORT2              10/15/74  MENG          E.1
*D*         ADD     BUFL    ADD I/O BUFFER LENGTH
*A*         ADDE    BUFL        ADD I/O BUFFER LENGTH                       E.1
   .REPLACE,1558               VSORT2              06/13/75  MENG          F
*D*         ANAI    0404    USER EXIT SPECIFIED
*A*         ANAI    01404       USER EXIT(S) OR SKIP FINAL MERGE            F
   .REPLACE,1568               VSORT2              10/15/74  MENG          E.1
*D*         STA     BUFLB       OUTEXIT ONLY - WORKRECLEN = INLEN
*A*         STAE    BUFLB       OUTEXIT ONLY - WORKRECLEN = INLEN           E.1
   .REPLACE,1575               VSORT2              10/15/74  MENG          E.1
*D*         STA     BUFLB       INEXIT ONLY - WORKRECLEN = OUTLEN
*A*         STAE    BUFLB       INEXIT ONLY - WORKRECLEN = OUTLEN           E.1
   .REPLACE,1577               VSORT2              10/15/74  MENG          E.1
*D*         LDA     BUFL        GET 2ND BUFFER LENGTH
*A*         LDAE    BUFL        GET 2ND BUFFER LENGTH                       E.1
   .INSERT,1959                VSORT2              05/21/75  MENG          F
*A*LHW      EQU     0462                                                    F
*A*BS0      EQU     ONE                                                     F
*A*BS1      EQU     TWO                                                     F
   .REPLACE,2001,2015          VSORT2              06/12/75  MENG          F
*D*         TAS             SAVE LUN
*D*         JAP     DNM1
*D*DNM0     TZA             A REG =0 IF LUN NOT 0-255
*D*         JMP*    DNAME   RETURN
*D*DNM1     SUBI    101
*D*         JAN     DNM2    JUMP IF LUN 0-100
*D*         TAB
*D*         IXR             BUMP PTR TO PART2 OF LUN TABLE
*D*         SUBI    79
*D*         JAN     DNM2    JUMP IF LUN 101-179
*D*         TAB
*D*         SUBI    76
*D*         JAP     DNM0    JUMP IF LUN GREATER THAN 255
*D*         IXR             BUMP PTR TO PART3 OF LUN TABLE
*D*DNM2     TBA             GET LUN
   .REPLACE,2027               VSORT2              06/12/75  MENG          F
*D*         JMP*    DNAME   RETURN
*A*         ERAI    'SP'        IS IT A SPOOLER                             F
*A*         JAZ     DNM3        -YES-                                       F
```

```
*A*         LDA     1,B         -NO-                                F
*A*         JMP*    DNAME       RETURN                              F
*A*DNM3     LDA     0,B         DEVICE NUMBER (BITS 12-4)           F
*A*         LSRA    10          POSITION MSB OF UNIT NUMBER         F
*A*         ANA     SEVEN                                           F
*A*         ADD     V*LOT3      PLUS SPOOLER BASE                   F
*A*         INCR    014                                             F
*A*         LDA     0,X                                             F
*A*         ANA     RHW                                             F
*A*         OAR                                                     F
*A*         STA     OSAVE                                           F
*A*         ASLA    1                                               F
*A*         ADD     OSAVE                                           F
*A*         ADD     V*OSTB                                          F
*A*         TAX                                                     F
*A*         LDA     1,1                                             F
*A*         JMP*    DNAME                                           F
   .INSERT,2097             VSORT2          05/19/75  MENG         F
*A**                                                                F
*A*FORMOD   ENTR             CHECK FOR FOREGROUND MODIFICATIONS      F
*A**                                                                F
*A*         JAZ*    FORMOD      NO MODIFICATIONS                    F
*A*         STA     FORFLG      SAVE FLAG WORD                      F
*A*         ANA     BS0         SAVE FINAL MERGE BIT                F
*A*         LRLA    9           MOVE TO BIT 9                       F
*A*         LDXI    SRTC                                            F
*A*         STA     S*CB,X      SAVE IN SWITCH                      F
*A*         LDA     FORFLG      GET FLAG                            F
*A*         ANA     BS1         SAVE PARAMETER FLAG                 F
*A*         JAZ*    FORMOD      EXIT IF OFF                         F
*A*         LDAE    OSCB+3      MODIFY OPEN "SI"                    F
*A*         ANA     MS10                                            F
*A*         ORA     0,B                                             F
*A*         STAE    OSCB+3                                          F
*A*         LDAE    RSCB+3      MODIFY READ "SI"                    F
*A*         ANA     MS10                                            F
*A*         ORA     0,1                                             F
*A*         STAE    RSCB+3                                          F
*A*         LDXI    SIFCB       MODIFY FCB                          F
*A*         LDA     2,1                                             F
*A*         ANA     LHW                                             F
*A*         ORA     1,2                                             F
*A*         STA     2,1          KEY                                F
```

```
*A*          LDA     2,2
*A*          STA     0,1         LENGTH
*A*          LDA     3,2
*A*          STA     7,1         NAME1
*A*          LDA     4,2
*A*          STA     6,1             2
*A*          LDA     5,2
*A*          STA     9,1             3
*A*          LDX1    WRTS
*A*          LDA     4,1
*A*          ANA     MS10
*A*          STA     4,1
*A*          LDAB    SIOB+3      SKIP ECHO
*A*          ANA     MS10
*A*          STAB    SIOB+3
*A*          JMP*    FORE02
*A*MS10      DATA    0177700
*A*FORFLG    DATA    0
```

```
   .INSERT,46                    RPGRT                 03/24/75  MENG     E.2
*A*BM7     EQU     0467                                                    E.2
*A*V$LUT3  EQU     0402                                                    E.2
   .INSERT,875                   RPGRT                 03/24/75  MENG     E.2
*A*        LDA     RENO                                                    E.2
*A*        JAP     ARGERR                                                  E.2
   .INSERT,2675                  RPGRT                 06/06/75  MENG     E.2
*A*        TZA                                                             E.2
*A*        STA     RETI                                                    E.2
*A*        STA     PSTI                                                    E.2
   .INSERT,3487                  RPGRT                 03/25/75  MENG     E.2
*A*        JAZ     *+4           TOP-OF-FORM                               E.2
   .INSERT,3644                  RPGRT                 03/24/75  MENG     E.2
*A*        ERAI    'SP'          IS IT A SPOOLER                           E.2
*A*        JANZ    DEVTS1        NO                                        E.2
*A*        LDA     0,1           DEVICE NUMBER (BITS 12-4)                 E.2
*A*        LSRA    10            POSITITION MSB OF UNIT NUMBER             E.2
*A*        ANA     BM7                                                     E.2
*A*        ADD     V$LUT3        SPOOL LUT BASE                            E.2
*A*        INCR    014           ADJUST PAST LUT SIZE WORD TO X            E.2
*A*        LDA     0,1           CURRENT DST ENTRY NUMBER                  E.2
*A*        ANA     RHW                                                     E.2
*A*        DAR                                                             E.2
*A*        STA     OBTA                                                    E.2
*A*        ASLA    1             TIMES 2                                   E.2
*A*        ADD     OBTA          PLUS 1                                    E.2
*A*        ADD     V$DSTB                                                  E.2
*A*        TAX                                                             E.2
*A*DEVTS1  LDA     1,1                                                     E.2
   .INSERT,3656                  RPGRT                 03/24/75  MENG     E.2
*A*        LDA     1,1                                                     E.2
*A*        ERAI    'CT'          IS IT A CRT                               E.2
*A*        JAZ     DEVST1        GOOD AS A LINE PRINTER                    E.2
*A*        LDA     1,1                                                     E.2
*A*        ANA     LHW                                                     E.2
*A*        ERAI    0152000       IS IT A TTY                               E.2
*A*        JAZ     DEVST1        GOOD AS A LINE PRINTER                    E.2
```

```
   .INSERT,1                                                                E.2
:A*V75     SET      1                                                       E.2
   .INSERT,13
   .INSERT,201                    C52LIP    SMR-791              METR       E.2
:A**                                                                        E.2
:A**                TF PREVIOUS INTERRUPT WAS CONTROL CHARACTER             E.2
:A**                AND BYTE COUNT ZERO THAN LCIGN IS SET                   E.2
:A**                                                                        E.2
:A*        LDB      LILCW                                                   E.2
:A*LCW     SET      B                                                       E.2
:A*        TESTF    LCW,LCIGN,LCIGNB,LCIGNZ   IGNORE FLAG                   E.2
:A*        JAZ      LIB01                                                   E.2
:A*        CLEARF   LCW,LCIGN,LCIGNB,LCIGNZ                                 E.2
:A*        JMP      LIF0                      IGNORE INTERRUPT              E.2
:A*LIB01   EQU      *                                                       E.2
   .INSERT,303                    C52LIP    SMR-791              METR       E.2
:A*        JMPM     LID00                                                   E.2
   .REPLACE,312                   C52LIP    SMR-791              METR       E.2
:D*        JXZ      LID1           FORCE BYTE COUNT 0 TERMINATION
:A*        JXZ      LID2           FORCE BYTE COUNT ZERO TERMINATION        E.2
   .INSERT,319                    C52LIP    SMR-791              METR       E.2
:A*LID00   ENTR                                                             E.2
:A*        LDX      LILCW                                                   E.2
:A*LCW     SET      X                                                       E.2
:A*        FETCHA   LCW,LCTBL,LCTBLB,LCIBLZ  CHECK IF ALSO                  E.2
:A*        SUB      BM7777                   BYTE COUNT ZERO                E.2
:A*        JANZ     LID0C          NO                                       E.2
:A*        SETF     LCW,LCIGN,LCIGNB,LCIGNZ  SET IGNORE FLAG                 E.2
:A*LCW     SET      B                                                       E.2
:A*LID0C   EQU      *                                                       E.2
:A*        JMP*     LID00                                                   E.2
   .REPLACE,322,323               C52LIP    SMR-791              METR       E.2
:D*        JAZ      LID1           CRC>0
:D*        STA      LIBL
:A*        STA      LIBL                                                    E.2
:A*        JAZ      LID2           CRC EQUAL 0                              E.2
   .REPLACE,382                   C52LIP    SMR-779              METR       E.2
:D*LTE2    LDA      LCLCB,LCW                                              BSC
:A*LTE2    TESTF    LCW,LCPOL,LCPOLB,LCPOLZ       POLL MODE?                E.2
:A*        JANZ     LIF4                          YES                       E.2
:A*LTE21   LDA      LCLCB,LCW                                               E.2
   .INSERT,392                    C52LIP    SMR-791              METR       E.2
:A*        JMPM     LID00          CHECK IF ALSO BYTE COUNT ZERO            E.2
```

```
   .INSERT,397                          C52LIP   SMR=779                 METR      E.2
;A**                BSC RECEIVE POLL MODE, CHECK IF EOT RECV (BEGIN POLL)          E.2
;A*LTE4    STB      LIPOLL              SAVE B REGISTER                            E.2
;A*         LDA      LCTBA,LCW           NEXT BYTE ADDRESS                          E.2
;A*         JAN      LIF6                EOT IS IN PREVIOUS WORD, RIGHT BYTE        E.2
;A*         TAB                                                                     E.2
;A*         IFF      VORTEX=2                                                       E.2
;A*         DMF      MAP,LIMAP3          STATE NN                                   E.2
;A*         LDA      0,B                 EOT , LEFT BYTE IS LOADED                  E.2
;A*         IFF      VORTEX=2                                                       E.2
;A*         DMF      MAP,VSSTO           STATE 00                                   E.2
;A*LTE5     ANAI     0177400             CLEAR RIGHT BYTE                           E.2
;A*         SUB      LCTC1,LCW           CONTROL CHAR, IN LCW TO MATCH              E.2
;A*         JAZ      LIF0                YES, IGNORE INTERRUPT                      E.2
;A*         LDB      LIPOLL              RESTORE B REGISTER                         E.2
;A*         JMP      LIF21               CONTINUE                                   E.2
;A*LTE6     ANAI     077777              CLEAR SIGN BIT                             E.2
;A*         SUB      B1                  GET ADDRESS OF PREVIOUS WORD               E.2
;A*         TAB                                                                     E.2
;A*         IFF      VORTEX=2                                                       E.2
;A*         DMF      MAP,LIMAP3          STATE NN                                   E.2
;A*         LDB      0,B                 LOAD EOT                                   E.2
;A*         IFF      VORTEX=2                                                       E.2
;A*         DMF      MAP,VSSTO           STATE 00                                   E.2
;A*         IRLA     B8                  SHIFT TO LEFT BYTE                         E.2
;A*         JMP      LIF5                                                           E.2
   .INSERT,398                          C52LTP   SMR=779                 METR      E.2
;A*LTPOLI   DATA     0                   POLL WORD ADDRESS                          E.2
   .INSERT,489                          C52LTP   V75       2/12/75   J.METR      V75
;A*         IFT      V75=1                                                          V75
;A*         GOTO     2                                                              V75
;A*         TESTF    X,TB75,TB75B,TB75Z                                             V75
;A*         JAZ      LIH2                                                           V75
;A*         SAVER    Y,TBTSR3            SAVE V75 REGS IN TTDR INTERRUPT STACK      V75
;A*2        CONT                                                                    V75
;A*LTH2     EQU      *                                                              V75
```

