```
 OPTIONS NODECK,LIST,XREF,NOREL,OBJ(P)
THE LIST OF OPTIONS USED DURING THIS ASSEMBLY IS--   NODECK,LIST,XREF,NOREL,OBJ
```

```
                                EXTERNAL SYMBOL LIST
SYMBOL   TYPE                                                  VER 15, MOD 00  20/07/20  PAGE   1
#BOVLY   MODULE
```

```
    0000                          1 #BOVLY START 0
                                  2        PRINT ON,NODATA
                                  3 *      @SYS  EXP-N
                                214+       PRINT ON
                                215 *      @FXD  EXP-N
                                620+       PRINT ON
                                621 *      @CAN  EXP-N
                                724+       PRINT ON
                                725 *      @SPF  EXP-N
                               1188+       PRINT ON
                               1189 *      @B@E  EXP-N
                               2089+       PRINT ON
                               2090 *      @ERM  EXP-N
                               2712+       PRINT ON
                               2713 *      $V$E  EXP-N
                               3135+       PRINT ON
    0800                       3145        ORG   #$$BOV
```

```
                               3147 ***********************************************************************
                               3148 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               3149 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               3150 *                                                                     *
                               3151 ***********************************************************************
                               3152 *STATUS                                                               *
                               3153 *   VERSION 1 MODIFICATION 0                                          *
                               3154 *                                                                     *
                               3155 *FUNCTION                                                             *
                               3156 *   BXINPT IS EXECUTED TO TRANSLATE INPUT STATEMENTS AS THEY OCCUR IN *
                               3157 *   A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE  *
                               3158 *   PSEUDOCODE IN VIRTUAL MEMORY.                                     *
                               3159 *                                                                     *
                               3160 *ENTRY POINTS                                                         *
                               3161 *   BXINPI HAS ONLY ONE ENTRY POINT:                                  *
                               3162 *        BXINPT - TRANSLATE INPUT STATEMENT                           *
                               3163 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               3164 *        B     BXINPT                                                 *
                               3165 *                                                                     *
                               3166 *INPUT                                                                *
                               3167 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               3168 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               3169 *     LEADING KEYWORD, INPUT.                                         *
                               3170 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF IF FIRST  *
                               3171 *     CHARACTER IN THE LEADING KEYWORD, INPUT.                        *
                               3172 *                                                                     *
                               3173 *OUTPUT                                                               *
                               3174 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               3175 *     GENERATED BY BXINPT IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               3176 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               3177 *     SEQUENCES.                                                      *
                               3178 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               3179 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               3180 *                                                                     *
                               3181 *EXTERNAL REFERENCES                                                  *
                               3182 *     B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.   *
                               3183 *     B$PUTC - (B$PCAD, B$PNBY, B$PVAD) - ENTRY TO COMPILER VIRTUAL   *
                               3184 *              MEMORY OUTPUT ROUTINE.                                 *
                               3185 *     B$LIST - (B$LTYP) - ENTRY TO BASIC COMPILER LIST ADDR ROUTINE.  *
                               3186 *     B$BTAL - (B$BRVA, B$BRLN) - ENTRY TO BASIC COMPILER BRANCH      *
                               3187 *              TABLE ROUTINE.                                         *
                               3188 *     B$INVT - ENTRY TO THE INPUT VERIFICATION TABLE.                 *
                               3189 *     B$CSXA - STARTING CORE ADDRESS FOR EXCESS CORE.                 *
                               3190 *     B$CSBF - DISK RESIDENT PMC GENERATORS.                          *
                               3191 *     $EXFTR - EXTENSION FACTOR.                                      *
                               3192 *     B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                   *
                               3193 *                                                                     *
                               3194 *EXITS, NORMAL                                                        *
                               3195 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               3196 *                                                                     *
                               3197 *EXITS, ERROR                                                         *
                               3198 *   N/A                                                               *
                               3199 *                                                                     *
                               3200 *TABLES/WORK AREAS                                                    *
                               3201 *   * INPUT VERIFICATION TABLE - EXTENSAL TO BXINPT, THIS 97-BYTE     *
                               3202 *     TABLE IS USED TO LOG DATA ELEMENT TYPES ASSOCIATED WITH THE     *
                               3203 *     DATA LIST FOR AN INPUT STATEMENT.  EACH SINGLE-BYTE TABLE ENTRY *
                               3204 *     CONTAINS A DATA TYPE CODE AND A COUNT OF THE NUMBER OF DATA     *
                               3205 *     ELEMENTS OF THE SAME TYPE WHICH OCCUR CONTIGUOUSLY AT SOME      *
                               3206 *     BXINT IN THE LIST; A NEW ENTRY IS STARTED EACH TIME A DIFFERENT *
                               3207 *     TYPE IS ENCOUNTERED.  THE TABLE IS ALWAYS SET TO BINARY ZEROS   *
                               3208 *     BEFORE A LIST IS PROCESSED, AND A ZERO ENTRY MARKS THE END OF   *
                               3209 *     THE CURRENT SERIES OF LOGGED ENTRIES.                           *
                               3210 *                                                                     *
                               3211 *ATTRIBUTES                                                           *
                               3212 *   BXINPT IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               3213 *                                                                     *
                               3214 *CHARACTER CODE DEPENDENCY                                            *
                               3215 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               3216 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               3217 *                                                                     *
                               3218 *NOTES                                                                *
                               3219 *   ERROR PROCEDURES                                                  *
                               3220 *      N/A                                                            *
                               3221 *                                                                     *
                               3222 *   REGISTER USAGE                                                    *
                               3223 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               3224 *                                                                     *
                               3225 *   SAVED/RESTORED AREAS                                              *
                               3226 *      N/A                                                            *
                               3227 *                                                                     *
                               3228 *   MODIFICATION CONSIDERATIONS                                       *
                               3229 *      BXINPT CROSSES A SECTOR BOUNDARY AND RESIDES ON TWO SECTORS,   *
                               3230 *      CO-RESIDENT ON THE SECOND ONE WITH BNADIM.  ANY MODIFICATIONS  *
                               3231 *      MUST MAINTAIN LINKAGE BETWEEN THE TWO SECTORS, CONSIDER ANY    *
                               3232 *      CHANGE IN THE ENTRY ADDRESS OF BNADIM, AND REALIZE THE         *
                               3233 *      LIMITATION OF THE SECTOR BOUNDARY UPON SIZE.                   *
                               3234 *                                                                     *
                               3235 *   REQUIRED MODULES                                                  *
                               3236 *        @SYSEQ - COMMON SYSTEM EQUATES.                              *
                               3237 *        @FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUE EQUATES. *
                               3238 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS ECUATES.     *
                               3239 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               3240 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               3241 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               3242 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               3243 *        $B$EQU - COMPILER FIXED EQUATES.                             *
                               3244 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               3245 *                                                                     *
                               3246 *   OTHER                                                             *
                               3247 *      BXINPT IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               3248 ***********************************************************************

    0800                       3250       ORG   *,256,0                    BEGIN AT CORE PAGE BOUNDARY
                          0800 3251       USING *,@BR                      DEFINE BASE ADDR FOR CORE PAGE
                               3252 *
                               3253 * ENTER BXINPT - 'INPUT' STATEMENT PROCESSOR
                               3254 *
                          0800 3255 BXINPT EQU   *                        BXINPT ENTRY POINT
    0800 74 01 EF              3256       ST    BXICA2(,@BR),@BR         SAVE BXINPT BASE ADDRESS
                               3257 *
                               3258 * SET POINTER TO SKIP TO 'T' IN KEYWORD 'INPUT'
                               3259 *
    0803 3C 04 0873            3260 BXI010 MVI   B$NUMC,B@LINP-1           SET GET RTN TO SKIP TO 'T'
    0807 C0 87 0867            3261        B     B$GETC                    LINK TO ADVANCE POINTER
                               3262 *
                               3263 * GENERATE AN 'STA' INSTRUCTION IMAGE
                               3264 *
    080B D2 02 E5              3265 BXI020 LA    BXISTC(,@BR),@XR          LOAD CADDR OF 'STA' OPCODE
    080E 34 02 0A40            3266        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'STA'
    0812 3C 02 0A41            3267        MVI   B$PNBY,B@LSTA-1           SET LNG PARM OF PUT FOR 'STA'
    0816 C0 87 093A            3268        B     B$PUTC                    LINK TO GENERATE 'BRA' IMAGE
                               3269 *