```
   .INSERT,13
   .REPLACE,106                    C52FUN    SMR-779              METR      E.2
;D*FU02    LDB      FUFUNC
;A*FU02    LDBE     FUFUNC
   .INSERT,150                     C52FUN    SMR-779              METR      E.2
;A*        DATA     FUFF           32 SET BSC RECEIVE POLL MODE
;A*        DATA     FUGG           33 RESET BSC RECEIVE POLL MODE
   .INSERT,439                     C52FUN    SMR-                           E.2
;A*        ANA      BM77           GET 6 BITS ONLY
   .REPLACE,568                    C52FUN    SMR-779              METR      E.2
;D*        LDAI     04640          ERROR CODE
;A*FUAA2   LDAI     04640          ERROR CODE
   .INSERT,612                     C52FUN    SMR-779              METR      E.2
;A**                FUNC 32 SET IN BSC RECEIVE POLL MODE
;A*FUFF    TESTF    LSD,LSBSC,LSBSCB,LSBSCZ   TEST IF IN BSC
;A*        JAZ      FUAA2          ERROR
;A**FUNC IS ALLOWED ONLY WITH A MODIFIED ADAPTER. INDICATION IS WHEN
;A** VSPOLL#1. SET AT SGEN TIME VIA DEF DIRECTIVE
;A*        EXT      VSPOLL
;A*        LDAE     VSPOLL
;A*        JAZ      FUAA2                          ERROR
;A*        SETF     LSD,LSPOL,LSPOLB,LSPOLZ   POLL MODE FLAG
;A*        JMP      FU70
;A*FUGG    TESTF    LSD,LSBSC,LSBSCB,LSBSCZ   TEST IF IN BSC
;A*        CLEARF   LSD,LSPOL,LSPOLB,LSPOLZ
;A*        JMP      FU70
```

```
   .INSERT,13
   .REPLACE,41                    VT$IOC    REENT     03/06/75  METR      E.2
;D*VT$IOC PUSH     CC00
;A*VT$IOC EQU      *                                                     E.2
;A**               FORCE STATE 00                                        E.2
;A*        IFF     VORTEX-2                                              E.2
;A*        OMF     MAP,V$STO                                             E.2
;A*        PUSH    CC00                                                  E.2
   .INSERT,42                     VT$IOC    REENT     03/06/75  METR      E.2
;A*        DINTS                                                         E.2
;A**               DECREMENT NO OF ACTIV  REQUESTS IN TBIO               E.2
;A*        LDB     V$CTL          TIDB AD RESS                           E.2
;A*TIDB    SET     B                                                     E.2
;A*        LDB     TBRSTS,TIDB    CTB                                    E.2
;A*CTB     SET     B                                                     E.2
;A*        LDB     CTRQB,CTB      RQST                                   E.2
;A*RQST    SET     B                                                     E.2
;A*        LDB     RTIDB,RQST     USER TI B                              E.2
;A*        LDX     TBIO,B                                                E.2
;A*        DXR                                                           E.2
;A*        STX     TBIO,B                                                E.2
   .INSERT,113                    VT$IOC    REENT     03/06/75  METR      E.2
;A**               AFTER REQUEST IS DEQU UED, INCREMENT NUM              E.2
;A**               OF ACTIVE TO IN TIDB                                  E.2
;A*        LDB     5,STACK        RQST                                   E.2
;A*        LDB     RTIDB,B        USER TI B                              E.2
;A*TIDB    SET     B                                                     E.2
;A*        LDA     TBIO,TIDB                                             E.2
;A*        IAR                                                           E.2
;A*        STA     TBIO,TIDB                                             E.2
   .INSERT,131                    VT*IOC    SMR-670   04/08/75  KERNS     E.2
;A*        DAR                    ADJUST TO BASE ZERO INDEXING           E.2
   .REPLACE,253,254               VT$IOC    REENT     03/06/75  METR      E.2
;D*        DELAY   1              NO. DELAY 1 COUNT                     V2
;D*        IMP     CC09           AND TRY AGAIN                         V2
;A*        DINTS                                                         E.2
;A**                        RETHREAD USER REQUEST TO CTB                 E.2
;A*        LDXE    VTSTAK                                                E.2
;A*STACK   SET     X                                                     E.2
;A*        LDB     5,STACK          RQBLK                                E.2
;A*RQST    SET     B                                                     E.2
;A*        LDX     6,STACK          CTB                                  E.2
;A*CTB     SET     X                                                     E.2
```

```
:A*        LDA     CTRQB,CTB        NEXT  EQUEST ON CTB                E.2
:A*        STA     RADNR,RQST    LINK TO CURRENT REQUEST               E.2
:A*        STB     CTRQB,CTB        RETHR AD REQUEST TO CTB            E.2
:A*        LDXE    VTSTAK  RELOAD STACK  OINTER                        E.2
:A*STACK   SET     Y                                                   E.2
:A*        LDA     4,STACK P COUNTER                                   E.2
:A*        ANA     BS15                                                E.2
:A*        ORAI    VTNEXT                                              E.2
:A*        STA     4,STACK                                             E.2
:A*        EXT     VTPOP                                               E.2
:A*        JMP     VTPOP                                               E.2
:A**RETURN HERE FROM VTPOP, DISABLE INT RRUPTS AND GO                  E.2
:A**TO THE DISPATCHER TO ACTIVATE NEXT  ASK                            E.2
:A*VTNEXT  EQU     *                                                   E.2
:A*        EXT     DISP1                                               E.2
:A*        DINTS                                                       E.2
:A*        LDX     V$CTL                                               E.2
:A*        JMP     DISP1                                               E.2
```

```
    .INSERT,13
    .REPLACE,470                    TYREAD                          KERNS   F.2
D*          STA     RDBUFF
A*          STAE    RDBUFF                                                  F.2
    .REPLACE,475                    TYREAD                          KERNS   F.2
D*          LDA     RDBUFF
A*          LDAE    RDBUFF                                                  F.2
    .REPLACE,503                    TYREAD                          KERNS   F.2
D*          LDA     RDBUFF          SET WORD/BYTE COUNT FLAG IN TCD
A*          LDAE    RDBUFF          SET WORD/BYTE COUNT FLAG IN TCD          F.2
    .REPLACE,613                    TYREAD  SMR-                    KERNS   F.2
D*          LDB     RDBUST                                                  V2
A*          JMPM    TC$ERR                                                  F.2
A*          TAB                                                             F.2
    .INSERT,1692                    TYREAD  SMR-    10/31/74        KERNS   F.1
A*          GETMEM  VT$MP1,14                                               F.1
```

```
   .INSERT,13
   .INSERT,14                    VT$TCQ    REENT     03/06/75  METR      E.2
*A*RLDX    MAC                                                           E.2
*A*        LDX      V$CTL    TIDB                                        E.2
*A*        LDX      TBRSTS,X        CTB A DRESS                          E.2
*A*CTB     SET      X                                                    E.2
*A*        EMAC                                                          E.2
   .REPLACE,16,18                VT$TCQ    REENT     03/06/75  METR      E.2
*D*        LDAI     P(2)
*D*        LDBI     P(1)
*D*        STX      *+6
*A*        TXA                                                           E.2
*A*        ADAT     P(2)                                                 E.2
*A*        TAB                                                           E.2
*A*        LDAI     P(1)                                                 E.2
   .REPLACE,25                   VT$TCQ    REENT     03/06/75  METR      E.2
*D*        LDXI     *
   .REPLACE,28,30                VT$TCQ    REENT     03/06/75  METR      E.2
*D*        LDAI     P(1)
*D*        LDB      P(2)
*D*        STX      *+6
*A*        TXA                                                           E.2
*A*        ADAT     P(2)                                                 E.2
*A*        TAB                                                           E.2
*A*        LDAI     P(1)                                                 E.2
   .REPLACE,37                   VT$TCQ    REENT     03/06/75  METR      E.2
*D*        LDXI     *
   .REPLACE,73,76                VT$TCQ    REENT     03/06/75  METR      E.2
*D*        LDX      V$CTL
*D*        LDX      TBRSTS,X
*D*        STX      TQCTBA         SAVE CTBL ADDR
*D*CTB     SET      X
*A*        RLDX              SET X TO CTB                                E.2
*A*        LDB      CTRQB,CTB       REQUE T                              E.2
*A*RQST    SET      B                                                    E.2
*A*        LDB      RTIDB,RQST      USER'  TIDB                          E.2
*A*        STB      CTUIDB,CTB      SAVE  N CTB TEMP                     E.2
*A*TIDB    SET      B                                                    E.2
*A**                DECREMENT NO OF ACTIV  I/O IN USER TIDB             E.2
*A*        DINTS                                                         E.2
*A*        LDA      TBTO,TIDB                                            E.2
*A*        DAR                                                           E.2
*A*        STA      TBTO,TIDB                                            E.2
```

```
*A*         FINTS                                                          E.2
*A*VTTC01   EQU     *                                                      E.2
*A*         RLDX              SET X TO CTB                                 E.2
   .REPLACE,77,78             VTSTCQ   REENT   03/06/75  METR             E.2
*D*         TFF     VORTEX-2                                              V2
*D*         DINTS                                                         V2
   .INSERT,84                 VTSTCQ   REENT   03/06/75  METR             E.2
*A**                INCREMENT NO OF ACTIV  I/O INUSER'STIOB               E.2
*A*         DINTS                                                         E.2
*A*         LDB     RTTOB,ROST    USER' TTOB                              E.2
*A*TTOB     SET     B                                                     E.2
*A*         LDA     TBIO,TTOB                                             E.2
*A*         TAB                                                           E.2
*A*         STA     TBIO,TTOB                                             E.2
*A*         LDB     CTBOB,CTB       RESTO TO USER'S REQUEST               E.2
*A*ROST     SET     B                                                     E.2
   .REPLACE,107,110           VTSTCQ   REENT   03/06/75  METR             E.2
*D*         STA     TQTCOA        TOC VALIDATED LUN, GET LOG, TERM, TBL ENTRY
*D*         LDAE*   TQTCOA
*D**                              SAVE TCO ADOR
*D*         STA     TQTCOA
*A**        STA     CTCOA,CTB       TOC V LIOATED LUN, GET L              E.2
*A**                                TERM, TBL, ENTRY                      E.2
*A*         LDAE*   CTCOA,CTB                                             E.2
*A**                               SAVE  CO ADDRESS                       E.2
*A*         STA     CTCOA,CTB                                             E.2
   .REPLACE,121               VTSTCQ   REENT   03/06/75  METR             E.2
*D*         LDAE    TQTCOA
*A*         LDA     CTCOA,CTB                                             E.2
   .INSERT,176                VTSTCQ   REENT   03/06/75  METR             E.2
*A*         FINTS                                                         E.2
   .REPLACE,180,181           VTSTCQ   REENT   03/06/75  METR             E.2
*D*CTB      SET     X
*D*         LDA     TBRSTS,CTB
*A*         LDA     TBRSTS,X      CTB                                     E.2
   .INSERT,189                VTSTCQ   REENT   03/06/75  METR             E.2
*A*         DINTS                                                         E.2
   .REPLACE,197               VTSTCQ   REENT   03/06/75  METR             E.2
*D*         JMP     VTSTCQ
*A*         JMP     VTTC01                                                E.2
   .REPLACE,222               VTSTCQ   REENT   03/06/75  METR             E.2
*D*         STB     TQROST
*A*         STB     CTBOS,CTB                                             E.2
```

```
   .REPLACE,231                    VT$TCQ    REENT     03/06/75  METR      E.2
*D*TQ100A  LDAE    TQTCDA
*A*TQ100A LDA      CTCDA,CTB                                              E.2
   .REPLACE,244                    VT$TCQ    REENT     03/06/75  METR      E.2
*D*        JMP     VT$TCQ          PROCESS NEXT REQUEST
*A*        JMP     VTTCQ1          PROCESS NEXT REQUEST                   E.2
   .INSERT,268                     VT$TCQ    REENT     06/12/75  KADISON   E.2
*A*CTB     SET     X
   .INSERT,269                     VT$TCQ    REENT     03/06/75  METR      E.2
*A*        FINTS                                                          E.2
   .REPLACE,277                    VT$TCQ    REENT     03/06/75  METR      E.2
*D*        JANZ    TQ130           QUEUED FUNCTION
*A*        DINTS                                                          E.2
*A*        JAN     TQ130           QUEUED  UNCTION                        E.2
*A*        STA     CTENT,CTB                                              E.2
*A*        ROE                                                            E.2
*A*        ADD     BR15            SET OF IF NOT FUNC 4                   E.2
   .INSERT,279                     VT$TCQ    REENT     03/06/75  METR      E.2
*A**                               FUNC 4, 5, OR 6,                       E.2
   .REPLACE,285                    VT$TCQ    REENT     03/06/75  METR      E.2
*D*        LDB     TQRQST          GET LOGICAL UNIT NO. FOR TERMINAL
*A*        LDB     CTRQS,CTB         GET L GICAL UNIT NO.FOR              E.2
   .INSERT,291                     VT$TCQ    REENT     03/06/75  METR      E.2
*A*        TOE     TQ1102          SKIP IF NOT FUNC 4                     E.2
   .REPLACE,297,299                VT$TCQ    REENT     03/06/75  METR      E.2
*D**
*D*        LDX     TQTCDA
*D*TCD     SET     X
*A*TQ1102  EQU     *                                                      E.2
*A*        LDB     CTCDA,CTB         TCD A DRESS                          E.2
*A*TCD     SET     B                                                      E.2
   .REPLACE,301                    VT$TCQ    REENT     03/06/75  METR      E.2
*D*        STA     TQRCA
*A*        STA     CTRCA,CTB                                              E.2
   .REPLACE,303                    VT$TCQ    REENT     03/06/75  METR      E.2
*D*        STA     TQWCA
*A*        STA     CTWCA,CTB                                              E.2
   .REPLACE,304                    VT$TCQ    I/O CLEAR 04/22/75  KADISON   E.2
*D*        TZA                     CLEAR CCM RQBLK ADDR
*A*        LDA     CTENT,CTB       FUN TYPE                               E.2
*A*        JANZ    TC1103          SKIP IF NOT FUNC 4                    E.2
   .REPLACE,307                    VT$TCQ    REENT     03/06/75  METR      E.2
*D*        LDBI    TQMEM
```

```
*A*TC1103 EQU     *
*A*       TXA                     CTL TBL ADDR                          F.2
*A*       TBX                     TCD ADD                               F.2
*A*       ADAT    CTMEM           REQ BLK ADDR                          F.2
*A*       TAB                                                           F.2
   .REPLACE,317                   VT$TCO    REENT     03/06/75  MEIR    F.2
*D*       LDAI    21              FUNC CODE = 21 FOR I/O CLEAR
*A*       RLDX            SET X TO CTB                                  F.2
*A*       LDA     CTFNT,CTB       FUN TYP  FLAG                         F.2
*A*L      SET     *                                                     F.2
*A*       XAZ     KFCND           FUNC TO 21 IF FUNC 4 REQUE            F.2
   .REPLACE,321,325               VT$TCO    REENT     03/06/75  MEIR    F.2
*D*       JMP     TQMEM
*D**                              I/O CLEAR DONE, RETURN MEMORY FOR CCM RQBLK
*D*TC110A LDX     TQTCDA          RESTORE X REG, TO TCD ADDR.
*D*TCD    SET     X
*D*       LDA     TQRCA           CHECK IF READ ACTIVE
*A*       TJMP    CTMEM,CTB                                             F.2
*A*TC110A RLDX            X ON CTB                                      F.2
*A*       LDA     CTFNT,CTB       FUNC TY E                             F.2
*A*       JANZ    TC120A          SKIP IF NOT FUNC CODE 4               F.2
*A**                              I/O CLEAR DONE, RETU  MEMORY FOR CCM RQ F.2
*A*       LDB     CTCDA,CTB         TCD A DRESS                         F.2
*A*TCD    SET     B                                                     F.2
*A*       LDA     CTRCA,CTB         CHECK IF READ ACTIVE                F.2
   .REPLACE,330                   VT$TCO    REENT     03/06/75  METR    F.2
*D*       PUTMEM  VT$MP1,TQRCA
*A*       PUTMEM  VT$MP1,CTRCA                                          F.2
*A*       RLDX                    X TO CT                               F.2
   .REPLACE,331                   VT$TCO    REENT     03/06/75  METR    F.2
*D*TC110B LDA     TQWCA           CHECK IF WRITE ACTIVE
*A*TC110B LDA     CTWCA,CTB         CHECK IF WRITE ACTIVE               F.2
   .REPLACE,336                   VT$TCO    REENT     03/06/75  METR    F.2
*D*       PUTMEM  VT$MP1,TQWCA
*A*       PUTMEM  VT$MP1,CTWCA                                          F.2
*A*       RLDX                    X TO CT                               F.2
   .REPLACE,338                   VT$TCO    REENT     03/06/75  METR    F.2
*D*TC120  LDB     TCRQH,TCD
*A*TC120  LDX     CTCDA,CTB                                             F.2
*A*TCD    SET     X                                                     F.2
*A*       LDB     TCRQH,TCD                                             F.2
*A*       DINTS                                                         F.2
   .REPLACE,346                   VT$TCO    REENT     03/06/75  METR    F.2
```

```
*D*        STX     TQX            YES, SAVE X                                V2
   .REPLACE,351                   VT$TCQ    REENT     03/06/75  METR        E.2
*D*        LDX     TQX                                                      V2
   .REPLACE,352                   VT$TCQ    REENT     03/06/75  METR        E.2
*D*        DINTS                                                            V2
   .INSERT,353                    VT$TCQ    REENT     03/06/75  METR        E.2
*A*        RLDX            X ON CTB                                         E.2
   .REPLACE,356                   VT$TCQ    REENT     03/06/75  METR        E.2
*D*TQ120A  LDB     TQRQST         GET REQUEST ADDRESS
*A*TQ120A  RLDX    X ON CTB                                                 E.2
*A*        LDB     CTRQS,CTB        GET R QUEST ADDRESS                     E.2
   .INSERT,357                    VT$TCQ    REENT     03/06/75  METR        E.2
*A*        LDX     CTCDA,CTB        TCD                                     E.2
   .REPLACE,368,369               VT$TCQ    REENT     03/06/75  METR        E.2
*D*        FINTS
*D*        LDX     TQCTBA         RESTORE X TO CTBL ADDR
*A*        DINTS                                                            E.2
*A*        RLDX            X ON CTB                                         E.2
   .REPLACE,374                   VT$TCQ    REENT     03/06/75  METR        E.2
*D*TQ130   LDB     TQRQST
*A*TQ130   EQU     *                                                        E.2
*A*        RLDX            SET X ON CTB                                     E.2
*A*        LDB     CTRQS,CTB                                                E.2
   .REPLACE,376,377               VT$TCQ    REENT     03/06/75  METR        E.2
*D*        TFF     VORTEX-2                                                 V2
*D*TQX     DATA    0              SAVE X                                    V2
   .REPLACE,381,387               VT$TCQ    REENT     03/06/75  METR        E.2
*D**
*D*TQCTBA  DATA    0              CONTROLLER TABLE ADDR
*D*TQRCA   DATA    0              SAVE CELL FOR TCRCA ENTRY IN TCD
*D*TQWCA   DATA    0              SAVE CELL FOR TCWCA ENTRY IN TCD
*D*TQTCDA  DATA    0              ADDR OF TERMINAL CONTROLLER DESC(TCD)
*D*TQRQST  DATA    0              TCM REQUEST ADDRESS
*D*TQMEM   PSS     12             MEMORY FOR CCM RQST BLOCK
*A****             STORAGE IS IN CTB:                                       E.2
*A****             CTRCA-- FOR TCRCA IN  CD                                 E.2
*A****             CTWCA-- FOR TCWCA IN  CD                                 E.2
*A****             CTCDA-- FOR ADDRESS O  TCD                               E.2
*A****             CTRQS-- FOR TCM RESQU ST ADDRESS                         E.2
*A****             CTUIDB-- FOR USER@S T DB ADDRESS                         E.2
*A***              CTMEM-- FOR MEMORY FO  CCM RQST BLOCK                    E.2
*A*KFCNO   ADD     K21+*-L-1      CREATE  UNC CODE 21                       E.2
*A*K21     DATA    21                                                       E.2
```

```
   .INSERT,83                        CTTCOA    REENT                  METR      E,2
*A**                THE FOLLOWING ARE USED AS A TEMPORARY STORAGE                E,2
*A**                BY VT$TCO                                                    E,2
*A*        DATA     0                CTRCA (TQRCA) FOR TCRCA IN TCD               E,2
*A*        DATA     0                CTWCA (TQWCA) FOR TCWCA IN TCD               E,2
*A*        DATA     0                CTCDA (TQTCDA) ADDRESS OF TCD                E,2
*A*        DATA     0                CTRQS (TQRQST) TCM REQST ADDRESS             E,2
*A*        DATA     0                CTUIDB  USER TIDB ADDRESS
*A*        BSS      14               CTMEM (TQMEM) MEMORY FOR CCM RQST BLK        E,2
*A*        DATA     0                CTENT - FUNCTION TYPE                        E,2
```

```
      .INSERT,13
      .INSERT,53                          C52SST    SMR-787   02/20/75  MCDANIELS E.2
*A*         ANA      RM77                                                          E.2
```

```
    .INSERT,13
    .INSERT,92                          C52XMT    MODE 4    05/01/75  KADISON   F.2
*A*         FRA     BS12                ZERO IF MODE = 1                          F.2
    .REPLACE,110                        C52XMT    MODE 4    05/01/75  KADISON   F.2
*D**                TF MODE >1',  SET LCITB FLAG TN LCW                         BSC
*A**                TF MODE =1,   SET LCITB FLAG TN LCW                         F.2
    .REPLACE,112                        C52XMT    MODE 4    05/01/75  KADISON   F.2
*D*         JAZ     XM22                                                        BSC
*A*         JANZ    XM22                SKIP IF NOT MODE 1                      F.2
    .REPLACE,168                        C52XMT    BREAK ERR 05/01/75  KADISON   F.2
*D*         ANAI    0237                CLEAR BITS 5,6,8,9
*A*         ANAI    0637                CLEAR BITS 5, 6, 9                      F.2
```

```
   .REPLACE,1                    CC$OCL    NO SMR                KERNS     E.2
*D*        EJEC
*A*VORTEX SET      1                                                       E.2
   .INSERT,13
   .REPLACE,35,38                CC$OCL    SMR-809   04/08/75   KERNS     E.2
*D**
*D**       EXIT EXECUTION
*D**
*D*        EXIT
*A*        SUSPND  0                                                       E.2
```