                               3270 * ESTABLISH 'STA' OPERAND FOR ADDRESS RESOLUTION CONDITIONS
                               3271 *
    081A 0C 01 19EF 0A43       3272 BXI030 MVC   B$BRVA,B$PVAD(@VADDR)     SET BR TABLE VADDR PARM
    0820 1F 01 19EF FA         3273        SLC   B$BRVA,BXIBN1(@VADDR,@BR)  ADJ VADDR FOR 'STA' RESOLUTION
                               3274 *
                               3275 * GENERATE A 'BRA' INSTRUCTION IMAGE TO VERIFICATION SEQUENCE
                               3276 *
    0825 D2 02 E8              3277 BXI040 LA    BXIBRC(,@BR),@XR          LOAD CADDR OF 'BRA' INSTR
    0828 34 02 0A40            3278        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'STA'
    082C 3C 02 0A41            3279        MVI   B$PNBY,B@LBRA-1           SET LNG PARM OF PUT FOR 'STA'
    0830 C0 87 093A            3280        B     B$PUTC                    LINK TO GENERATE 'BRA' IMAGE
                               3281 *
                               3282 * SAVE NEXT VIRTUAL MEMORY ADDRESS FOR LINE NUMBER RESOLUTION
                               3283 *
    0834 0C 01 19F1 0A43       3284 BXI050 MVC   B$BRLN,B$PVAD(@VADDR)     SET BR TABLE LENGTH NO PARM
                               3285 *
                               3286 * BRANCH TO BRANCH TABLE ROUTINE TO SET UP RESOLUTION COND FOR 'STA'
                               3287 *
    083A C0 87 1996            3288 BXI060 B     B$BTAB                    LINK TO RESOLVE 'STA' INSTR
                               3289 *
                               3290 *ESTABLISH 'BRA' OPERAND FOR ADDRESS RESOLUTION CONDITIONS
                               3291 *
    083E 0C 01 19EF 0A43       3292 BXI070 MVC   B$BRVA,B$PVAD(@VADDR)     SET BR TABLE VADDR PARM
    0844 1F 01 19EF FA         3293        SLC   B$BRVA,BXIBN1(@VADDR,@BR)  ADJ VADDR FOR 'BRA' RESOLUTION
                               3294 *
                               3295 * PROPAGATE ZEROES TO INITIALIZE THE INPUT VERIFICATION TABLE
                               3296 *
    0849 0F 56 1B8E 1B8E       3297 BXI080 SLC   BXITB1,BXITB1(B@NIVT)     INITIALIZE TABLE TO ZEROES
    084F 7C 1D 6A              3298        MVI   BXI145+@D1(,@BR),BXI185-BXI145-@INST3  SET 1ST PASS BRANCH
                               3299 *
                               3300 * SET THE TABLE DISPLACEMENT(POINTER) TO REFERENCE FIRST ENTRY
                               3301 *
    0852 7C 00 63              3302 BXI090 MVI   BXI130+@D1(,@BR),@ZERO    SET POINTER TO FIRST ENTRY
                               3303 *
                               3304 * LINK TO GET THE NEXT LIST ELEMENT
                               3305 *
    0855 C0 87 0867            3306 BXI100 B     B$GETC                    LINK TO GET NEXT CHAR
                               3307 *
                               3308 * LINK TO LIST ROUTINE TO GENERATE PMC FOR LIST VARIABLE
                               3309 *
    0859 C0 87 1853            3310 BXI110 B     B$LIST                    LINK TO GENERATE PMC
                               3311 *
                               3312 * SET INPUT VERIFICATION TABLE POINTER
                               3313 *
    085D C2 02 1B38            3314 BXI120 LA    B$INVT,@XR                SET TABLE POINTER
    0861 E2 02 00              3315 BXI130 LA    *-*(,@XR),@XR             * BY DISPLACEMENT
                               3316 *
                               3317 * TEST THIS ELEMENT FOR BEING A CHARACTER VARIABLE
                               3318 *
    0864 38 01 18F2            3319 BXI140 TBN   B$LTYP,B$CRMK             IF THIS IS A CHAR VARIABLE
    0868 F2 10 00              3320 BXI145 JT    *-*                       * GO PROCESS TABLE ENTRY
                               3321 *
                               3322 * IF THIS IS ARITH VARIABLE TEST TABLE ENTRY FOR ARITH SETTING
                               3323 *
    086B B9 80 00              3324 BXI150 TBF   BXIVTE(,@XR),BXICMK       IF ENTRY IS SET FOR ARITH VAR
    086E F2 10 1A              3325        JT    BXI190                    * GO ADD TO ENTRY ELEMENT COUNT
                               3326 *
                               3327 * IF ENTRY IS NOT ARITP ADVANCE TO NEXT ENTRY AND SET IT FOR AN ARITH
                               3328 * ELEMENT WITH A COUNT OF ZERO
                               3329 *
    0871 5E 00 63 FA           3330 BXI160 ALC   BXI130+@D1(,@BR),BXIBN1(1,@BR)  INCREMENT POINTER
    0875 E2 02 01              3331        LA    @B1(,@XR),@XR             ADVANCE TO NEXT ENTRY
    0878 F2 87 10              3332        J     BXI190                    GO ADD TO ENTRY COUNT
                               3333 *
                               3334 * TEST TABLE ENTRY FOR BEING SET FOR CHARACTER ELEMENTS
                               3335 *
    087B B8 80 00              3336 BXI170 TBN   BXIVTE(,@XR),BXICMK       IF ENTRY IS SET FOR CHAR
    087E F2 10 0A              3337        JT    BXI190                    * GO ADD TO ENTRY COUNT
                               3338 *
                               3339 * IF ENTRY IS NOT CHAR ADVANCE TO NEXT ENTRY AND SET IT FOR A CHARACTER
                               3340 * ELEMENT WITH A COUNT OF ZERO
                               3341 *
    0881 5E 00 63 FA           3342 BXI180 ALC   BXI130+@D1(,@BR),BXIBN1(1,@BR)  INCREMENT POINTER
    0885 E2 02 01              3343        LA    @B1(,@XR),@XR             ADVANCE TO NEXT ENTRY
    0888 BA 80 00              3344 BXI185 SBN   BXIVTE(,@XR),BXICMK       SET ENTRY FOR CHAR ELEMENT - 0
                               3345 *
                               3346 * BXINCREMENT INPUT VERIFICATION TABLE ENTRY ELEMENT COUNT
                               3347 *
    088B 9E 00 00 FA           3348 BXI190 ALC   BXIVTE(,@XR),BXIBN1(1,@BR)  INCREMENT ELEMENT COUNT
    088F 7C 10 6A              3349        MVI   BXI145+@D1(,@BR),BXI170-BXI145-@INST3  SET NORMAL CHAR BR
                               3350 *
                               3351 * GENERATE 'GET' INSTRUCTION IN VIRTUAL MEMORY
                               3352 *
    0892 D2 02 EB              3353 BXI210 LA    BXIGTC(,@BR),@XR          LOAD CADDR OF 'GET' OPCOOE
    0895 34 02 0A40            3354        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'GET'
    0899 3C 02 0A41            3355        MVI   B$PNBY,B@LGET-1           SET LNG PARM OF PUT FOR 'GET'
    089D C0 87 093A            3356        B     B$PUTC                    LINK TO GENERATE 'GET' INSTR
                               3357 *
                               3358 * TEST FOR THE LIST DELIMITER BEING A STATEMENT TERMINATOR
                               3359 *
    08A1 35 02 0878            3360 BXI220 L     B$GPTR,@XR                RESTORE TEXT POINTER
    08A5 BD 1E 00              3361        CLI   B@CHAR(,@XR),B@EOST       IF THIS IS NOT THE END OF STMT
    08A8 D0 01 55              3362        BNE   BXI100(,@BR)              * GO PROCESS THE NEXT ELEMENT
                               3363 *
                               3364 * GENERATE A 'BRA' IMAGE IN VIRT MEMORY
                               3365 *
    08AB D2 02 E8              3366 BXI230 LA    BXIBRC(,@BR),@XR          LOAD CADDR OF 'BRA' INSTR
    08AE 34 02 0A40            3367        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'BRA'
    08B2 3C 02 0A41            3368        MVI   B$PNBY,B@LBRA-1           SET LNG PARM OF PUT FOR 'BRA'
    08B6 C0 87 093A            3369        B     B$PUTC                    LINK TO GENERATE PMC
```

```
                               3371 ***********************************************************************
                               3372 * INPUT 2ND SEGMENT CALLING SEQUENCE ROUTINE
                               3373 ***********************************************************************
                               3374 *
                               3375 * TEST WHETHER CURRENT SEGMENT WAS DISK OR CORE RESIDENT
                               3376 *
    08BA 5D 01 EF F8           3377 BXI240 CLC   BXICA2(,@BR),BXIPBA(@CADDR,@BR)  IF CURR SEG CAME FR DISK
    08BE F2 81 10              3378        JE    BXI260                           * GO LOAD & EXEC 2ND SEG
                               3379 *
                               3380 * CURRENT SEGMENT WAS CORE RESIDENT - TEST WHETHER 2ND SEGMENT HAS ALSO
                               3381 * BEEN LOADED INTO CORE
                               3382 *
    08C1 5C 01 F2 F4           3383 BXI250 MVC   BXIFCP(,@BR),BXIFPE(@CADDR,@BR)  SET FINAL CORE PAGE
    08C5 4E 00 F1 043B         3384        ALC   BXIFCP-1(,@BR),$EXFTR(1)  CALC MAX PROCESSOR CORE PAGE
    08CA 5D 01 EF F2           3385        CLC   BXICA2(,@BR),BXIFCP(@CADDR,@BR)  IF 2ND SEGMENT IN CORE
    08CE F2 82 0B              3386        JL    BXI280                           * GO SET TO EXEC 2ND SEG
                               3387 *
                               3388 * 2ND SEGMENT IS DISK RESIDENT - ESTABLISH DISTRIBUTOR PARAMETERS FOR
                               3389 * CORELOADING AND EXECUTING THE 2ND SEGMENT
                               3390 *
    08D1 5C 01 EF F8           3391 BXI260 MVC   BXICA2(,@BR),BXIPBA(@CADDR,@BR)  SET UP DISY:LOAD CADDR
                               3392 *
                               3393 * EXIT TO DISTRIBUTOR FOR SECOND SEGMENT CORELOAD AND EXECUTION
                               3394 *
    08D5 D2 02 EE              3395 BXI270 LA    BXIAD2(,@BR),@XR          LOAD DISTRIBUTOR PARM CADDR
    08D8 C0 87 073A            3396        B     B$DST2                    GO LOAD AND EXECUTE 2ND SEGMENT
                               3397 *
                               3398 * 2ND SEGMENT IS CORE RESIDENT BRANCH TO NEXT CONSECUTIVE CORE PAGE
                               3399 * AND CONTINUE INPUT EXECUTION
                               3400 *
    08DC 75 01 EF              3401 BXI280 L     BXICA2(,@BR),@BR          LOAD THE BASE ADDRESS FOR
    08DF 76 01 F6              3402        A     BXIBLS(,@BR),@BR          * 2ND INPUT SEGMENT
    08E2 D0 87 00              3403        B     BXISG2(,@BR)              GO EXECUTE THE 2ND SEGMENT
```

```
                               3405 ***********************************************************************
                               3406 * 'INPUT' STATEMENT SEGMENT-1 STORAGE AMP PARAMEIER AREAS
                               3407 ***********************************************************************
                               3408 *
    08E5 34                  08E5 3409 BXISTC DC    AL(B@LCOP)(B@CSTA)        'STA' INSTR OPCODE
    08E6 0000                08E7 3410 BXISTO DC    XL(B@LCVA)'00'            'STA' INSTR OPERAND IMAGE
                               3411 *
    08E8 46                  08E8 3412 BXIBRC DC    AL(B@LCOP)(B@CBRA)        'BRA' INSTR OPCODE
    08E9 0000                08EA 3413 BXIBRO DC    XL(B@LCVA)'00'            'BRA' INSTR OPERAND IMAGE
                               3414 *
    08EB 52                  08EB 3415 BXIGTC DC    AL(B@LCOP)(B@CGET)        'GET' INSTR OPCODE
    08EC 2B6E                08ED 3416 BXIGTO DC    AL(B@LCVA)(V$XSIN)        'GET' INSTR OPERAND

                               3418 ***********************************************************************
                               3419 * 'INPUT' STATEMENT SEGMENT-1 WORK AREAS
                               3420 ***********************************************************************
                               3421 *
                          08EE 3422 BXIAD2 EQU   *                         DISTR PARMS FOR SEG-2 EXEC
    08EE                     08EF 3423 BXICA2 DS    CL(@CADDR)                INPUT SEGMENT CORE ADDRESS
    08F0 04                  08F0 3424        DC    AL1(B@DINP+BXIPSI)        BXINPT SEG-2 PHYS SECTOR ADDR
    08F1                     08F2 3425 BXIFCP DS    CL(@CADDR)                FINAL AVAILABLE CORE PAGE ADDR
    08F3 1F00                08F4 3426 BXIFPE DC    AL(@CADDR)(B$CSXA-B@BLSZ)  FINAL PAGE BEFORE EXTENSION

                               3428 ***********************************************************************
                               3429 * 'INPUT' STATEMENT SEGMENT-1 CONSTANTS
                               3430 ***********************************************************************
                               3431 *
    08F5 0100                08F6 3432 BXIBLS DC    AL(@CADDR)(B@BLSZ)        LENGTH OF CORE BLOCK OR PAGE
    08F7 0600                08F8 3433 BXIPBA DC    AL(@CADDR)(B$CSBF)        PROCESSOR DISK BUFFER CADDR
    08F9 0001                08FA 3434 BXIBN1 DC    IL(@VADDR)'1'             BINARY 1

                               3436 ***********************************************************************
                               3437 * 'INPUT' STATEMENT SEGMENT-1 EQUATES
                               3438 ***********************************************************************
                               3439 *
                          0004 3440 BXIPSI EQU   X'04'                     PHYSICAL SECTOR INCREMENT
                          0000 3441 BXISG2 EQU   0                         DISP FOR BXINPT SEG-2 ENTRY PT
                          0000 3442 BXIVTE EQU   0                         DISP FOR INPT VERIF'N TBL ENT
                          0001 3443 BXILTE EQU   1                         LENGTH OF TABLE ENTRY
                               3444 *
                          1B8E 3445 BXITB1 EQU   B$INVT+B@NIVT-1           LENGTH TO LAST TABLE ENTRY
                          0080 3446 BXICMK EQU   X'80'                     MASK FOR CHAR ELEMENTS

                               3448 ***********************************************************************
                               3449 * 'INPUT' STATEMENT SECOND SEGMENT
                               3450 ***********************************************************************
                               3451 *
                               3452 * ESTABLISH INPUT SEGMENT-2 ADDRESSABILITY
                               3453 *
    0900                       3454        ORG   BXINPT+B@BLSZ             BEGIN SEGMENT-2 AT PAGE BOUND
                          0900 3455        USING *,@BR                     DEFINE SEGMENT-2 BASE ADDRESS
                               3456 *
                               3457 * ESTABLISH NEXT VIRT MEM LOCATION AS RESOLUTION ADDRESS
                               3458 *
    0900 0C 01 19F1 0A43       3459 BXI290 MVC   B$BRLN,B$PVAD(@VADDR)     SET BR TABLE LENGTH NO PARM
    0906 C0 87 1996            3460        B     B$BTAB                    LINK TO RESOLVE 1ST 'BRA' INSTR
```

```
                               3461 *
                               3462 * ESTABLISH 2ND BRA OPERAND FOR BR TABLE ADDRESS RESOLUTION
                               3463 *
    090A 0C 01 19EF 0A43       3464 BXI300 MVC   B$BRVA,B$PVAD(@VADDR)     SET VADDR PARM FOR BR TBL RTN
    0910 1F 01 19EF 72         3465        SLC   B$BRVA,BXIONE(@VADDR,@BR) ADJUST TO BRA OPND
                               3466 *
                               3467 * SET THE INPUT VERIFICATION TABLE ENTRY COUNT TO ZERO
                               3468 *
    0915 7C 00 6D              3469 BXI310 MVI   BXIINO(,@BR),@ZERO        SET COUNT TO ZERO
                               3470 *
                               3471 * SET THE INPUT VERIFICATION TABLE POINTER TO THE FIRST TABLE ENTR -
                               3472 *
    0918 7C 00 21              3473 BXI320 MVI   BXI340+@D1(,@BR),@ZERO    SET POINTER FOR 1ST ENTRY
                               3474 *
                               3475 * CHECK TABLE FOR HAVING BEEN EXHAUSTED
                               3476 *
    091B C2 02 1B38            3477 BXI330 LA    B$INVT,@XR                LOAD ADDR OF I.V. TABLE
    091F E2 02 00              3478 BXI340 LA    *-*(,@XR),@XR             SET I.V. TABLE POINTER DISP
                               3479 *
    0922 BD 00 00              3480 BXI350 CLI   BXIVTE(,@XR),@ZERO        END OF THE TABLE REACHED ?
    0925 F2 81 1E              3481        JE    BXI410                    * GO GENERATE 'INI' INSTR
                               3482 *
                               3483 * MOVE THE TABLE ENTRY TO THE OPERAND OF AN 'STX' INSTR
                               3484 *
    0928 6C 00 6F 00           3485 BXI360 MVC   BXISXO(,@BR),BXIVTE(1,@XR) SET OPND OF 'STX' INSTR
                               3486 *
                               3487 * GENERATE THE 'STX' INSTR IN VIRTUAL MEMORY
                               3488 *
    092C D2 02 6E              3489 BXI370 LA    BXISXC(,@BR),@XR          LOAD CADDR OF 'STX' INSTR
    092F 34 02 0A40            3490        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'STX'
    0933 3C 01 0A41            3491        MVI   B$PNBY,B@LSTX-1           SET LNG PARM OF PUT FOR 'STX'
    0937 C0 87 093A            3492        B     B$PUTC                    LINK TO GENERATE 'STX' INSTR
                               3493 *
                               3494 * INCREMENT THE TABLE ENTRY COUNTER BY ONE
                               3495 *
    093B 5E 00 6D 72           3496 BXI380 ALC   BXIINO(,@BR),BXIONE(B@LCNN,@BR)  INCREMENT ENTRY COUNTER
                               3497 *
                               3498 * INCREMENT THE TABLE POINTER TO REFERENCE THE NEXT ENTRY
                               3499 *
    093F 5E 00 21 72           3500 BXI390 ALC   BXI340+@D1(,@BR),BXIONE(1,@BR)  INCREMENT POINTER
                               3501 *
                               3502 * BRANCH TO PROCESS THE NEXT TABLE ENTRY
                               3503 *
    0943 D0 87 1B              3504 BXI400 B     BXI330(,@BR)              GO PROCESS NEXT ENTRY
                               3505 *
                               3506 * GENERATE THE 'INI' INSTRUCTION IN VIRTUAL MEMORY
                               3507 *
    0946 D2 02 6C              3508 BXI410 LA    BXIINC(,@BR),@XR          LOAD CADDR OF 'INI' INSTR
    0949 34 02 0A40            3509        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'INI'
    094D 3C 01 0A41            3510        MVI   B$PNBY,B@LINI-1           SET LNG PARM OF PUT FOR 'INI'
    0951 C0 87 093A            3511        B     B$PUTC                    LINK TO GENERATE 'INI' INSTR
                               3512 *
                               3513 * GENERATE A 'BRS' INSTR IN VIRTUAL MEMORY
                               3514 *
    0955 D2 02 70              3515 BXI420 LA    BXIBSC(,@BR),@XR          LOAD CADDR OF 'BRS' INSTR
    0958 34 02 0A40            3516        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'BRS'
    095C 3C 00 0A41            3517        MVI   B$PNBY,B@LBRS-1           SET LNG PARM OF PUT FOR 'BRS'
    0960 C0 87 093A            3518        B     B$PUTC                    LINK TO GENERATE 'BRS' INSTR
                               3519 *
                               3520 * SET SWITCH FOR BRANCH TABLE RESOLUTION OF 2ND 'BRA' INSTR
                               3521 *
    0964 3A 07 071D            3522 BXI430 SBN   B$NXSW,B$NXMK             SET 'NEXT' SWITCH
                               3523 *
                               3524 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               3525 *
    0968 C0 87 0700            3526 BXI440 B     B$DIST                    RETURN TO DISTRIBUTOR

                               3528 ***********************************************************************
                               3529 * 'INPUT' STATEMENT SEGMENT-2 STORAGE AND PARAMETER AREAS
                               3530 ***********************************************************************
                               3531 *
    096C 56               096C 3532 BXIINC DC    AL(B@LCOP)(B@CINI)        'INI' INSTR OPCODE
    096D                  096D 3533 BXIINO DS    CL(B@LCNN)                'INI' INSTR OPM - COUNTER
                               3534 *
    096E 3C               096E 3535 BXISXC DC    AL(B@LCOP)(B@CSTX)        'STX' INSTR OPCODE
    096F                  096F 3536 BXISXO DS    CL(B@LCXX)                'STX' INSTR OPMD
                               3537 *
    0970 4C               0970 3538 BXIBSC DC    AL(B@LCOP)(B@CBRS)        'BRS' 1NSTR OPCODE

                               3540 ***********************************************************************
                               3541 * 'INPUT' STATEMENT SEGMENT-2 CONSTANTS
                               3542 ***********************************************************************
                               3543 *
    0971 0001             0972 3544 BXIONE DC    IL(@VADDR)'1'             BINARY 1
                               3545 *
                               3546 ***********************************************************************
                               3547 *
                               3548 * END OF 'INPUT' STATEMENT CODING
                               3549 *
                               3551 ***********************************************************************
                               3552 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               3553 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               3554 *                                                                     *
                               3555 ***********************************************************************
                               3556 *STATUS                                                               *
                               3557 *   VERSION 1 MODIFICATION 0                                          *
                               3558 *                                                                     *
                               3559 *FUNCTION                                                             *
                               3560 *   BNADIM IS PERFORMED TO ESTABLISH EXPLICIT DIMENSIONS FOR ARITHME- *
                               3561 *   TIC AND CHARACTER ARRAYS APPEARING IN DIM STATEMEN'S AS THEY      *
                               3562 *   OCCUR IN A BASIC PROGRAM.  THESE ARRAYS CANNOT HAVE APPEARED IN A *
                               3563 *   PREVIOUS STATEMENT IN THE PROGRAM.  DIMENSIONS ARE USED TO DEFINE *
                               3564 *   THE SHAPE OF EACH ARRAY AND TO ESTABLISH THE MAXIMUM NUMBER OF    *
                               3565 *   ELEMENTS TO BE CONTAINED IN EACH ARRAY DURING THE COURSE OF THE   *
                               3566 *   PROGRAM.                                                          *
                               3567 *                                                                     *
                               3568 *INPUT                                                                *
                               3569 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               3570 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               3571 *     LEADING KEYWORD, DIM.                                           *
                               3572 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                               3573 *     CHARACTER IN THE LEADING KEYWORD, DIM.                          *
                               3574 *   * ARRAY ATTRIBUTE FIELDS - THE CORE-RESIDENT DOPE VECTOR FOR EACH *
                               3575 *     ARRAY REFERENCE CONTAINS ONSET (0) INDICATOR BITS IN THE FIRST  *
                               3576 *     DIMENSION FIELD BYTE UNLESS THE ARRAY HAS BEEN ENCOUNTERED IN A *
                               3577 *     PRIOR STATEMENT IN THE PROGRAM.                                 *
                               3578 *                                                                     *
                               3579 *OUTPUT                                                               *
                               3580 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               3581 *     GENERATED BY BNADIM IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               3582 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               3583 *     SEQUENCES.                                                      *
                               3584 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               3585 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               3586 *   * ARRAY ATTRIBUTE FIELDS - THESE ARE UPDATED WITH DEFINING        *
                               3587 *     INDICATORS AND VALUES FOR THE FIRST AND SECOND DIMENSION        *
                               3588 *     SPECIFICATIONS IN ARITHMETIC ARRAY DOPE VECTORS AND FOR THE     *
                               3589 *     SINGLE DIMENSION SPECIFICATION IN CHARACTER DOPE VECTORS        *
                               3590 *     ASSOCIATED WITH EACH ARRAY ENCOUNTERED IN THE DIM STATEMENT.    *
                               3591 *                                                                     *
                               3592 *EXTERNAL REFERENCES                                                  *
                               3593 *     B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.   *
                               3594 *     B$PUTC - (BSPFNC, B$PERC) - ENTRY TO COMPILER VIRTUAL MEMORY    *
                               3595 *              OUTPUT ROUTINE.                                        *
                               3596 *     B$SYMB - (B$BCKT, B$FACA, B$CRSW) - ENTRY TO BASIC SYMBOL       *
                               3597 *              TRANSLATION ROUTINE.                                   *
                               3598 *     BSZDBN - (B$BINO) - ENTRY TO BASIC COMPILER ZONED DECIMAL TO    *
                               3599 *              BINARY CONVERSION ROUTINE.                             *
                               3600 *     B$DIST - ENTRY TO BASIC COMPLIER DISTRIBUTOR.                   *
                               3601 *                                                                     *