```
   .REPLACE,1                    CC$CBS    NO SMR                KERNS     E.2
*D*        EJEC
*A*VORTEX SET      1                                                       E.2
   .INSERT,13
   .REPLACE,60                   CC$CBS    SMR-                  BRACKETT E.2
*D*        LDAE     CC$FIL
*A*CB01    LDAE     CC$FIL                                                 E.2
   .REPLACE,63                   CC$CBS    SMR-                  BRACKETT E.2
*D*CB01    LDX      CBSAVX       CALL
*A*        LDX      CBSAVX       CALL                                      E.2
   .INSERT,96                    CC$CBS    SMR-                  BRACKETT E.2
*A*        LDAE     CC$FLG       TEST IF OVERFLOW                          E.2
*A*        JAZ      CF02                                                   E.2
   .REPLACE,100                  CC$CBS    SMR-                  BRACKETT E.2
*D*        LDAE*    CC$EMT       MOVE INTERRUPT WORDS AND MOVE EMPTY POINTER
*A*CF00    LDAE*    CC$EMT       MOVE INTERRUPT WORDS AND EMPTY POINTER  E.2
   .INSERT,112                   CC$CBS    SMR-                  BRACKETT E.2
*A*CF02    INRE     CC$FLG       RESET OVERFLOW CONDITION                  E.2
*A*        JMP      CF00                                                   E.2
```

```
   .INSERT,13
   .INSERT,167                    CC$SCW    NO SMR    03/31/75  METR      E.2
*A**              CLEAR LAST DATA BLOCK FLAG                                E.2
*A*        CLEARF  LCW,LCLDB,LCLDBB,LCLDBZ                                  E.2
   .INSERT,171                    CC$SCW    SMR-779              METR      E.2
*A**              SET IN BSC RECEIVE POLL MODE FLAG (AB=11)                 E.2
*A*        FETCHA  LSD,LSPOL,LSPOLB,LSPOLZ                                  E.2
*A*        SETA    LCW,LCPOL,LCPOLB,LCPOLZ                                  E.2
   .INSERT,173                    CC$SCW    SMR-787              MCDANIELS E.2
*A*        ANA     BM77                                                     E.2
```

```
    .INSERT,13
    .REPLACE,249                        CCSCRO    SMR-R12   07/01/75  KADISON   F.2
*D*         LDA     0,X                 A = STATUS                                V2
    .INSERT,250                         CCSCRO    SMR-R12   07/01/75  KADISON   F.2
*A*         LDA     RADNR,R                                                     F.2
*A*         EXT     CCSDCM                                                      F.2
*A*         STAE    CCSDCM                                                      F.2
*A*         LDA     0,X                 A = STATUS                              F.2
```

```
    .INSERT,13
    .REPLACE,270                    TCSCEX    ERROR CODE05/01/75  KADISON   F.2
*D*         ANAI    0440            MASK OFF IRRECOVERABLE ERRORS
*A*         ANAI    01440           MASK OFF IRRECOVERABLE ERRORS           F.2
    .REPLACE,272                    TCSCEX    ERROR CODE05/01/75  KADISON   F.2
*D*         BT      RA1+5,CR20A     PARITY ERROR, FLSE BREAK
*A*         BT      RA1+5,CR20A     PARITY ERROR                            F.2
*A*         BT      RA1+9,CR20D     OVERFLOW                                F.2
    .INSERT,286                     TCSCEX    ERROR CODE05/01/75  KADISON   F.2
*A*CR20D    LDAI    01635           CODE = 071, ERR FLG, COMP CODE = 5      F.2
*A*         JMP     CR50                                                    F.2
    .REPLACE,388                    TCSCEX    SMR-      07/02/75  KADISON   F.2
*D*         LDA     0,X             A=STATUS                                V2
    .INSERT,389                     TCSCEX    SMR-      07/02/75  KADISON   F.2
*A*         LDA     RADNR,R                                                 F.2
*A*         EXT     TCSUCM                                                  F.2
*A*         STAE    TCSUCM                                                  F.2
*A*         LDA     0,X             A = STATUS                              F.2
    .REPLACE,482                    TCSCEX    CHAINING  05/01/75  KADISON   F.2
*D*         JMP     CE30            NOT COMPLETE, CHECK WRITE
*A*         TESTF   TCD,TCIBC,TCIBCB,TCIBCZ                                 F.2
*A*         IAZ     CE30            NO BUFFER CHAINING                      F.2
*A*         FETCHA  TCD,TCCCF,TCCCFB,TCCCFZ                                 F.2
*A*         IAZ     CE30            NO COMPLETED BUFFERS                    F.2
    .REPLACE,588,589                TCSCEX    SMR-842   07/03/75  KERNS     F.2
*D*         TZA                     RESET DELTA TIME
*D*         STA     CEDELT
*A*         IMPM    CETIME          CALCULATE DELTA TIME                    F.2
*A*         LDA     VSTMS                                                   F.2
*A*         STA     TCSTMS                                                  F.2
*A*         LDA     VSTMN                                                   F.2
*A*         STA     TCSTMN                                                  F.2
    .REPLACE,629                    TCSCEX    DELAY 3   05/01/75  KADISON   F.2
*D*CE90     DELAY   0,0,2
*A*CE90     DELAY   0,0,3                                                   F.2
    .INSERT,642                     TCSCEX    SMR-842   07/03/75  KERNS     F.2
*A*         IMPM    CETIME          CALCULATE DELTA TIME                    F.2
*A*         IMP     TCSCEX                                                  F.2
*A*CETIME   ENTR                                                            F.2
    .REPLACE,667                    TCSCEX    SMR-842   07/03/75  KERNS     F.2
*D*         IMP     TCSCEX          START TCMEVEC
*A*         RETU*   CETIME                                                  F.2
```

```
    .INSERT,13
    .REPLACE,35                         VTSOCT    TCM NO     05/01/75  KADISON   E.2
*D*NTCTYP EQU     1                     NUMBER OF TCM TYPES
*A*NTCTYP EQU     5                     NUMBER OF TCM TYPES                       E.2
    .INSERT,250                         VTSOCT    TCM NO     05/01/75  KADISON   E.2
*A*         JMP     OVLAY2                                                        E.2
*A*         JMP     OVLAY3                                                        E.2
*A*         JMP     OVLAY4                                                        E.2
*A*         JMP     OVLAY5                                                        E.2
*A*         JMP     OVLAY6                                                        E.2
*A*         JMP     OVLAY7                                                        E.2
*A*         JMP     OVLAY8                                                        E.2
*A*         JMP     OVLAY9                                                        E.2
    .INSERT,253                         VTSOCT    TCM NO     05/01/75  KADISON   E.2
*A*OVLAY9 OVLAY   0,'TF',IRM',!8M!                                                E.2
*A*OVLAY8 OVLAY   0,'TF',IRM',!7M!                                                E.2
*A*OVLAY7 OVLAY   0,'TF',IRM',!6M!                                                E.2
*A*OVLAY6 OVLAY   0,'TF',IRM',!5M!                                                E.2
*A*OVLAY5 OVLAY   0,'TF',IRM',!4M!                                                E.2
*A*OVLAY4 OVLAY   0,'TF',IRM',!3M!                                                E.2
*A*OVLAY3 OVLAY   0,'TF',IRM',!2M!                                                E.2
*A*OVLAY2 OVLAY   0,'TF',IRM',!1M!                                                E.2
```

```
    .INSERT,12
    .REPLACE,22                          COMPAR    TCM NO    05/01/75  MADISON   E.2
*0*         DIMENSION STRING( 54), POOL  (291)
*A*         DIMENSION STRING( 56), POOL (302)
    .INSERT,290                          COMPAR    TCM NO    05/01/75  MADISON   E.2
*A*C
*A*C------STRING 54   3HTCM
*A*         DATA    STRING( 55)/ 292/, POOL (292)/  3/
*A*         DATA    POOL(293)/2H T/ ,  POOL (294)/2H C/
*A*         DATA    POOL(295)/2H M/
*A*C
*A*C------STRING 55   6HBAUDOT
*A*         DATA    STRING( 56)/ 296/, POOL (296)/   6/
*A*         DATA    POOL(297)/2H B/ ,  POOL (298)/2H A/
*A*         DATA    POOL(299)/2H U/ ,  POOL (300)/2H D/
*A*         DATA    POOL(301)/2H O/ ,  POOL (302)/2H T/
```

```
    .INSERT,12
    .REPLACE,544                    PARSE     BAUDOT    05/01/75  KADISON   E.2
*D*         CALL DTAG
*A*         IF (ITEST) 5311, 70, 5311
*A*5311     CONTINUE
    .INSERT,546                     PARSE     BAUDOT    05/01/75  KADISON   E.2
*A*C          (  'BAUDOT '    )
*A*  70     CALL COMPAR (55)
*A*         CALL DTAG
*A*         CALL BITSET(TCD(3), 15, 12, 1)
*A*         GO TO 69
    .REPLACE,630                    PARSE     TCM NO    05/01/75  KADISON   E.2
*D*         CALL DTAG
*A*         IF (ITEST) 5381, 845, 5381
*A*5381     CONTINUE
    .INSERT,632                     PARSE     TCM NO    05/01/75  KADISON   E.2
*A*C          ('TCM'      )
*A*  845    CALL COMPAR(54)
*A*         CALL DTAG
*A*C         USE NUMBER
*A*         CALL GETCHR
*A*         J = INCHAR - 176
*A*         IF( J .LT. 0 .OR. J .GT. 9 ) GO TO 8451
*A*         CALL ADVANC
*A*         CALL BITSET( TCD(3), 11, 8, J)
*A*         GO TO 69
*A*8451     ITEST = 0
*A*         CALL DTAG
```

```
.REPLACE,122,169                VTAMTABL MACROS   05/01/75 MADISON  E.2
*D*SUBAT  MAC
*D*       TEF     P(1),11,11
*D*       SUB1    P(1)
*D*       TFT     9,P(1),10
*D*       SUB     TEN
*D*       TFT     8,P(1),9
*D*       SUB     NINE
*D*       TFT     7,P(1),8
*D*       SUB     EIGHT
*D*       TFT     6,P(1),7
*D*       SUB     SEVEN
*D*       TFT     5,P(1),6
*D*       SUB     SIX
*D*       TFT     4,P(1),5
*D*       SUB     FIVE
*D*       TFT     3,P(1),4
*D*       SUB     FOUR
*D*       TFT     2,P(1),3
*D*       SUB     THREE
*D*       TFT     1,P(1),2
*D*       SUB     TWO
*D*       TFT     0,P(1),1
*D*       SUB     ONE
*D*       EMAC
*D*ADAT   MAC                   ADD ATTRIBUTE DESCRIBED BY P(1), AND P(2)
*D*       TEF     P(1),11,11
*D*       ADD1    P(1)
*D*       TFT     9,P(1),10
*D*       ADD     TEN
*D*       TFT     8,P(1),9
*D*       ADD     NINE
*D*       TFT     7,P(1),8
*D*       ADD     EIGHT
*D*       TFT     6,P(1),7
*D*       ADD     SEVEN
*D*       TFT     5,P(1),6
*D*       ADD     SIX
*D*       TFT     4,P(1),5
*D*       ADD     FIVE
*D*       TFT     3,P(1),4
*D*       ADD     FOUR
*D*       TFT     2,P(1),3
```

```
*D*         ADD     THREE
*D*         TFT     1,P(1),2
*D*         ADD     TWO
*D*         TFT     0,P(1),1
*D*         ADD     ONE
*D*         EMAC
*A*ERAT     MAC     VALUE           ERA ATTRIBUTE                          E.2
*A*         GC      P(1),013        GENERATE  ERA INST                     E.2
*A*         EMAC                                                           E.2
*A*ANAT     MAC     VALUE           ANA ATTRIBUTE                          E.2
*A*         GC      P(1),015        GENERATE  ANA INST                     E.2
*A*         EMAC                                                           E.2
*A*ORAT     MAC     VALUE           ORA ATTRIBUTE                          E.2
*A*         GC      P(1),011        GENERATE  ORA INST                     E.2
*A*         EMAC                                                           E.2
*A*ADAT     MAC     VALUE           ADD ATTRIBUTE                          E.2
*A*         GC      P(1),012        GENERATE  ADD INST                     E.2
*A*         EMAC                                                           E.2
*A*SUBAT    MAC     VALUE           SUBTRACT ATTRIBUTE                     E.2
*A*         GC      P(1),014        GENERATE  SUB INST                     E.2
*A*         EMAC                                                           E.2
*A*LDAT     MAC     VALUE           LOAD ATTRIBUTE                         E.2
*A*         GC      P(1),01         GENERATE  LDA INST                     E.2
*A*         EMAC                                                           E.2
*A*LDBT     MAC     VALUE           LDB ATTRIBUTE                          E.2
*A*         GC      P(1),02         GENERATE  LDB INST                     E.2
*A*         EMAC                                                           E.2
*A*LDXT     MAC     VALUE           LDX ATTRIBUTE                          E.2
*A*         GC      P(1),03         GENERATE  LDX INST                     E.2
*A*         EMAC                                                           E.2
*A*GC       MAC     VALUE,OPCODE    GENERATE VARIABLE INSTRUCTION CODE     E.2
*A*##GEN    SET     0               NO INSTRUC GENERATED                   E.2
*A*         IFT     9,P(1),10       SKIP IF NOT 10                         E.2
*A*         INST    TEN,P(2)        GENERATE                               E.2
*A*         IFT     8,P(1),9        SKIP IF NOT 9                          E.2
*A*         INST    NINE,P(2)       GENERATE                               E.2
*A*         IFT     6,P(1),7        SKIP IF NOT 7                          E.2
*A*         INST    SEVEN,P(2)      GENERATE                               E.2
*A*         IFT     5,P(1),6        SKIP IF NOT 6                          E.2
*A*         INST    SIX,P(2)        GENERATE                               E.2
*A*         IFT     4,P(1),5        SKIP IF NOT 5                          E.2
*A*         INST    FIVE,P(2)       GENERATE                               E.2
*A*         IFT     2,P(1),3        SKIP IF NOT 3                          E.2
```

```
*A*       INSI    THREE,P(2)      GENERATE                                E.2
*A*       IFT     016,P(1),017    SKIP IF NOT 15                          E.2
*A*       INSI    BM17,P(2)       GENERATE                                E.2
*A*       IFT     036,P(1),037    SKIP IF NOT 31                          E.2
*A*       INSI    BM37,P(2)       GENERATE                                E.2
*A*       IFT     076,P(1),077    SKIP IF NOT 63                          E.2
*A*       INSI    BM77,P(2)       GENERATE                                E.2
*A*       IFT     0176,P(1),0177  SKIP IF NOT 127                         E.2
*A*       INSI    BM177,P(2)      GENERATE                                E.2
*A*       IFT     0776,P(1),0777  SKIP IF NOT 511                         E.2
*A*       INSI    BM777,P(2)      GENERATE                                E.2
*A*       IFT     01776,P(1),01777  SKIP IF NOT 1023                      E.2
*A*       INSI    BM1777,P(2)     GENERATE                                E.2
*A*       IFT     P(1),,0177400   SKIP IF NOT LEFT HAND BYTE              E.2
*A*       INSI    LHW,P(2)        GENERATE                                E.2
*A*       IFT     P(1),,0377      SKIP IF NOT RIGHT HAND BYTE             E.2
*A*       INSI    RHW,P(2)        GENERATE                                E.2
*A*       IFT     P(1),,1         SKIP IF 1                               E.2
*A*       GOTO    5               GO IF NOT 1                             E.2
*A*       IFT     P(2),04,04      SKIP IF LOAD REGISTER OPCODE            E.2
*A*       GOTO    1               CONTINUE OTHERWISE                      E.2
*A*       INCK    P(2)            GENERATE QUICK LOAD 1                   E.2
*A*       GOTO    4               AND EXIT                                E.2
*A*1      CONT                    NOT LOAD INST                           E.2
*A*       IFT     P(2)-012        SKIP IF ADD OPCODE                      E.2
*A*       GOTO    2                                                       E.2
*A*       TAR                     ADD 1                                   E.2
*A*       GOTO    4               EXIT                                    E.2
*A*2      CONT                    NOT ADD                                 E.2
*A*       IFT     P(2)-014        SKIP IF SUB OPCODE                      E.2
*A*       GOTO    3                                                       E.2
*A*       DAR                     SUB 1 INST                              E.2
*A*       GOTO    4               EXIT                                    E.2
*A*3      CONT                    NOT A SIMPLE OPCODE                     E.2
*A*       INSI    ONE,P(2)        GENERATE                                E.2
*A*4      CONT                                                            E.2
*A*##GEN  SET     1               SET INSTRUCTION GENERATED               E.2
*A*5      CONT                                                            E.2
*A*       IFT     P(1)            SKIP IF 0                               E.2
*A*       GOTO    1               GO OTHERWISE                            E.2
*A*##GEN  SET     1               SET INST GENERATED                      E.2
*A*       IFT     P(2),04,04      SKIP IF NOT A LOAD OPCODE               E.2
*A*       ZERO    P(2)            GENERATE SIMPLE LOAD 0                  E.2
```

```
*A*          IFF     P(2)-015          SKIP IF NOT ANA OPCODE                    E.2
*A*          TZA                       ANA 0 = 0                                 E.2
*A*1         CONT                                                                E.2
*A*          IFT     ##GEN             SKIP IF NO INSTRUCTION GENERATED          E.2
*A*          GOTO    DONE              EXIT IF GEND                              E.2
*A*          FIND    P(1),0,0,0         CHECK FOR POSSIBLE MASK VALUE            E.2
*A*          IFT     ##R,0             SKIP IF FOUND                             E.2
*A*          GOTO    ASSORT            GO IF NOT                                 E.2
*A*##ST      SET     ##ST+##NEG        SET STATE OF RESULT FLAG                  E.2
*A*##R       SET     BS0+##R           CALCULATE MASK ADDRESS                    E.2
*A*          IFT     ##ST              SKIP IF BS TYPE MASK                      E.2
*A*##R       SET     ##R+BR0-BS0       ADJUST TO BR TYPE MASK ADDRESS            E.2
*A*          INST    ##R,P(2)          GENERATE INSTRUCTION                      E.2
*A*ASSORT    CONT                      CANT GENERATE 1 WORD INST                 E.2
*A*          INST    P(1),P(2),1           GENERATE IMMIDIATE INSTRUCTION        E.2
*A*DONE      CONT                                                                E.2
*A*          EMAC                                                                E.2
*A*INST      MAC     ADDR,OPCODE,TYPE  GEN REQUESTED INST AND ADDR CODE          E.2
*A*          IFT     ##GEN             SKIP IF NOT ALREADY GENERATED             E.2
*A*          GOTO    2                 EXIT OTHERWISE                            E.2
*A*          IFF     P(3)              SKIP IF TWO WORD INST REQUIRED            E.2
*A*          GOTO    1                 GO GENERATE 1 WORD INSTRUCTION            E.2
*A*          DATA    P(2)*8+06000,P(1)              XXXT    VALUE                E.2
*A*          GOTO    2                 EXIT                                      E.2
*A*1         CONT                                                                E.2
*A*##INST    FORM    4,3,9                                                       E.2
*A*          ##INST  P(2),0,P(1)       GENERATE ABSOLUTE REFERENCE INST          E.2
*A*2         CONT                                                                E.2
*A*##GEN     SET     1                 SET INSTRUCTION GENERATED FLAG            E.2
*A*          EMAC                                                                E.2
*A*FIND      MAC     VALUE,STATE,ITERATION,FLAG  CHECK FOR BS OR BR MASK         E.2
*A*          IFT     P(1),,0100000     SKIP IF BS15                              E.2
*A*          GOTO    START             START OTHERWISE                           E.2
*A*##R       SET     15                RESULT TO BIT 15                          E.2
*A*##ST      SET     0                 TEST STATE = 0                            E.2
*A*##NEG     SET     0                 USE BS MASK                               E.2
*A*          GOTO    EXIT              EXIT                                      E.2
*A*START     CONT                                                                E.2
*A*##VA      SET     P(1)              INITIALIZE VA                             E.2
*A*          IFT     P(1),0            SKIP IF VALUE POSITIVE                    E.2
*A*##VA      SET     -##VA-1           2S COMPLEMENT VA                          E.2
*A*##V       SET     ##VA-##VA/2*2     GET LOW ORDER BIT                         E.2
*A*          IFT     P(3)              SKIP IF FIRST ITERATION                   E.2
```

```
*A*        GOTO    NFRST                                                  E.2
*A*##R     SET     -1              RESULT NOT FOUND                       E.2
*A*##ST    SET     ##V             TEST STATE = STATE OF BIT 0            E.2
*A*##NEG   SET     0               NOT NEGATIVE                           E.2
*A*        IFT     ##VA,,P(1)      SKIP IF VA = VALUE                     E.2
*A*##NEG   SET     1               SET NEG FLAG IF VALUE WAS NEGATIVE     E.2
*A*        GOTO    RECURS          GO RECURSE  NEXT LEVEL                 E.2
*A*NFRST   CONT                    NOT FIRST ENTRY                        E.2
*A*        IFT     P(3)-1          SKIP IF SECOND ITERATION               E.2
*A*        GOTO    NSECND                                                 E.2
*A*        IFF     ##V-P(2)        SKIP IF BITS 0 & 1 ARENT SAME          E.2
*A*        GOTO    RECURS          RECURSE A LEVEL IF SAME                E.2
*A*        FIND    ##VA/2,##ST,P(3)+1,1  MAKE FLAG TEST FOR RES STATE     E.2
*A*        GOTO    EXIT            THEN LEAVE                             E.2
*A*NSECND  CONT                    NOT SECOND ITERATION                   E.2
*A*        IFF     P(4)            SKIP IF FLAG SET                       E.2
*A*        GOTO    TEST                                                   E.2
*A*        IFF     ##V-P(2)        SKIP IF BIT NOT = BIT 0                E.2
*A*##R     SET     1               CALL BIT 1 THE DIFFERENT ONE           E.2
*A*        IFT     ##V-P(2)        SKIP IF BIT 2 = BIT 0                  E.2
*A*##R     SET     0               SET BIT 0 THE DIFFERENT ONE            E.2
*A*##ST    SET     ##V             TEST STATE = BIT 2 STATE               E.2
*A*        GOTO    RECURS          GO RECURSE NEXT LEVEL                  E.2
*A*TEST    CONT                    NOT FLAG TEST                          E.2
*A*        IFF     ##V-P(2)        SKIP IF TEST BIT NOT = TEST STATE      E.2
*A*        GOTO    RECURS          CONTINUE RECURSING IF SAME             E.2
*A*        IFT     ##R,0           SKIP IF RESULT ALREADY SET             E.2
*A*        GOTO    RESULT          SAVE THIS AS RESULT                    E.2
*A*##R     SET     -1              VOID RESULT - 2 BITS DIFFER            E.2
*A*        GOTO    OUT             EXIT                                   E.2
*A*RESULT  CONT                                                           E.2
*A*##R     SET     P(3)            SET RESULT TO ITERATION COUNTER        E.2
*A*RECURS  CONT                                                           E.2
*A*        IFT     P(3)-15         SKIP IF 15 ITERATIONS                  E.2
*A*        FIND    ##VA/2,##ST,P(3)+1,0    RECURSE TESTING NEXT BIT       E.2
*A*EXIT    CONT                                                           E.2
*A*OUT     CONT                                                           E.2
*A*        FMAC                                                           E.2
   .INSERT,246                     VTAMTABL  V75                 MEIR     E.2
*A*SAVER   MAC                                                            E.2
*A*        ST,3    P(2),P(1)                                              E.2
*A*        ST,4    P(2)+1,P(1)                                            E.2
*A*        ST,5    P(2)+2,P(1)                                            E.2
```