                               3602 *EXITS, NORMAL                                                        *
                               3603 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               3604 *                                                                     *
                               3605 *EXITS, ERROR                                                         *
                               3606 *   N/A                                                               *
                               3607 *                                                                     *
                               3608 *TABLES/WORK AREAS                                                    *
                               3609 *   * ARRAY ATTRIBUTE FIELDS - EXTERNAL TO BNADIM, THESE DOPE VECTOR  *
                               3610 *     IMAGE SEGMENTS REMAIN CORE-RESIDENT DURING COMPILATION AS PART  *
                               3611 *     OF THE ARRAY SYMBOL TABLES, BEING FLAGGED OR FILLED WITH        *
                               3612 *     DIMENSION INFORMATION AS REFERENCED IN INC PROGRAM              *
                               3613 *                                                                     *
                               3614 *ATTRIBUTES                                                           *
                               3615 *   BNADIM IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               3616 *                                                                     *
                               3617 *CHARACTER CODE DEPENDENCY                                            *
                               3618 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               3619 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               3620 *NOTES                                                                *
                               3621 *   ERROR PROCEDURES                                                  *
                               3622 *      IF ARRAY DEFINITION IS ATTEMPTED FOR AN ARRAY WHICH HAS BEEN   *
                               3623 *      DEFINED PREVIOUSLY, THE ERROR COWITION CODE FOR 'DIM ARRAY     *
                               3624 *      NAME PREVIOUSLY DEFINED' IS LOGGED IN VIRTUAL MEMORY USING     *
                               3625 *      OUTPUT ROUTINE BBPUTC.  BNADIM EXECUTION IS OTHERWISE          *
                               3626 *      UNAFFECTED.                                                    *
                               3627 *                                                                     *
                               3628 *   REGISTER USAGE                                                    *
                               3629 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               3630 *                                                                     *
                               3631 *   SAVED/RESTORED AREAS                                              *
                               3632 *      N/A                                                            *
                               3633 *                                                                     *
                               3634 *   MODIFICATION CONSIDERATIONS                                       *
                               3635 *      BNADIM RESIDES ON A SECTOR WITH BXINPT.  ANY MODIFICATIONS     *
                               3636 *      SHOULD CONSIDER THIS CO-RESIDENCY AND THE SIZE LIMIT OF ONE    *
                               3637 *      SECTOR.                                                        *
                               3638 *                                                                     *
                               3639 *   REQUIRED MODULES                                                  *
                               3640 *        @SYSEQ - COMMON SYSTEM EQUATES                               *
                               3641 *        @FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUE EQUATES. *
                               3642 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS               *
                               3643 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES                    *
                               3644 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES                         *
                               3645 *        @ERMEQ - ERROR MESSAGE EQUATES                               *
                               3646 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES                       *
                               3647 *        $B$EQU - COMPILER FIXED EQUATES                              *
                               3648 *        $B@EQU - COMPILER SYSTEM EQUATES                             *
                               3649 *                                                                     *
                               3650 *   OTHER                                                             *
                               3651 *      BNADIM IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               3652 ***********************************************************************

                               3654 *
                               3655 * ENTER BNADIM - DIMENSION STATEMENT ROUTINE
                               3656 *
                          0973 3657 BNADIN EQU   *                         BNADIM ENTRY POINT
                               3658 *
                               3659 * SET INPUT PARAMETER TO SKIP OVER 'DI' IN KEYWORD
                               3660 *
    0973 3C 02 0873            3661 BNA010 MVI   B$NUMC,B@LDIM-1           SET INPUT PARAMETER
    0977 C0 87 0867            3662        B     B$GETC                    LINK TO SKIP TO 'M' IN KEYWORD
```

```
                               3663 *
                               3664 *  ACCESS NEXT CHARACTER FOR FIRST CHARACTER OF ARRAY TO BE DEFINED
                               3665 *
    097B C0 87 0867            3666 BNA020 B     B$GETC                    LINK TO GET NEXT CHARACTER
                               3667 *
                               3668 * DETERMINE THE CORE ADDRESS OF THE DOPE VECTOR
                               3669 *
    097F C0 87 0DBC            3670 BNA030 B     B$SYMB                    LINK TO GET DOPE VECTOR VADDR
    0983 35 02 0E53            3671        L     B$FACA,@XR                LOAD 'DOPE VECTOR' CADDR
    0987 74 02 AE              3672        ST    BNA090+@OP1(,@BR),@XR     SAVE CADDR IN LOAD ADDR OPERAND
                               3673 *
                               3674 * TEST DOPE VECTOR FOR PREVIOUS ARRAY DEFINITION
                               3675 *
    098A B8 80 00              3676 BNA040 TBN   B@AFLG(,@XR),B@DAMK       CHECK FOR BIT ON INDICATING
                               3677 *                                      * DEFINED ARRAY
    098D F2 90 0C              3678        JF    BNA060                    IF ARRAY DEFINED FALL THROUGH
                               3679 *
                               3680 * GENERATE ERROR MESSAGE IF ARRAY HAS BEEN PREVIOUSLY DEFINED
                               3681 *
    0990 3C 33 094E            3682 BRA050 MVI   B$PFNC,B$PFAE             SET PUT RTN FOR ERROR OUTPUT
    0994 3C A6 0A39            3683        MVI   B$PERC,@@E600             SET ERROR CODE
    0998 C0 87 093A            3684        B     B$PUTC                    LINK TO OUTPUT CHARACTER STRING
                               3685 *
                               3686 * GET NEXT CHARACTER FOR FIRST DIMENSION
                               3687 *
    099C C0 87 0867            3688 BNA060 B     B$GETC                    LINK TO GET NEXT CHARACTER
                               3689 *
                               3690 * CONVERT 1ST DIMENSION OF ARRAY FROM DECIMAL TO BINARY
                               3691 *
    09A0 C0 87 19F2            3692 BNA070 B     B$ZDBN                    LINK TO CONVERT DIMENSION
                               3693 *
                               3694 * TEST SYMBOL TABLE PARAMETER FOR INDICATION OF CHARACTER ARRAY
                               3695 *
    09A4 38 01 0E42            3696 BNA080 TBN   B$CRSW,B$CRMK             IF ARRAY NOT CHARACTER
    09A8 F2 90 0F              3697        JF    BNA120                    * JUMP TO PROCESS ARITH ARRAY
    09AB C2 02 0000            3698 BNA090 LA    *-*,@XR                   LOAD DOPE VECTOR CADDR
                               3699 *
                               3700 * ESTABLISH BINARY DIMENSION OF THE CHARACTER DOPE VECTOR
                               3701 *
    09AF 8C 01 01 1A6A         3702 BNA100 MVC   B@CDMN(,@XR),B$BINO(B@LDMN)  SET DIMENSION FOR CHAR ARRAY
                               3703 *
                               3704 * SET VECTOR BIT ON TO INDICATE ARRAY DEFINITION
                               3705 *
    09B4 BA 80 00              3706 BNA110 SBN   B@AFLG(,@XR),B@DAMK       DEFINE VECTOR ARRAY
    09B7 D0 87 E8              3707        B     BNA190(,@BR)              GO TEST DELIMITER
                               3708 *
                               3709 * PROCESS FOR A ONE-DIMENSION ARITHMETIC ARRAY
                               3710 *
    09BA 35 02 0878            3711 BNA120 L     B$GPTR,@XR                RESTORE TEXT POINTER
    09BE BD 5D 00              3712        CLI   B@CHAR(,@XR),B@RPAR       IF DELIMITER IS NOT ')'
    09C1 D0 01 CD              3713        BNE   BNA130(,@BR)              * GO PROCESS 2ND DELIMITER
    09C4 75 02 AE              3714        L     BNA090+@OP1(,@BR),@XR     LOAD DOPE VECTOR CORE ADDRESS
                               3715 *
                               3716 * THE FIRST DIMENSION OF A SINGLE DIMENSION ARRAY IS 0 TO INDICATE A
                               3717 * ONE DIMENSION ARRAY
                               3718 *
    09C7 BA 80 00              3719        SBN   B@AFLG(,@XR),B@D1MK       SET VECTOR BIT ON
    09CA D0 87 E3              3720        B     BNA180(,@BR)              GO SET 2ND DIMENSION OF ARRAY
                               3721 *
                               3722 * PROCESS FO4 A TWO DIMENSION ARRAY
                               3723 *
    09CD 75 02 AE              3724 BNA130 L     BNA090+@OP1(,@BR),@XR     LOAD DOPE VECTOR CADDR
                               3725 *
                               3726 * MOVE DIMENSION IN BINARY FORM INTO THE 1ST DIMENSION OF THE ARITH
                               3727 * DOPE VECTOR
                               3728 *
    09D0 8C 01 01 1A6A         3729 BNA140 MVC   B@ACD1(,@XR),B$BINO(B@LDMN)  SET 1ST DIMENSION
                               3730 *
                               3731 * GET NEXT CHARACTER FOR 2ND DIMENSION EVALUATION
                               3732 *
    09D5 C0 87 0867            3733 BNA150 B     B$GETC                    LINK TO GET 1ST CHAR OF 2ND DIM
                               3734 *
                               3735 * CONVERT 2ND DIMENSION TO BINARY FORM FROM DECIMAL FORM
                               3736 *
    09D9 C0 87 19F2            3737 BNA160 B     B$ZDBN                    LINK TO CONVERT 2ND DIM
                               3738 *
    09DD 75 02 AE              3739        L     BNA090+@OP1(,@BR),@XR     LOAD DOPE VECTOR CADDR
                               3740 *
                               3741 * SET VECTOR AND MATRIX BITS ON IN DOPE VECTOR TO INDICATE A DEFINED
                               3742 * ARRAY OF TWO DIMENSIONS
                               3743 *
    09E0 BA C0 00              3744 BNA170 SBN   B@AFLG(,@XR),B@D2MK       SET VECTOR AND MATRIX BITS ON
                               3745 *
                               3746 * SET 2ND DIMENSION IN DOPE VECTOR
                               3747 *
    09E3 8C 01 03 1A6A         3748 BNA180 MVC   B@ACD2(,@XR),B$BINO(B@LDMN)  SET 2ND DIMENSION
                               3749 *
                               3750 * GET NEXT CHARACTER AND TEST FOR END OF DIMENSION STATEMENT
                               3751 *
    09E8 C0 87 0867            3752 BNA190 B     B$GETC                    GET NEXT CHARACTER
    09EC BD 1E 00              3753        CLI   B@CHAR(,@XR),B@EOST       IF CHAR NOT A STMT TERMINATOR
    09EF D0 01 7B              3754        BNE   BNA020(,@BR)              * GO PROCESS NEXT ARRAY IN LIST
    09F2 C0 87 0700            3755        B     B$DIST                    RETURN TO DISTRIBUTOR

                               3757 ***********************************************************************
                               3758 * 'DIMENSION' STATEMENT EQUATES
                               3759 ***********************************************************************
                               3760 *
    09F6 0001             09F7 3761 BNABNI DC    IL(@VADDR)'1'             BINARY 1
                               3762 *
                               3763 ***********************************************************************
                               3764 *
                               3765 * END OF 'DIMENSION' ROUTINE CODING
                               3766 *
                               3768 ***********************************************************************
                               3769 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               3770 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               3771 *                                                                     *
                               3772 ***********************************************************************
                               3773 *STATUS                                                               *
                               3774 *   VERSION 1 MODIFICATION 0                                          *
                               3775 *                                                                     *
                               3776 *FUNCTION                                                             *
                               3777 *   BMMATA IS EXECUTED TO TRANSLATE MAT ASSIGNMENT STATEMENTS AS THEY *
                               3778 *   OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO   *
                               3779 *   PLACE THE PSEUDOCODE IN VIRTUAL MEMORY.                           *
                               3780 *                                                                     *
                               3781 *ENTRY POINTS                                                         *
                               3782 *   BMMATA HAS ONLY ONE ENTRY POINT:                                  *
                               3783 *        BMMATA - TRANSLATE MAT ASSIGNMENT STATEMENT                  *
                               3784 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               3785 *        B     BMMATA                                                 *
                               3786 *                                                                     *
                               3787 *INPUT                                                                *
                               3788 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               3789 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               3790 *     LEADING KEYWORD, MAT.                                           *
                               3791 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               3792 *     FIRST CHARACTER IN THE LEADING KEYWORD, MAT.                    *
                               3793 *                                                                     *
                               3794 *OUTPUT                                                               *
                               3795 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               3796 *     GENERATED BY BMMATA IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               3797 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               3798 *     SEQUENCES.                                                      *
                               3799 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               3800 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               3801 *                                                                     *
                               3802 *EXTERNAL REFERENCES                                                  *
                               3803 *      B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.  *
                               3804 *      B$PUTC - (B$PCAD, B$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY   *
                               3805 *               OUTPUT ROUTINE.                                       *
                               3806 *      B$SCAN - ENTRY TO COMPILER ARITHMETIC EXPRESSION SCAN          *
                               3807 *               ROUTINE.                                              *
                               3808 *      B$MATR - (B$MGSM, B$MPSW, B$MBSW) - ENTRY TO COMPILER MATRIX.  *
                               3809 *      BLMFBK - ENTRY TO MAT ASSIGNMENT FUNCTION BUCKET.              *
                               3810 *      B$CSBF - ENTRY TO DISK RESIDENT PMC GENERATORS.                *
                               3811 *      B$CSXA - STARTING CORE ADDRESS FOR EXCESS CORE.                *
                               3812 *      $EXFTR - EXTENSION FACTOR                                      *
                               3813 *      B$DIST - (B$DST2) - ENTRY TO BASIC COMPILER DISTRIBUTOR.       *
                               3814 *                                                                     *
                               3815 *EXITS, NORMAL                                                        *
                               3816 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               3817 *                                                                     *
                               3818 *EXITS, ERROR                                                         *
                               3819 *   N/A                                                               *
                               3820 *                                                                     *
                               3821 *TABLES/WORK AREAS                                                    *
                               3822 *   MATRIX FUNCTION TABLE - INTERNAL TO BMMATA.  THIS TABLE CONTAINS  *
                               3823 *   MATRIX FUNCTION INSTRUCTIONS ASSOCIATED WITH EVERY MATRIX         *
                               3824 *   EXPRESSION EXCEPT SIMPLE ASSIGNMENT AND SCALAR MULTIPLICATION.    *
                               3825 *   EACH OF THE EIGHT 6-BYTE TABLE ENTRIES CONTAINS A 3-BYTE          *
                               3826 *   FUNCTION CHARACTER IDENTIFIER AND THE 3-BYTE MATRIX FUNCTION      *
                               3827 *   INSTRUCTION REQUIRED TO PERFORM THAT FUNCTION AT RUN-TIME.        *
                               3828 *   MATRIX FUNCTION BUCKET - 3 BYTES (B$MFBK), FOR THE EXTERNAL       *
                               3829 *   CORE-RESIDENT BUCKET, USED TO ACCUMULATE MATRIX EXPRESSION        *
                               3830 *   FUNCTION CHARACTERS.                                              *
                               3831 *                                                                     *
                               3832 *ATTRIBUTES                                                           *
                               3833 *   BMMATA IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               3834 *                                                                     *
                               3835 *CHARACTER CODE DEPENDENCY                                            *
                               3836 *   THE OPERATION OF THIS MODULE DEPENDS UPON AN INTERNAL REPRESENTA- *
                               3837 *   TION OF THE EXTERNAL CHARACTER SET WHICH IS EQUIVALENT TO THE ONE *
                               3838 *   USED AT ASSEMBLY TIME. THE CODING HAS BEEN ARRANGED SO THAT       *
                               3839 *   REDEFINITION OF CHARACTER CONSTANTS, BY REASSEMBLY, WILL RESULT   *
                               3840 *   IN A CORRECT MODULE FOR THE NEW DEFINITIONS.                      *
                               3841 *                                                                     *
                               3842 *NOTES                                                                *
                               3843 *   ERROR PROCEDURES                                                  *
                               3844 *      N/A                                                            *
                               3845 *                                                                     *
                               3846 *   REGISTER USAGE                                                    *
                               3847 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               3848 *                                                                     *
                               3849 *   SAVED/RESTORED AREAS                                              *
                               3850 *      N/A                                                            *
                               3851 *                                                                     *
                               3852 *   MODIFICATION CONSIDERATIONS                                       *
                               3853 *      BMMATA RESIDES ON TWO SECTORS AND IS CO-RESIDENT ON THE     1-4*
                               3854 *      SECOND SECTOR WITH BPREAD.  ANY MODIFICATIONS MUST MAINTAIN 1-4*
                               3855 *      LINKAGE BETWEEN THE TWO SECTORS, CONSIDER ANY CHANGE IN THE 1-4*
                               3856 *      ENTRY ADDRESS OF BPREAD, AND REALIZE THE LIMITATION OF THE  1-4*
                               3857 *      SECTOR BOUNDARY UPON SIZE.                                  1-4*
                               3858 *                                                                     *
                               3859 *   REQUIRED MODULES                                                  *
                               3860 *        @SYSEQ - COMMON SYSTEM EQUATES.                              *
                               3861 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.   *
                               3862 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.      *
                               3863 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               3864 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               3865 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               3866 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               3867 *        $B$EQU - COMPILER FIXED EQUATES.                             *
                               3868 *        $B@EQU - COMPILER SYSTEM EQUATES                             *
                               3869 *                                                                     *
                               3870 *   OTHER                                                             *
                               3871 *      BMMATA IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               3872 ***********************************************************************

    0A00                       3874        ORG   *,256,0                   BEGIN AT CORE PAGE BOUNDARY
                          0A00 3875        USING *,@BR                     DEFINE BASE ADDR FOR CORE PAGE
                               3876 *
                               3877 * ENTER BMMATA - MAT ASSIGNMENT STATEMENT ROUTINE
                               3878 *
                          0A00 3879 BMMATA EQU   *
    0A00 74 01 F4              3880        ST    BMMCA2(,@BR),@BR
                               3881 *
                               3882 * SET MATRIX PROCESSING ROUTINE NOT TO GENERATE PMC AND ADVANCE POINTER
                               3883 * TO REFERENCE CHAR BEFORE 1ST MAT REFERENCE
                               3884 *
    0A03 3C 02 0873            3885        MVI   B$NUMC,B@LMAT-1           SET GET TO SKIP TO 'T' IN MAT
    0A07 C0 87 0867            3886        B     B$GETC                    LINK TO ADVANCE POINTER
    0A0B 3B 07 1981            3887        SBF   B$MPSW,B$MPMK             SET PUT SWITCH OFF
    0A0F 3C 00 0A39            3888        MVI   B$PERC,@ZERO              INITIALIZE ERROR CODE TO ZERO
    0A13 C0 87 18F3            3889        B     B$MATR                    LINK TO PROCESS MAT REFERENCE
    0A17 C0 87 0867            3890        B     B$GETC                    LINK TO GET NEXT CHAR
    0A1B 3A 07 1981            3891        SBN   B$MPSW,B$MPMK             SET PUT SWITCH ON
                               3892 *
                               3893 * TEST CHAR FOR INDICATION OF MAT MULTIPLICATION BY A SCALAR VALUE
                               3894 *
    0A1F BD 4D 00              3895        CLI   B@CHAR(,@XR),B@LPAR       IF SCALAR MULTIPLICATION
    0A22 F2 81 9C              3896        JE    BMM060                    * GO PROCESS EXPRESSION
                               3897 *
                               3898 * SET UP FUNCTION SAVE BUCKET FOR COMPARISON
                               3899 *
    0A25 2C 00 1B8F 00         3900        MVC   B$MFBK+BMMBK0,B@CHAR(1,@XR)  MOVE CHAR TO 1ST BUCKET BYTE.