```
*A*        ST,6     P(2)+3,P(1)                                          E.2
*A*        ST,7     P(2)+4,P(1)                                          E.2
*A*        EMAC                                                          E.2
   .INSERT,281                    VTAMTABL                    MEIR       E.2
*A*TBEND   SET      TBRSE+1                                              E.2
*A*        IFF      VORTEX-2                                             E.2
*A*TBEND   SET      TBIST+1                                              E.2
*A*TBRSR3  EQU      TBEND+0       R3        REENTRANT AND SUSPEND STACK  E.2
*A*TBRSR4  EQU      TBEND+1       R4        REENTRANT AND SUSPEND STACK  E.2
*A*TBRSR5  EQU      TBEND+2       R5        REENTRANT AND SUSPEND STACK  E.2
*A*TBRSR6  EQU      TBEND+3       R6        REENTRANT AND SUSPEND STACK  E.2
*A*TBRSR7  EQU      TBEND+4       R7        REENTRANT AND SUSPEND STACK  E.2
*A*TBISR3  EQU      TBEND+5       R3        INTERRUPT STACK              E.2
*A*TBISR4  EQU      TBEND+6       R4        INTERRUPT STACK              E.2
*A*TBISR5  EQU      TBEND+7       R5        INTERRUPT STACK              E.2
*A*TBISR6  EQU      TBEND+8       R6        INTERRUPT STACK              E.2
*A*TBISR7  EQU      TBEND+9       R7        INTERRUPT STACK              E.2
   .REPLACE,319                   VTAMTABL                    MEIR       E.2
*D**       BIT 14  -  UNUSED                                              *
*A**                BIT 14 V75 FLAG U#V75 0# NOT A V75 MACHINE           E.2
*A*TB75    EQU      2             WORD 2                                 E.2
*A*TB75B   EQU      14            BIT 14                                 E.2
*A*TB75Z   EQU      1             1 BIT LONG                             E.2
   .INSERT,900                    VTAMTABL  SMR-779           MEIR       E.2
*A***      POL          6              7-7  BSC POLL MODE                E.2
*A*LCPOL   EQU      6                                                    E.2
*A*LCPOLB  EQU      7                                                    E.2
*A*LCPOLZ  EQU      1                                                    E.2
   .INSERT,920                    VTAMTABL  SMR-779           MEIR       E.2
*A***      IGN          7             10-10 IGNORE BYTE COUNT 0 INTRPT   E.2
*A*LCIGN   EQU      7                                                    E.2
*A*LCIGNB  EQU      10                                                   E.2
*A*LCIGNZ  EQU      1                                                    E.2
   .INSERT,1054                   VTAMTABL                    MEIR       E.2
*A***      POL         16              15-15 BSC POLL MODE               E.2
*A*LSPOL   EQU      16                                                   E.2
*A*LSPOLB  EQU      15                                                   E.2
*A*LSPOLZ  EQU      1                                                    E.2
   .INSERT,1160                   VTAMTABL                    MEIR       E.2
*A***      IBC          6              13-13  INPUT BUFFER CHAINING FLAG E.2
*A*TCIBC   EQU      6                                                    E.2
*A*TCIBCB  EQU      13                                                   E.2
*A*TCIBCZ  EQU      1                                                    E.2
```