    0A2A C0 87 0867            3901        B     B$GETC                    LINK TO GET NEXT CHAP
    0A2E BD 1E 00              3902        CLI   B@CHAR(,@XR),B@EOST       IF CHAR IS NOT AN EOS
    0A31 F2 01 09              3903        JNE   BMM005                    * GO SET 2ND CHAR IN BUCKET
    0A34 7C 6F F2              3904        MVI   BMMPBA(,@BR),BMM160-BMMAT2  SET BR ADDR TO 4TH ENTRY PT
    0A37 7C 6F EE              3905        MVI   BMM095+@D1(,@BR),BMM160-BMMAT2  SET RR ADDR TO 4TH ENT PT
    0A3A F2 87 8A              3906        J     BMM070                    GO CALL SECOND SEGMENT
    0A3D 2C 00 1B90 00         3907 BMM005 MVC   B$MFBK+BMMBK1,B@CHAR(1,@XR)  MOVE CHAR TO 2ND BUCKET 'ME
    0A42 C0 87 0867            3908        B     B$GETC                    LINK TO GET NEXT CHAR
    0A46 2C 00 1B91 00         3909        MVC   B$MFBK+BMMBK2,B@CHAR(1,@XR)  MOVE CHAR TO 3RD BUCKET BYTE
    0A4B C0 87 0867            3910        B     B$GETC                    LINK TO GET NEXT CHAR
                               3911 *
                               3912 * SET POINTER TO 2ND BUCKET BYTE AND TEST FOC CHAR BEING '.', '-' OR 'A'
                               3913 *
    0A4F C2 02 1B90            3914        LA    B$MFBK+BMMBK1,@XR         SET POINTER TO 2ND CHAR OF FUNC
    0A53 BD 4E 00              3915        CLI   B@CHAR(,@XR),B@PLUS       IF CHAR IS A
    0A56 F2 81 0C              3916        JE    BMM010                    * GO SET AN!) CALL 2ND SEGMENT
    0A59 BD 60 00              3917        CLI   B@CHAR(,@XR),B@MINS       IF CHAR LI A '-'
    0A5C F2 81 06              3918        JE    BMM010                    * GO SET ALD CALL 2ND SEGMENT
    0A5F BD 5C 00              3919        CLI   B@CHAR(,@XR),B@MULT       IF CHAR NOT
    0A62 F2 01 09              3920        JNE   BMM020                    * GO SET FUNC TYPE
                               3921 *
                               3922 * SET SECOND SEGMENT BRANCH ADDRESS FOR MAIN ENTRY POINT
                               3923 *
    0A65 7C 00 F2              3924 BMM010 MVI   BMMPBA(,@BR),BMM100-BMMAT2  SET BR ADDR TO MAIN ENTRY PT
    0A68 7C 00 EE              3925        MVI   BMM095+@D1(,@BR),BMM100-BMMAT2  SET BR ADDR TO MAIN ENT PT
    0A6B F2 87 59              3926        J     BMM070                    GO CALL SECOND SEGMENT
                               3927 *
                               3928 * SET BRANCH ADDRESS IN CALLING SEQUENCE FOR SEG-2 SECONDARY ENTRY PT
                               3929 *
    0A6E 7C 2C F2              3930 BMM020 MVI   BMMPBA(,@BR),BMM110-BMMAT2  SET BR ADDR TO MAIN ENTRY PT
    0A71 7C 2C EE              3931        MVI   BMM095+@D1(,@BR),BMM110-BMMAT2  SET 3R ADDR TO 2ND ENT PT
                               3932 *
                               3933 * TEST DELIMITEP FOR BEING A STATEMENT TERMINATOR
                               3934 *
    0A74 35 02 0878            3935        L     B$GPTR,@XR                RESTORE TEXT POINTER
    0A78 BD 1E 00              3936        CLI   B@CHAR(,@XR),B@EOST       IF DELIMITER IS NOT AN EOS
    0A7B F2 01 07              3937        JNE   BMM030                    * GO PROCESS FUNC SUBSCRIPT
    0A7E C0 87 093A            3938        B     B$PUTC                    LINK TO GENERATE 'SDO' PMC
    0A82 F2 87 42              3939        J     BMM070                    GO CALL SECOND SEGMENT
                               3940 *
                               3941 * TEST IF FUNCTION IS 'INV' OR 'TRN'
                               3942 *
    0A85 3D D5 1B90            3943 BMM030 CLI   B$MFBK+BMMBK1,BMMINV      IF FUNC IS 'INV'
    0A89 F2 81 07              3944        JE    BMM040                    * GO PROCESS NAT REFERENCE
    0A8C 3D D9 1B90            3945        CLI   B$MFBK+BMMBK1,BMMTRN      IF FUNC IS 'TRN'
    0A90 F2 01 0F              3946        JNE   BMM050                    * GO PROCESS OTHER FUNCTIONS
                               3947 *
                               3948 * PROCESS MATRIX REFERENCED 'INV' OR 'TRN'.
                               3949 *
    0A93 C0 87 093A            3950 BMM040 B     B$PUTC                    LINK TO GENERATE 'SDO' PMC
    0A97 C0 87 18F3            3951        B     B$MATR                    LINK TO PROCESS MAT REFERENCE
    0A9B C0 87 0867            3952        B     B$GETC                    LINK TO GET NEXT CHAR
    0A9F F2 87 25              3953        J     BMM070                    GO CALL SECOND SEGMENT
                               3954 *
                               3955 * PROCESS MATRIX FOR 'IDN', 'CON', OR 'ZER' FUNC
                               3956 *
    0AA2 3D 00 0A39            3957 BMM050 CLI   B$PERC,@ZERO              IF ERROR IS FOR UNDEFINED ARRAY
    0AA6 C0 01 1AE6            3958        BNE   B$RMRK                    * NRURN TO DIST VIA REMARK
    0AAA 3B 07 18FF            3959        SBF   B$MGSW,B$MGMK             SET MAT RTN NOT TO CALL GET RTN
    0AAE 3A 07 1903            3960        SBN   B$MBSW,B$MBMK             SET TO SKIP DOPE VECTOR STK
    0AB2 C0 87 18F3            3961        B     B$MATR                    LINK TO REDIM AND GENERATE PMC
    0AB6 3B 07 1903            3962        SBF   B$MBSW,B$MBMK             SET SN NOT TO SKIP D.V. STK
    0ABA 3A 07 18FF            3963        SBN   B$MGSW,B$MGMK             ENABLE MAT RTN TO CALL GET RTN
    0ABE F2 87 06              3964        J     BMM070                    GO CALL SECOND SEGMENT
                               3965 *
                               3966 * SET BRANCH ADDRESS FOR 3RD ENTRY POINT BEFORE GOING TO CALLING SEG
                               3967 *
    0AC1 7C 4C F2              3968 BMM060 MVI   BMMPBA(,@BR),BMM140-BMMAT2  SET BR ADDR FOR 3RD ENTRY PT
    0AC4 7C 4C EE              3969        MVI   BMM095+@D1(,@BR),BMM140-BMMAT2 SET BR ADDR TO 3RD ENT PT

                               3971 ***********************************************************************
                               3972 * MAT ASSIGNMENT 2ND SEGMENT CALLING SEQUENCE ROUTINE
                               3973 ***********************************************************************
                               3974 *
                               3975 * TEST WHETHER CURRENT SEGMENT WAS CORE OR DISK RESIDENT
                               3976 *
    0AC7 5D 00 F3 F1           3977 BMM070 CLC   BMMCA2-1(,@BR),BMMPBA-1(@CADDR-1,@BR) IF CURR SEG FR DISK
    0ACB F2 81 10              3978        JE    BMM080                    * GO LOAD & EXEC 2ND SEGMENT
                               3979 *
                               3980 * CURRENT SEGMENT WAS CORE RESIDENT TEST WHETHER 2ND SEGMENT HAS
                               3981 * ALSO BEEN LOADED INTO CORE
                               3982 *
    0ACE 5C 01 F7 F9           3983        MVC   BMMFCP(,@BR),BMMFPE(@CADDR,@BR) SET FINAL CORE PAGE ADDR
    0AD2 4E 00 F6 043B         3984        ALC   BMMFCP-1(,@BR),$EXFTR(1)  CALC MAX PROCESSOR CORE PAGE
                               3985 *
    0AD7 5D 01 F4 F7           3986        CLC   BMMCA2(,@BR),BMMFCP(@CADDR,@BR)  IF 2ND SEGMENT IN CORE
    0ADB F2 82 0B              3987        JL    BMM090                    * GO SET TO EXEC 2ND SEGMENT
                               3988 *
                               3989 * 2ND SEGMENT IS DISK RESIDENT - ESTABLISH DISTRIBUTOR PARAMETERS FOR
                               3990 * CORE-LOADING AND EXECIIT!NG DE 2ND SEGMENT
                               3991 *
    0ADE 5C 01 F4 F2           3992 BMM080 MVC   BMMCA2(,@BR),BMMPBA(@CADDR,@BR)  SET UP DISKLOAD ADDR
                               3993 *
                               3994 * EXIT TO DISTRIBUTOR FOR 2ND SEGMENT CORELOAD AND EXECUTION
                               3995 *
    0AE2 D2 02 F3              3996        LA    BMMAD2(,@BR),@XR          LOAD DISTRIBUTOR PARM CADDR
    0AE5 C0 87 073A            3997        B     B$DST2                    GO LOAD & EXECUTE 2ND SEGMENT
                               3998 *
                               3999 * 2ND SEGMENT IS CORE RESIDENT- BRANCH TO NEXT CONSECUTIVE CORE PAGE
                               4000 * AND CONTINUE MAT ASSIGNMENT EXECUTION
                               4001 *
    0AE9 76 01 F0              4002 BMM090 A     BMMBLS(,@BR),@BR          ADJUST BASE ADDR FOR 2ND SEG
    0AEC D0 87 00              4003 BMM095 B     *-*(,@BR)                 GO EXECUTE 2ND SEGMENT

                               4005 ***********************************************************************
                               4006 * MAT ASSIGNMENT SEGMENT-1 CONSTANTS AND WORK AREAS, AND EQUATES
                               4007 ***********************************************************************
                               4008 *
    0AEF 0100             0AF0 4009 BMMBLS DC    AL(@CADDR)(B@BLSZ)        * REFERNECE NEXT PAGE BOUNDARY
                               4010 *
                               4011 *
    0AF1                  0AF2 4012 BMMPBA DS    CL(@CADDR)                PROCESSOR DISK BUFFER CADDR
    0AF1                       4013        ORG   *-@CADDR                  INITIALIZE DISK BUFFER CADDR TO
    0AF1 0600             0AF2 4014        DC    AL(@CADDR)(B$CSBF)        * REFERENCE PAGE BOUNDARY
                               4015 *
                          00D5 4016 BMMINV EQU   C'N'                      COMPARISON FOR FUNC 'INV'
                          00D9 4017 BMMTRN EQU   C'R'                      COMPARISON FOR FUNC 'TRN'
                               4018 *
                          0AF3 4019 BMMAD2 EQU   *                         DISTR PARAMS FOR SEG-2 EXEC
    0AF3                  0AF4 4020 BMMCA2 DS    CL(@CADDR)                MAT ASSIGNMENT SEG CORE ADDRESS
    0AF5 0C               0AF5 4021 BMMIA2 DC    AL1(B@DMAT+BMMPSI)        BMMATA SEG-2 PHYS SECTOR ADDR
                               4022 *
    0AF6                  0AF7 4023 BMMFCP DS    CL(@CADDR)                FINAL AVAILABLE CORE PAGE ADDR
    0AF8 1F00             0AF9 4024 BMMFPE DC    AL(@CADDR)(B$CSXA-B@BLSZ) FINAL PAGE BEFORE EXTENSION
                               4025 *
                               4026 * EQUATES
                               4027 *
                          0000 4028 BMMBK0 EQU   0                         DISP TO 1ST BUCKET BYTE
                          0001 4029 BMMBK1 EQU   1                         DISP TO 2ND BUCKET BYTE
                          0002 4030 BMMBK2 EQU   2                         DISP TO 3RD BUCKET BYTE
                               4031 *
                          0000 4032 BMMSG2 EQU   0                         DISP FOR BMMATA SEG-2 ENTRY PT
                          0004 4033 BMMPSI EQU   X'04'                     PHYS SECTOR ADM INCREMENT

                               4035 ***********************************************************************
                               4036 * MAT ASSIGNMENT SECOND SEGMENT
                               4037 ***********************************************************************
                               4038 *
                               4039 * ESTABLISH MAT ASSIGNMENT SEGMENT-2 ADDRESSABILITY
                               4040 *
    0B00                       4041        ORG   BMMATA+B@BLSZ             BEGIN SEG-2 AT PAGE BOUNDARY
                          0B00 4042        USING *,@BR                     DEFINE SEG-2 BASE ADDRESS
                          0B00 4043 BMMAT2 EQU   *                         BMMATA - SEG-2 MAIN ENTRY PT
                               4044 *
                               4045 * GENERATE THE 'SD0' PMC IN VIRTUAL MEMORY
                               4046 *
    0B00 C0 87 093A            4047 BMM100 B     B$PUTC                    LINK TO GENERATE 'SDO' PMC
```

```
                               4048 *
                               4049 * PROCESS FIRST MATRIX REFERENCE IN MAT OPERATION
                               4050 *
    0B04 3C 00 0873            4051        MVI   B$NUMC,B@GETS             SET GET NOT TO SKIP CHAR
    0B08 C2 02 1B8F            4052        LA    B$MFBK+BMMBK0,@XR         SET PTR TO 15T BUCKET BYTE
    0B0C 3B 07 18FF            4053        SBF   B$MGSW,B$MGMK             DISABLE BMATXR TO CALL GET RTN
    0B10 C0 87 18F3            4054        B     B$MATR                    LINK TO PROCESS MAT REFERENCE
                               4055 *
                               4056 * PROCESS THE SECOND MATRIX REFERENCE IN MAT OPERATION
                               4057 *
    0B14 3C 00 0873            4058        MVI   B$NUMC,B@GETS             SET GET NOT TO SKIP CHAR
    0B18 C2 02 1B91            4059        LA    B$MFBK+BMMBK2,@XR         SET PTR TO 3RD BUCKET BYTE
    0B1C C0 87 18F3            4060        B     B$MATR                    LINK TO PROCESS MAT REFERENCE
    0B20 3A 07 18FF            4061        SBN   B$MGSW,B$MGMK             ENABLE BMATXR TO CALL GET RTN
                               4062 *
                               4063 * MOVE BLANKS INTO THE 1ST AND 3RD BYTES OF THE SAVE BUCKET
                               4064 *
    0B24 3C 40 1B8F            4065        MVI   B$MFBK+BMMBK0,B@BLNK      SET 15T BUCKET BYTE TO BLANK
    0B28 3C 40 1B91            4066        MVI   B$MFBK+BMMBK2,B@BLNK      SET 3RD BUCKET BYTE TO BLANK
                               4067 *
                               4068 * SEARCH TABLE FOR MATCHING FUNCTION - 2ND ENTRY PT FOR 2ND SEGMENT
                               4069 *
    0B2C D2 02 99              4070 BMM110 LA    BMMTBS(,@BR),@XR          LOAD FUNC TBL POINTER
    0B2F E2 02 06              4071 BMM120 LA    BMMTEL(,@XR),@XR          INCREMENT POINTER TO NEXT ENTRY
    0B32 2D 02 1B91 02         4072        CLC   B$MFBK+BMMBK2,BMMFND(B@LIFN,@XR)  IF FUNC = TBL ENTRY
    0B37 D0 01 2F              4073        BNE   BMM120(,@BR)              GO COMPARE FUNC TO NXT TBL ENT
                               4074 *
                               4075 * GENERATE THE PMC ASSOCIATED WITH THE TABLE ENTRY FUNCTION
                               4076 *
    0B3A E2 02 03              4077 BMM130 LA    B@LIFN(,@XR),@XR          LOAD CADDR OF 'MF1' INSTR
    0B3D 34 02 0A40            4078        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'MF1'
    0B41 3C 02 0A41            4079        MVI   B$PNBY,B@LMF1-1           SET LNG PARM OF PUT FOR 'MF1'
    0B45 C0 87 093A            4080        B     B$PUTC                    LINK TO GENERATE PMC
                               4081 *
                               4082 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               4083 *
    0B49 F2 87 1F              4084        J     BMM150                    GO CALL DISTRIBUTOR
                               4085 *
                               4086 *
                               4087 * GENERATE THE 'SDO' PMC IN VIRT MEM BEFORE PROCESSING THE EXPRESSION
                               4088 *
    0B4C C0 87 093A            4089 BMM140 B     B$PUTC                    LINK TO GENERATE 'SD0' PMC
                               4090 *
                               4091 * PROCESS ARITHMETIC EXPRESSION AND MAT REFERENCE
                               4092 *
    0B50 C0 87 1514            4093        B     B$SCAN                    LINK TO PROCESS ARITH DPP
    0B54 C0 87 0867            4094        B     B$GETC                    LINK TO GET NEXT CHAR
    0B58 C0 87 18F3            4095        B     B$MATR                    LINK TO PROCESS MAT REFERENCE
                               4096 *
                               4097 * GENERATE AN 'MSM' INSTR IN VIRTUAL MEMORY
                               4098 *
    0B5C D2 02 99              4099        LA    BMMMSC(,@BR),@XR          LOAD CADDR OF 'MSM' INSTR
    0B5F 34 02 0A40            4100        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'MSM'
    0B63 3C 02 0A41            4101        MVI   B$PNBY,B@LMSM-1           SET LNG PARM OF PUT FOR 'MSM'
    0B67 C0 87 093A            4102        B     B$PUTC                    LINK TO GENERATE 'NSM' PMC
                               4103 *
                               4104 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               4105 *
    0B6B C0 87 0700            4106 BMM150 B     B$DIST                        RETURN TO DISTRIBUTOR
                               4107 *
                               4108 * GENERATE 'SDO' FOR 1ST MAT REFERENCE AND PROCESS 2ND MAT REFERENCE
                               4109 *
    0B6F C0 87 093A            4110 BMM160 B     B$PUTC                        LINK TO GENERATE 'SDO' PMC
    0B73 3C 00 0873            4111        MVI   B$NUMC,B@GETS                 DISABLE GET RTN TO GET CHARS
    0B77 3B 07 18FF            4112        SBF   B$MGSW,B$MGMK                 SET GET RTN NOT TO ADVANCE PTR
    0B7B C2 02 1B8F            4113        LA    B$MFBK+BMMBK0,@XR             SET PTR TO MAT REFERENCE
    0B7F C0 87 18F3            4114        B     B$MATR                        LINK TO PROCESS MAT REFERENCE
    0B83 3A 07 18FF            4115        SBN   B$MGSW,B$MGMK                 ENABLE GET RTN TO GET CHARS
                               4116 *
                               4117 * GENERATE AN 'MF2' INSTR IN VIRTUAL MEMORY
                               4118 *
    0B87 D2 02 9C              4119        LA    BMMM2C(,@BR),@XR              LOAD CADDR OF 'MF2' INSTR
    0B8A 34 02 0A40            4120        ST    B$PCAD,@XR                    SET VADDR PARM OF PUT FOR 'MF2'
    0B8E 3C 02 0A41            4121        MVI   B$PNBY,B@LMF2-1               SET LNG PARM OF PUT FOR 'MF2'
    0B92 C0 87 093A            4122        B     B$PUTC                        LINK TO GENERATE 'MF2' PMC
    0B96 D0 87 6B              4123        B     BMM150(,@BR)                  RETURN TO DIST
```

```
                               4125 ************************************************************************
                               4126 * MAT ASSIGNMENT SEGMENT-2 STORAGE AND PARAMETER AREA
                               4127 ************************************************************************
                               4128 *
    0B99 1E               0B99 4129 BMMMSC DC    AL(B@LCOP)(B@CMSM)        CADDR OF 'MSM' INSTR OPCODE
    0B9A 4264             0B9B 4130 BMMMSO DC    AL(B@LCVA)(V$MSMY)        CADDR OF 'MSM' INSTR OPERAND
                               4131 *
    0B9C 1A               0B9C 4132 BMMM2C DC    AL(B@LCOP)(B@CMF2)        CADDR OF 'MF2' INSTR OPCODE
    0B9D 43A0             0B9E 4133 BMMM2O DC    AL(B@LCVA)(V$MASN)        CADDR OF 'MF2' INSTR OPERAND

                               4135 ************************************************************************
                               4136 * 'MAT' ASSIGNMENT STATEMENT MATRIX FUNCTION TABLE
                               4137 ************************************************************************
                               4138 *
                          0006 4139 BMMTEL EQU   6                         LENGTH OF TABLE ENTRY
                          0003 4140 BMMPID EQU   3                         LENGTH OF PSEUDO INSTR DISP
                          0002 4141 BMMFND EQU   2                         LENGTH OF FUNCTION DISP
                          0B9F 4142 BMMTAB EQU   *                         BEGINNING OF MAT FUNCTION TBL
                          0B99 4143 BMMTBS EQU   BMMTAB-BMMTEL             INITIAL FUNC TOL ENTRY POINT
                               4144 *
    0B9F 404E40           0BA1 4145        DC    CL(B@LIFN)' + '           FUNC FOR MATRIX ADDITION
    0BA2 1C               0BA2 4146        DC    AL(B@LCOP)(B@CMF3)        CADDR OF 'MF3' INSTR OPCODE
    0BA3 4007             0BA4 4147        DC    AL(B@LCVA)(V$MADD)        CADDR OF 'MF3' INSTR OPERAND
                               4148 *
    0BA5 406040           0BA7 4149        DC    CL(B@LIFN)' - '           FUNC FOR MATRIX SUBTRACTION
    0BA8 1C               0BA8 4150        DC    AL(B@LCOP)(B@CMF3)        CADDR FOR 'MF3' INSTR OPCODE
    0BA9 4000             0BAA 4151        DC    AL(B@LCVA)(V$MSUB)        CADDR FOR 'MF3' INSTR OPERAND
                               4152 *
    0BAB 405C40           0BAD 4153        DC    CL(B@LIFN)' * '           FUNC FOR MATRIX MULTIPLICATION
    0BAE 1C               0BAE 4154        DC    AL(B@LCOP)(B@CMF3)        CADDR FOR 'MF3' INSTR OPCODE
    0BAF 4100             0BB0 4155        DC    AL(B@LCVA)(V$MMPY)        CADDR FOR 'MF3' INSTR OPERAND
                               4156 *
    0BB1 C9D5E5           0BB3 4157        DC    CL(B@LIFN)'INV'           FUNC FOR MATRIX INVERSION
    0BB4 1A               0BB4 4158        DC    AL(B@LCOP)(B@CMF2)        CADDR FOR 'MF2' INSTR OPCODE
    0BB5 4500             0BB6 4159        DC    AL(B@LCVA)(V$MINV)        CADDR FOR 'MF2' INSTR OPERAND
                               4160 *
    0BB7 E3D9D5           0BB9 4161        DC    CL(B@LIFN)'TRN'           FUNC FOR MATRIX TRANSPOSITION
    0BBA 1A               0BBA 4162        DC    AL(B@LCOP)(B@CMF2)        CADDR FOR 'MF2' INSTR OPCODE
    0BBB 4400             0BBC 4163        DC    AL(B@LCVA)(V$MTRN)        CADDR FOR 'MF2' INSTR OPERAND
                               4164 *
    0BBD E9C5D9           0BBF 4165        DC    CL(B@LIFN)'ZER'           FUNC FOR MAT INITIALLY ZEROES
    0BC0 18               0BC0 4166        DC    AL(B@LCOP)(B@CMF1)        CADDR OF 'MF1' INSTR OPCODE
    0BC1 432B             0BC2 4167        DC    AL(B@LCVA)(V$MZER)        CADDR OF 'MF1' INSTR OPERAND
                               4168 *
    0BC3 C3D6D5           0BC5 4169        DC    CL(B@LIFN)'CON'           FUNC FOR MAT INITIALLY ONE'S
    0BC6 18               0BC6 4170        DC    AL(B@LCOP)(B@CMF1)        CADDR OF 'MF1' INSTR OPCODE
    0BC7 4324             0BC8 4171        DC    AL(B@LCVA)(V$MCON)        CADDR OF 'MF1' INSTR OPERAND
                               4172 *
    0BC9 C9C4D5           0BCB 4173        DC    CL(B@LIFN)'IDN'           FUNC FOR MATRIX IDENTITY
    0BCC 18               0BCC 4174        DC    AL(B@LCOP)(B@CMF1)        CADDR FOR 'MF1' INSTR OPCODE
    0BCD 4300             0BCE 4175        DC    AL(B@LCVA)(V$MIDN)        CADDR FOR 'MF1' INSTR OPERAND
                               4176 *
                               4177 ************************************************************************
                               4178 *
                               4179 * END OF 'MAT ASSIGNMENT' STATEMENT CODING
                               4180 *
                               4182 ***********************************************************************
                               4183 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               4184 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               4185 *                                                                     *
                               4186 ***********************************************************************
                               4187 *STATUS                                                               *
                               4188 *   VERSION 1 MODIFICATION 0                                          *
                               4189 *                                                                     *
                               4190 *FUNCTION                                                             *
                               4191 *   BPREAD IS EXECUTED TO TRANSLATE READ STATEMENTS AS THEY OCCUR IN  *
                               4192 *   A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE  *
                               4193 *   PSEUDOCODE IN VIRTUAL MEMORY.                                     *
                               4194 *                                                                     *
                               4195 *ENTRY POINTS                                                         *
                               4196 *   BPREAD HAS ONLY ONE ENTRY POINT:                                  *
                               4197 *        BPREAD - TRANSLATE READ STATEMENT                            *