```
   .TNSERT,1192                    VTAMTABL                    MEIR      E.2
*A***      CHR      15                  0-15  CHR (ONLY FOR BUF CHAIN)   E.2
*A*TCCHR   EQU      15                                                   E.2
*A*TCCHRB  EQU      0                                                    E.2
*A*TCCHRZ  EQU      16                                                   E.2
*A***         ACF       15              0-15  ACTIVE CHAIN FRONT POINTER E.2
*A*TCACF   EQU      15                                                   E.2
*A*TCACFB  EQU      0                                                    E.2
*A*TCACFZ  EQU      16                                                   E.2
*A***         ACR       16              0-15  ACTIVE CHAIN REAR POINTER  E.2
*A*TCACR   EQU      16                                                   E.2
*A*TCACRB  EQU      0                                                    E.2
*A*TCACRZ  EQU      16                                                   E.2
*A***      CCF      17                  0-15  COMPLETED CHAIN FRONT PTR  E.2
*A*TCCCF   EQU      17                                                   E.2
*A*TCCCFB  EQU      0                                                    E.2
*A*TCCCFZ  EQU      16                                                   E.2
*A***      CCR      18                  0-15  COMPLETED CHAIN REAR PTR   E.2
*A*TCCCR   EQU      18                                                   E.2
*A*TCCCRB  EQU      0                                                    E.2
*A*TCCCRZ  EQU      16                                                   E.2
*A***         NULL      19              0-15  RESERVED FOR CCM USE       E.2
*A**                                                                     E.2
*A**       END OF BUFFER CHAINING CHR                                    E.2
   .TNSERT,1247                    VTAMTABL                    MEIR      E.2
*A**************                                                         E.2
*A***              AN EXTENSION TO TCM CTB IS USED BY VT$TCQ AS          E.2
*A***              A TEMPORARY STORAGE                                   E.2
*A***                                                                    E.2
*A*CTRCA   EQU      16       FOTR TCRCA IN TCD                           E.2
*A*CTWCA   EQU      17       FOR TCWCA IN TCD                            E.2
*A*CTCDA   EQU      18       FOR TCD ADDRESS                             E.2
*A*CTRQS   EQU      19       TCM REQUEST ADDRESS                         E.2
*A*CTUTDB  EQU      20       USER'S TIDB ADDRESS                         E.2
*A*CTMFM   EQU      21       12 WORDS FOR CCM RQST BLOCK                 E.2
*A**************                                                         E.2
*A*CTFNT   EQU      33             0 FUNC 1, 1 FUNC 5                    E.2
```

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
$ 0	FPPWCS	000000
$ TOMP	FPPWCS	000200
$2K	DTADDSUB	000106
$2L	DTADDSUB	000037
$2M	DTMULDIV	000076
$2N	DTMULDIV	000153
$2NX	DTMULDIV	000131
$3S	$3S	000005
$4C	$4C	000051
$4E	$4E	000000
$6D	XDCOMP	000000
$6E	$6E	000160
$6K	DTADDSUB	000074
$6L	DTADDSUB	000030
$6M	DTMULDIV	000066
$6N	DTMULDIV	000302
$6NX	DTMULDIV	000175
$8F	V$8G	000063
$8G	V$8G	000013
$8K	CCMATH	000023
$8L	CCMATH	000056
$8M	CCMATH	000132
$8N	CCMATH	000235
$8S	V$8G	000042
$9E	$9E	000000
$AC	REAL	000057
$AK	CRMATH	000011
$AL	CRMATH	000032
$AM	CRMATH	000061
$AN	CRMATH	000110
$BA	VSFORTIO	000016
$CAC	AC	000004
$CACT	AC	000006
$CG	$CG	000012
$D2	$D2	000000
$DAD	DADDSUB	000354
$DAN	DATAN	000014
$DATE	V	000053
$DBUF	CCSMDL	000200
$DCO	DSINCOS	000031
$DDT	DDIVIDE	000313
$DE	$DE	000000

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
$DE	$DE	000000
$DEX	DEXP	000020
$DFR	$DFR	000052
$DIT	IDINT	000034
$DLN	DLOG	000020
$DLO	DLOADAC	000021
$DMP	DMULT	000210
$DNO	DNORMAL	000062
$DO	$DO	000000
$DQ	VSFORTIO	000121
$DRM1	CCSMDL	000000
$DRM1A	CCSMDL	000040
$DRM1C	CCSMDL	000100
$DRM1D	CCSMDL	000140
$DRM2	CCSMDL	000020
$DST	DSINCOS	000041
$DSQ	DSQRT	000165
$DSU	DADDSUB	000357
$DX	VSFORTDA	000000
$FE	VSEF	000000
$FLOC	V	000054
$FN	VSFORTIO	000024
$FO	VSFORTIO	000127
$FR	STOPALL	000000
$FC	$FC	000101
$FMS	SFPMANTI	000000
$FSM	SFPMANTT	000000
$FX	VSFORTDA	000182
$GC	$GC	000000
$HC37	VSHC37	000000
$HE	$HE	000103
$HH	$HMH	000036
$HN	$HNH	000027
$HS	$IFIX	000046
$HS2	$FC	000054
$I1	VSFORTIO	000405
$I2	VSFORTIO	000413
$I3	VSFORTIO	000421
$I4	VSFORTIO	000427
$I5	VSFORTIO	000435
$I6	VSFORTIO	000443
$IC	$IFIX	000000

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
$IC	$IFIX	000000
$JC	$JC	000000
$JOB	V	000060
$LUT	V	000061
$MPP	VSMPR	000043
$NO	VSFORTIO	000000
$NML	FNORMAL	000000
$OC	REAL	000140
$PA	STOPAUS	000003
$PC	$FLOAT	000000
$PE	$PE	000113
$QE	$QE	000042
$QK	FADDSUB	000207
$QL	FADDSUB	000212
$QM	FMULDIV	000160
$QN	FMULDIV	000163
$QS	$FLOAT	000043
$QS2	$IC	000075
$RC	SINGLE	000021
$RD	VSFORTIO	000361
$RE	VSFORTIO	000032
$RTENM	$RTENM	000000
$RX	VSFORTDA	000176
$SE	$SE	000036
$ST	STOPAUS	000007
$STO	DSTOREAC	000021
$SYST	V	000000
$TC	$TC	000104
$TC01	VSTC01	000000
$TE	$TE	000162
$WR	VSFORTIO	000367
$WX	VSFORTDA	000204
$XC	$XC	000000
$XE	$XE	000000
$YC	DOUBLE	000024
$YE	$YE	000000
$YK	$YK	000000
$YL	$YL	000000
$YM	$YM	000000
$YN	$YN	000000
$ZC	DBLECOMP	000007
$ZD	REAL	000101

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
$ZD	REAL	000103
$ZE	$ZE	000000
$7F	DLOADAC	000021
$7I	RLOADAC	000000
$7K	DADDSUB	000354
$7L	DADDSUB	000357
$7M	DMULT	000210
$7N	DDIVIDE	000313
$7S	DSTOREAC	000021
A$C	AC	000000
A2MT	A2MT	000104
ABORT	ABORT	000027
ABS	ABS	000000
AC	AC	000000
AC1	AC	000001
AC2	AC	000002
AC3	AC	000003
ADVANC	ADVANC	000000
ATMAG	REAL	000114
ATNT	ATNT	000000
ATRD	V$DPIO	000327
ATRDW	V$DPIO	000333
ATSD	V$DPIO	000317
ATSDW	V$DPIO	000323
ALOC	ALOC	000022
ALOG	ALOG	000165
ALOG10	ALOG10	000000
AMAX0	AMAX0	000000
AMAX1	AMAX1	000000
AMIN0	AMIN0	000000
AMIN1	AMIN1	000000
AMOD	AMOD	000000
AO	V$DPIO	000347
AOW	V$DPIO	000353
APPEND	APPEND	000132
ATAN	ATAN	000145
ATAN2	ATAN2	000000
AXIS	AXISV	000000
BACKUP	BACKUP	000000
BTAS	V$BTCA	000055
BIFCR	V$GFCRS	000161
BTRS	V$BTCA	000026

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
BTS$	V$BTCA	000043
BTTGET	BTTSET	000006
BTTSET	BTTSET	000000
BL$P	VBLP	000000
BOFCB	VSGFCBS	000173
BTOA	MERGE	004206
C52CIH	C52CIH	000000
C52FUN	C52FUN	000000
C52TWP	C52TWP	000000
C52LIP	C52LIP	000000
C52PCR	C52PCR	000000
C52RCV	C52RCV	000000
C52PCW	C52RCW	000000
C52SST	C52SST	000000
C52XMT	C52XMT	000000
CA2B	CA2B	000000
CABS	CSQRT	000113
CB2A	CB2A	000000
CC$ACF	CC$ACF	000000
CC$CBF	CC$CBS	000051
CC$CBS	CC$CBS	000000
CC$CEX	CC$CEX	000000
CC$CRO	CC$CRO	000010
CC$FCW	CC$FCW	000000
CC$FRR	CC$FRR	000007
CC$FWR	CC$FRR	000000
CC$OCL	CC$OCI	000000
CC$ORO	CC$CRO	000000
CC$RCF	CC$CEX	000326
CC$SCW	CC$SCW	000000
CCK$$	V$CRIF	000011
CCOS	CSINCCOS	000022
CEXP	CLOGCEXP	000141
CHAR	CHARM	000160
CHEB	CHEB	000140
CLEAR	CLEAR	000000
CLOG	CLOGCEXP	000041
CLOSE	CLOSE	000000
CL$$	MERGE	003350
CMPLX	REAL	000014
COMP	MERGE	003452
COMPAR	COMPAR	000000

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
CONC	VSCONC	000000
CONJG	REAL	000047
COS	SINCOS	000006
CROOA	VSCROOA	000000
CROSS	VSCRIF	000025
CSIN	CSINCCOS	000212
CSQRT	CSQRT	000055
CTCTOA	CTCTOA	000000
CTCT1A	CTCT1A	000000
CTCT2A	CTCT2A	000000
CTCT3A	CTCT3A	000000
CTCT4A	CTCT4A	000000
CTCT5A	CTCT5A	000000
CTCT6A	CTCT6A	000000
CTCT7A	CTCT7A	000000
CTCOOA	CTCOOA	000000
CTCO1A	CTCO1A	000000
CTCO2A	CTCO2A	000000
CTCO3A	CTCO3A	000000
CTCO4A	CTCO4A	000000
CTCO5A	CTCO5A	000000
CTCO6A	CTCO6A	000000
CTCO7A	CTCO7A	000000
CTCPOA	CTCPOA	000000
CTCROA	CTCROA	000000
CTCTOA	CTCTOA	000000
CTCT1A	CTCT1A	000000
CTCT2A	CTCT2A	000000
CTCT3A	CTCT3A	000000
CTCT4A	CTCT4A	000000
CTCT5A	CTCT5A	000000
CTCT6A	CTCT6A	000000
CTDOA	CTDOA	000000
CTDOB	CTDOB	000000
CTDOC	CTDOC	000000
CTDOD	CTDOD	000000
CTD1A	CTD1A	000000
CTD1B	CTD1B	000000
CTD1C	CTD1C	000000
CTD1D	CTD1D	000000
CTD2A	CTD2A	000000
CTD2B	CTD2B	000000

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
CTD2C	CTD2C	000000
CTD2D	CTD2D	000000
CTD3A	CTD3A	000000
CTD3B	CTD3B	000000
CTD3C	CTD3C	000000
CTD3D	CTD3D	000000
CTIME	CTIME	000000
CTLP0A	CTLP0A	000000
CTLP0D	CTLP0D	000000
CTLP0F	CTLP0F	000000
CTLP0G	CTLP0G	000000
CTLP0H	CTLP0H	000000
CTLP0J	CTLP0J	000000
CTLP1A	CTLP1A	000000
CTLP1D	CTLP1D	000000
CTLP1F	CTLP1F	000000
CTLP1G	CTLP1G	000000
CTLP1H	CTLP1H	000000
CTLP1J	CTLP1J	000000
CTLP2A	CTLP2A	000000
CTLP2D	CTLP2D	000000
CTLP2F	CTLP2F	000000
CTLP2G	CTLP2G	000000
CTLP2H	CTLP2H	000000
CTLP2J	CTLP2J	000000
CTLP3A	CTLP3A	000000
CTLS	MERGE	002615
CTMT0A	CTMT0A	000000
CTMT1A	CTMT1A	000000
CTMT2A	CTMT2A	000000
CTMT3A	CTMT3A	000000
CTMX0A	CTMX0A	000000
CTMX1A	CTMX1A	000000
CTMX2A	CTMX2A	000000
CTMX3A	CTMX3A	000000
CTPT0A	CTPT0A	000000
CTPT1A	CTPT1A	000000
CTSP0A	CTSP0A	000000
CTSP1A	CTSP1A	000000
CTSP2A	CTSP2A	000000
CTSP3A	CTSP3A	000000
CTSP4A	CTSP4A	000000

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
CTSP5A	CTSP5A	000000
CTSP6A	CTSP6A	000000
CTSP7A	CTSP7A	000000
CTTC0A	CTTC0A	000000
CTTY0A	CTTY0A	000000
CTTY1A	CTTY1A	000000
CTTY2A	CTTY2A	000000
CTTY3A	CTTY3A	000000
CTTY4A	CTTY4A	000000
CTTY5A	CTTY5A	000000
CTTY6A	CTTY6A	000000
CTWCS	CTWCS	000107
CUT	PLOTS	000020
D00A1	VSD00A1	000000
D00A2	VSD00A2	000000
D00A5	VSD00A5	000000
D00B	VSD00B	000000
D00C	VSD00C	000000
D00D	VSD00D	000000
D01B	VSD01B	000000
D01C	VSD01C	000000
D01D	VSD01D	000000
D02B	VSD02B	000000
D02C	VSD02C	000000
D02D	VSD02D	000000
D03B	VSD03B	000000
D03C	VSD03C	000000
D03D	VSD03D	000000
D0BCT	VSD0BCT	000000
D0CCT	VSD0CCT	000000
D0DCT	VSD0DCT	000000
D10A1	VSD10A1	000000
D10A2	VSD10A2	000000
D10A5	VSD10A5	000000
D10B	VSD10B	000000
D10C	VSD10C	000000
D10D	VSD10D	000000
D11B	VSD11B	000000
D11C	VSD11C	000000
D11D	VSD11D	000000
D12B	VSD12B	000000
D12C	VSD12C	000000

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
D12D	VSD12D	000000
D13B	VSD13B	000000
D13C	VSD13C	000000
D13D	VSD13D	000000
D1BCT	VSD1BCT	000000
D1CCT	VSD1CCT	000000
D1DCT	VSD1DCT	000000
D20A1	VSD20A1	000000
D20A2	VSD20A2	000000
D20A5	VSD20A5	000000
D20B	VSD20B	000000
D20C	VSD20C	000000
D20D	VSD20D	000000
D21B	VSD21B	000000
D21C	VSD21C	000000
D21D	VSD21D	000000
D22B	VSD22B	000000
D22C	VSD22C	000000
D22D	VSD22D	000000
D23B	VSD23B	000000
D23C	VSD23C	000000
D23D	VSD23D	000000
D2BCT	VSD2BCT	000000
D2CCT	VSD2CCT	000000
D2DCT	VSD2DCT	000000
D30A1	VSD30A1	000000
D30A2	VSD30A2	000000
D30A5	VSD30A5	000000
D30B	VSD30B	000000
D30C	VSD30C	000000
D30D	VSD30D	000000
D31B	VSD31B	000000
D31C	VSD31C	000000
D31D	VSD31D	000000
D32B	VSD32B	000000
D32C	VSD32C	000000
D32D	VSD32D	000000
D33B	VSD33B	000000
D33C	VSD33C	000000
D33D	VSD33D	000000
D3BCT	VSD3BCT	000000
D3CCT	VSD3CCT	000000

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
D3DCT	V$D3DCT	000000
DABS	DABS	000000
DASMR	V$DASMR	016532
DATA	DATAM	000000
DATAN	DATAN	000000
DATAN2	DATAN2	000000
DBLE	DBLE	000000
DCORE	DMEMRY	000451
DCOS	DSINCOS	000015
DCREAD	V$DMC	000276
DCREWD	V$DMC	000024
DCSKRD	V$DMC	000065
DCSTAT	V$DMC	000000
DCWRIT	V$DMC	000274
DDREAD	V$DMD	000240
DDREWD	V$DMD	000024
DDSKRD	V$DMD	000065
DDSTAT	V$DMD	000000
DDWRIT	V$DMD	000236
DEBUG	V$DEBUG	000000
DELAY	DELAY	000035
DEXP	DEXP	000013
DI	V$DPIO	000337
DIAG	DIAG	000000
DIM	DIM	000000
DIM$	VDIM	000000
DINT	DINT	000000
DISP1	V$FUNC	000000
DIW	V$DPIO	000343
DLOG	DLOG	000012
DLOG10	DLOG10	000000
DMAX1	DMAX1	000000
DMEMRY	DMEMRY	000000
DMIN1	DMIN1	000000
DMOD	DMOD	000000
DMREAD	V$DMAF	000124
DMREWD	V$DMAF	000056
DMRTTF	V$DMAF	000130
DMSFIL	V$DMAF	000330
DMSKRD	V$DMAF	000064
DMSTAT	V$DMAF	000000
DMWEOF	V$DMAF	000327

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
DDL	VSDPID	000357
DDLW	VSDPID	000363
DDM	VSDPID	000367
DDMW	VSDPID	000423
DPCLOS	PLOTS	000210
DPINIT	PLOTS	000256
DPPLOT	DPPLOT	000000
DPSORT	DPSORT	000000
DRREAD	VSDRAF	000101
DRREWD	VSDRAF	000033
DRRTTF	VSDRAF	000105
DRSFIL	VSDRAF	000252
DRSKRD	VSDRAF	000041
DRSTAT	VSDRAF	000000
DRWEOF	VSDRAF	000251
DSIGN	DSIGN	000000
DSIN	DSINCOS	000000
DSQRT	DSQRT	000014
DUMMYT	DUMMYT	000000
EMEM	MTCSIM	000177
ENDCUT	PLOTS	000041
EXC	MTCSIM	000715
EXIT	EXIT	000000
EXP	EXP	000146
FACTOR	PLOT	000011
FCHS	VSFCHS	000000
FFTCH	CA2R	000142
FLOAT	FLOAT	000000
FMAIN	VSFMAIN	000000
FMBA	VSFMAIN	001061
FMBF	VSFMAIN	005667
FMCR	VSFMAIN	001451
FMEN	VSFMAIN	002330
FMLF	VSFMAIN	003320
FMMV	VSFMAIN	003735
FMPT	VSFMAIN	004457
FMRR	VSFMAIN	004722
FMVH	VSFMAIN	005115
FMWI	VSFMAIN	005342
FMWLI	VSFMAIN	005166
FMWR	VSFMAIN	005611
FORT	VSFORT	001005

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
GETCHR	GETCHR	000000
GETLSD	GETLSD	000000
GOFCB	VSGFCBS	000217
HS871	HSCOMN	000042
HS872	HSCOMN	000043
HS873	HSCOMN	000044
HSCOMB	HSCOMN	000001
HSCPMN	HSCOMN	000040
HSINST	HSCOMN	000036
HSMAP	HSCOMN	000000
HSMAPI	HSCOMN	000046
HSMAPN	HSCOMN	000047
HSNTDS	HSCOMN	000003
HSNODS	HSCOMN	000006
HSNPAG	HSCOMN	000045
HSNPRS	HSCOMN	000004
HSNPUS	HSCOMN	000005
HSODST	HSCOMN	000035
HSOTST	HSCOMN	000034
HSOPCU	HSCOMN	000002
HSOUST	HSCOMN	000037
HSPRST	HSCOMN	000032
HSPUST	HSCOMN	000033
HSRCVQ	HSCOMN	000011
HSRDST	HSCOMN	000031
HSSFLG	HSSFLG	000000
HSTBSZ	HSCOMN	000041
HSXMTQ	HSCOMN	000007
HSXQLN	HSCOMN	000010
HCKS	HCKS	000000
HEADER	HEADER	000000
ISDA	ISDA	000600
ISDB	ISDB	000600
ISDC	ISDC	000600
ISDD	ISDD	000600
ISFA	ISFA	000022
IABS	IABS	000000
IAND	VSIAND	000011
IDIM	IDIM	000000
IDINT	IDINT	000034
IEOR	VSIEOR	000011
IF	IF	000038

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
IFCA	MERGE	003146
IFCB	MERGE	003147
IFCLS	MERGE	003204
IFIX	IFIX	000000
IFLAG	V$FUNC	002524
IFRD	MERGE	003160
INLINK	V	000040
INT	INT	000000
INTR	V	000043
INTRST	V	000023
IOAS	V$IOAS	000022
IOCHK	V$FORTIO	000070
IOOK	V$IOOK	000000
IOR	V$IOR	000011
ISHFT	V$ISHF	000036
ISIGN	ISIGN	000020
KNT$	V$KNTS	000000
LAS$	VLAS	000000
LCK$	V$LPD24	000011
LDELAY	LDELAY	000036
LINE	LINEV	000000
LMGEN	V$LMGEN	000000
LOC10	V$FUNC	001776
LOC11	V$FUNC	001777
LOC2	V$FUNC	001766
LOC3	V$FUNC	001767
LOC4	V$FUNC	001770
LOC5	V$FUNC	001771
LOC6	V$FUNC	001772
LOC7	V$FUNC	001773
LOC8	V$FUNC	001774
LOC9	V$FUNC	001775
LOFCB	V$GFCBS	000147
LPOOA	V$LPOOA	000000
LPOOD	V$LPOOD	000000
LPOOE	V$LPOOE	000000
LPL$	V$LPD24	000000
LU5	MERGE	003377
LU5R	MERGE	003402
LWR$	V$LPD24	000043
MAXO	MAXO	000000
MAX1	MAX1	000000

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
MAX1	MAX1	000000
MCK$	V$MTAF	000002
MEF$	V$MTAF	000731
MTDAS	MTDAS	010464
MTN0	MTN0	000000
MTN1	MTN1	000000
MTUTIL	MTUTIL	000002
MLTPLF	PLOT	000111
MMEM	MTCSIM	000000
MOD	MOD	000000
MOVF	MOVF	000000
MPCODF	V$FUNC	001765
MRD$	V$MTAF	000605
MRW$	V$MTAF	000745
MSF$	V$MTAF	000765
MSR$	V$MTAF	000774
MT$0	VMT0	000000
MT$1	VMT1	000000
MT00A	V$MT00A	000000
MT01A	V$MT01A	000000
MT10A	V$MT10A	000000
MT11A	V$MT11A	000000
MT2A	MT2A	000103
MWR$	V$MTAF	000725
NOT	V$NOT	000006
NUMBER	NUMBRV	000000
OABORT	V$OABORT	000000
OASSGN	V$OASSGN	000000
OCOUNT	V$OPCOM	000035
OFCA	MERGE	003216
OFCB	MERGE	003217
OFCI S	MERGE	003254
OIOLST	V$IOLST	000000
OPEN	OPEN	000000
OPPRO	V$OPPRO	000000
ORIG	PLOT	000143
OSCHD	V$OSCHD	000000
OSCLR	V$OPCOM	000014
OSERDO	V$OPCOM	000006
OSERR	V$OPCOM	000006
OSTAT	V$OSTAT	000000
OTIME	V$OTIME	000000

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
OTSCH	V$OTSCH	000000
OVLAY	OVLAY	000067
PARSE	PARSE	000000
PASS	PASS	000023
PIFCB	V$GFCBS	000135
PLOT	PLOT	000302
PLOTS	PLOTS	000312
PMSK	PMSK	000040
POFCB	V$GFCBS	000231
POLY	POLY	000051
PTOOA	V$PTOOA	000000
PUTLSD	PUTLSD	000000
PUTTCD	PUTTCD	000000
R$X	V$FORTIO	000470
RAZI	V$RAZI	000000
RDC$	VRDC	000000
REAL	REAL	000131
REPORT	REPORT	000000
RESUME	RESUME	000027
RPGC	RPGC	004561
RPGRT	RPGRT	011532
S1CA	MERGE	002756
S1CB	MERGE	002757
S1CLS	MERGE	003014
S2CA	MERGE	003026
S2CB	MERGE	003027
S2CLS	MERGE	003064
S3CA	MERGE	003076
S3CB	MERGE	003077
SALMAP	V$SYTASK	003174
SALNUC	V$SYTASK	004373
SCALE	SCALEV	000000
SCHED	SCHED	000056
SEDIT	V$SEDIT	000000
SGEN1	V$SGEN1	003000
SGEN2	V$SGEN2	003000
SGEN3	V$SGEN3	003000
SGEN4	V$SGEN4	003000
SGLDR	V$SGENLD	000000
SGNF	V$OPIO	000405
SGNFD	V$OPIO	000373
SGNT	V$OPIO	000412

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
SGNTD	VSOPIO	000400
SHRINK	SHRINK	000000
STFCB	VSGFCBS	000123
STGN	STGN	000022
STN	STNCOS	000000
STNT	SORT	001367
SMATN	VSSMATN	000000
SMLTR	MICSIM	000201
SNAP	VSSNAP	000000
SNGI	SNGI	000000
SORT	SORTH	000014
SRT1	SORT	000010
SRTC	MERGE	002473
SSFCB	VSGFCBS	000205
STFN	VSLPS31	000061
STRS	VSSTRS	000000
STUS	VSLPS31	000000
SUSPND	SUSPND	000023
SWRS	VSSWRS	000004
SYMBOI	SYMBLM	000000
SYSGET	BITSET	000146
TANH	TANH	000000
TCSBRO	TCSCEX	000000
TCSCEX	TCSCEX	000474
TCSCRO	TCSCEY	000120
TCSERO	TCSCEX	000114
TCSERR	TCSCEY	000032
TCSEWR	TCSCEX	000041
TCSOCM	CTTCOA	000445
TCSOCT	VTSTCO	000525
TCSTCO		000001
TCKS	VSTCKS	000000
TIME	TIME	000026
TLFC	MERGE	002543
TOPERM	PLOTS	000000
TTYTCM	TTYTCM	000000
TUID	TUID	000000
TYOOA	VSTYOOA	000000
TYFUNC	TYREAD	003001
TYREAD	TYREAD	000137
TYWRIT	TYREAD	001663
VSALTR	VSSERV	001313

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
VSBTC	VSIOC	002044
VSCLOK	VSFUNC	001236
VSCLOS	VSFORTIO	000112
VSCLPS	VSCLPS	000000
VSCLSR	VSFORTIO	000051
VSCRDR	VSLMEMBK	000341
VSDMD	VSFUNC	000444
VSDTSP	VSFUNC	000003
VSDP1	VSFUNC	000011
VSDPCN	VSDPCN	000126
VSDPCT	MATRIX	000000
VSDPCX	PLOT	000336
VSDPCY	PLOT	000417
VSDPER	PLOTS	000063
VSDPF1	PLOTS	000332
VSDPF3	PLOTS	000334
VSDPFA	PLOT	000503
VSDPFC	PLOTS	000331
VSDPFT	PLOT	000015
VSDPGR	CHARM	000174
VSDPIN	PLOTS	000205
VSDPIV	PLOT	000512
VSDPLC	PLOT	000507
VSDPNL	PLOTS	000344
VSDPPE	PLOTS	000133
VSDPR1	PLOTS	000346
VSDPRA	PLOTS	000345
VSDPST	MATRIX	000340
VSDPSY	PLOTS	000353
VSDPVE	PLOT	000661
VSDPXM	PLOT	000501
VSDPXO	PLOT	000502
VSDPYM	PLOT	000477
VSDPYO	PLOT	000500
VSDRTN	VSFUNC	000505
VSDUMP	VSJPDUMP	000000
VSERDR	VSSYTASK	000000
VSERR	VSIOC	001436
VSEXEC	VSSERV	000000
VSFNIS	VSSYTASK	002415
VSFNR	VSIOC	000507
VSFNRM	VSIOC	001034

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
V$FPP	V$FIINC	002000
V$GFCB	V$GFCBS	000123
V$IWD	V$FIINC	000600
V$IM	V$LMEMBK	000320
V$IOC	V$IOC	000000
V$IOST	V$IOC	001762
V$IUTL	V$IOUTIL	000000
V$JCP	V$JCP	000000
V$JPBF	V$GFCBS	000052
V$LFVT	V7SPA	000245
V$LIOP	V	000001
V$MALC	V$SYTASK	003413
V$MAP2	V$FIINC	001305
V$MAPF	V$FIINC	001415
V$MAPT	V$FIINC	001275
V$MDAI	V$SYTASK	003677
V$MLD	V$SYTASK	003132
V$MP2	V$FIINC	001425
V$MP3	V$FIINC	001445
V$MP4	V$FIINC	001606
V$MPIO	V$FIINC	001271
V$OPBF	V$GFCBS	000001
V$OPCM	V$OPCOM	000000
V$OPEN	V$FORTIO	000273
V$OPIO	V$OPIO	000313
V$OPNR	V$FORTIO	000264
V$PAL	V$SYTASK	004132
V$PDAI	V$SYTASK	004255
V$PFDN	V$FIINC	002406
V$PFUP	V$FIINC	002217
V$RER1	V$RERS	005305
V$RERF	V$RERF	000000
V$RERN	V$RERS	000000
V$RERR	V$RERS	005234
V$RERS	V$RERS	000000
V$SAL	V$SYTASK	000521
V$SCPT	V$STWD	000037
V$SMS	V$IOC	002147
V$SPAC	V$CLPS	000371
V$SPAM	V$SPRM	000261
V$SPAR	V$SPRM	000122
V$SPDR	V$SPRM	000017

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
V$SPDT	V$SPRM	000072
V$SPEM	V$SPRM	000426
V$SPIN	CTSPOA	000074
V$SPIR	V$SPLC	001100
V$SPLP	V$SPLC	000654
V$SPMR	V$SPRM	000335
V$SPMC	V$SPRM	000333
V$SPMT	V$SPRM	000334
V$SPPR	V$SPRM	000331
V$SPQR	V$SPRM	000000
V$SPQS	V$SPRM	000336
V$SPQT	V$SPRM	000042
V$SPRM	V$SPRM	000272
V$SPRS	V$SPRM	000332
V$SPSR	V$SPRM	000172
V$SPST	V$SPLC	001133
V$SPTR	V$SPLC	001122
V$SPTT	V$SPLC	001117
V$STBO	V$CLPS	001165
V$STEP	V$CLPS	000323
V$STPI	V$CLPS	000506
V$STWD	V$STWD	000000
V$TBSR	V$SERV	001424
V$TOFM	V$CLPS	000427
V$TYA	V$TYA	000000
V$VTAM	V$FIINC	000520
V$WCS	CTWCS	000074
V$WCSF	CTWCS	000064
V$WCSF	CTWCS	000032
V$WCST	CTWCS	000000
V$WCSO	CTWCS	000025
V$WPIT	V7LPDX	000000
V2$CRI	CC$CRO	000202
V2$CRT	TC$CEX	000364
VFCT	PLOT	000270
VT$PMT	VT$PMT	000000
VT$CLS	VT$CLS	000000
VT$GTM	VT$GTM	000000
VT$TOC	VT$TOC	000000
VT$LTT		000000
VT$MAP	CC$CEX	000340
VT$OCL	VT$OCL	000000

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
VTSOCT	VTSOCT	000000
VTSOCY	VTSOCT	000575
VTSOC7	VTSOCT	000577
VTSOPN	VTSOPN	000000
VTSPTM	VTSPTM	000000
VTSTCO	VTSTCO	000000
VTSTMP	TCSCEX	000451
VTPOP	VTPOP	000000
VTPUSH	VTPUSH	000000
V7CTA	V7CTA	000000
V7CDA	V7CDA	000000
V7CPA	V7CPA	000000
V7CRA	V7CRA	000000
V7DA	V7DA	000000
V7DB	V7DB	000000
V7DC	V7DC	000000
V7DD	V7DD	000000
V7LPA	V7LPA	000000
V7MTA	V7MTA	000000
V7PTA	V7PTA	000000
V7SP0A	V7SPA	000000
V7SP1A	V7SPA	000000
V7SP2A	V7SPA	000000
V7SP3A	V7SPA	000000
V7SP4A	V7SPA	000000
V7SP5A	V7SPA	000000
V7SP6A	V7SPA	000000
V7SP7A	V7SPA	000000
W$X	V$FORTIO	000477
WHERE	PLOT	000061
WRIT	V$LPS31	000154
WRT$	VWRT	000000
WRTS	MERGE	003365
XDAD	XDADD	000021
XDCO	XDCOMP	000000
XDDT	XDDTVH	000130
XDIV	XDIV	000124
XDMU	XDMULTW	000062
XDSU	XDSUB	000022
XTT1	MERGE	004254
XTT2	SORT	000000
XMUL	XMUL	000053

ENTRY NAME CROSS REFERENCE

ENTRY	PROGRAM	LOCATION
XSF1	MERGE	002720
XSF2	MERGE	002731
XSF3	MERGE	002742
/FINI		