                               4198 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               4199 *        B     BPREAD                                                 *
                               4200 *                                                                     *
                               4201 *INPUT                                                                *
                               4202 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               4203 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER OF THE       *
                               4204 *     LEADING KEYWORD, READ.                                          *
                               4205 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               4206 *     FIRST CHARACTER IN THE LEADING KEYWORD, READ.                   *
                               4207 *                                                                     *
                               4208 *OUTPUT                                                               *
                               4209 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               4210 *     GENERATED BY BPREAD IS STORED IN THE PEST AVAILABLE VIRTUAL     *
                               4211 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               4212 *     SEQUENCES.                                                      *
                               4213 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               4214 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               4215 *                                                                     *
                               4216 *EXTERNAL REFERENCES                                                  *
                               4217 *        BSGETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIEVAL RTN.    *
                               4218 *        BSPUTC - (B$PCAD, B$PNBY) - ENTRY TO COMPILER VIRTUAL        *
                               4219 *                 MEMORY OUTPUT ROUTINE.                              *
                               4220 *        B$LIST - ENTRY TO BASIC COMPILER LIST ADDRESS ROUTINE.       *
                               4221 *        B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                *
                               4222 *                                                                     *
                               4223 *EXITS, NORMAL                                                        *
                               4224 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               4225 *                                                                     *
                               4226 *EXITS, ERROR                                                         *
                               4227 *   N/A                                                               *
                               4228 *                                                                     *
                               4229 *TABLES/WORK AREAS                                                    *
                               4230 *   N/A                                                               *
                               4231 *                                                                     *
                               4232 *ATTRIBUTES                                                           *
                               4233 *   BPREAD IS NATURALLY RELOCATABLE AND REUSABLE                      *
                               4234 *                                                                     *
                               4235 *CHARACTER CODE DEPENDENCY                                            *
                               4236 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               4237 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               4238 *                                                                     *
                               4239 *NOTES                                                                *
                               4240 *   ERROR PROCEDURES                                                  *
                               4241 *      N/A                                                            *
                               4242 *                                                                     *
                               4243 *   REGISTER USAGE                                                    *
                               4244 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION    *
                               4245 *                                                                     *
                               4246 *   SAVED/RESTORED AREAS                                              *
                               4247 *      N/A                                                            *
                               4248 *                                                                     *
                               4249 *   MODIFICATION CONSIDERATIONS                                       *
                               4250 *      BPREAD IS CO-RESIDENT ON A SECTOR WITH BMMATA.             1-4*
                               4251 *      ANY MODIFICATION SHOULD CONSIDER THE CO-RESIDENCY AND       1-4*
                               4252 *      THE LIMITATION OF THE SECTOR BOUNDARY ON SIZE.              1-4*
                               4253 *                                                                     *
                               4254 *   REQUIRED MODULES                                                  *
                               4255 *        @SYSEQ - COMMON SYSTEM EQUATES.                              *
                               4256 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.   *
                               4257 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.      *
                               4258 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               4259 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               4260 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               4261 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               4262 *        $B$EQU - COMPILER FIXED EQUATES.                             *
                               4263 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               4264 *                                                                     *
                               4265 *   OTHER                                                             *
                               4266 *      BPREAD IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               4267 ***********************************************************************

                               4269 *
                               4270 * ENTER BPREAD - 'READ' STATEMENT ROUTINE
                               4271 *
                          0BCF 4272 BPREAD EQU   *                         BPREAD ENTRY POINT
                               4273 *
                               4274 * SET INPUT PARAMETER TO SKIP TO 'D' IN KEYWORD 'READ'
                               4275 *
    0BCF 3C 03 0873            4276 BPR010 MVI   B$NUMC,B@LREA-1           SKIP TO 'D' IN 'READ'
    0BD3 C0 87 0867            4277        B     B$GETC                    LINK TO ADVANCE POINTER
                               4278 *
                               4279 * ADVANCE POINTER TO GET NEXT CHARACTER
                               4280 *
    0BD7 C0 87 0867            4281 BPR020 B     B$GETC                    LINK TO GET NEXT CHARACTER
                               4282 *
                               4283 * CALL LIST ROUTINE TO PROCESS CURRENT LIST ELEMENT
                               4284 *
    0BDB C0 87 1853            4285 BPR030 B     B$LIST                    LINK TO PROCESS LIST ELEMENT
                               4286 *
                               4287 * GENERATE A GET INSTRUCTION PMC IN VIRTUAL MEMORY KHICH REFERENCES
                               4288 * THE VIRTUAL ENTRY ADDRESS OF THE RUN-TIME READ ROUTINE
                               4289 *
    0BDF D2 02 FC              4290 BPR040 LA    BPRGTC(,@BR),@XR          LOAD CADDR OF 'GET' INSTR
    0BE2 34 02 0A40            4291        ST    B$PCAD,@XR                SET PUT RTN VADDR FOR 'GET'
    0BE6 3C 02 0A41            4292        MVI   B$PNBY,B@LGET-1           SET PUT RTN LNG FOR 'GET'
    0BEA C0 87 093A            4293        B     B$PUTC                    LINK TO GENERATE 'GET' PNC
```

```
                               4294 *
                               4295 * TEST FOR STATEMENT TERMINATOR
                               4296 *
    0BEE 35 02 0878            4297 BPR050 L     B$GPTR,@XR                RESTORE TEXT POINTER
    0BF2 BD 1E 00              4298        CLI   B@CHAR(,@XR),B@EOST       IF ANOTHER LIST ELEMENT FOLLOWS
    0BF5 D0 01 D7              4299        BNE   BPR020(,@BR)              * GO PROCESS NEXT ELEMENT
                               4300 *
                               4301 * RETURN CONTROL TO COMPILER DISTRIBUTOR
                               4302 *
    0BF8 C0 87 0700            4303 BPR060 B     B$DIST                    RETURN TO DISTRIBUTOR

                               4305 ***********************************************************************
                               4306 * 'READ' STATEMENT ROUTINE STORAGE AND PARAMETER AREAS
                               4307 ***********************************************************************
                               4308 *
    0BFC 52               0BFC 4309 BPRGTC DC    AL(B@LCOP)(B@CGET)        'GET' OPCODE
    0BFD 3300             0BFE 4310 BPRGTO DC    AL(B@LCVA)(V$XSRD)        'GET' OPERAND

                               4312 ***********************************************************************
                               4313 *
                               4314 * END OF 'READ' STATEMENT ROUTINE CODING
                               4315 *
                               4317 ***********************************************************************
                               4318 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               4319 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               4320 *                                                                     *
                               4321 ***********************************************************************
                               4322 *STATUS                                                               *
                               4323 *   VERSION 1 MODIFICATION 0                                          *
                               4324 *                                                                     *
                               4325 *FUNCTION                                                             *
                               4326 *   BSTRLT IS EXECUTED TO TRANSLATE LET STATEMENTS WITH SUB-STRING    *
                               4327 *   OPERANDS AS THEY OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE    *
                               4328 *   PSEUDO INSTRUCTION SEQUENCE AND TO PLACE THE PSEUDO INSTRUCTION   *
                               4329 *   SEQUENCE IN VIRTUAL MEMORY.                                       *
                               4330 *                                                                     *
                               4331 *ENTRY POINTS                                                         *
                               4332 *   BSTRLT HAS TWO ENTRY POINTS:                                      *
                               4333 *        BSTRLT - TRANSLATE LET STATEMENTS                            *
                               4334 *        BSTRAS - TRANSLATE ASSIGNMENT STMT (KEYWORD-LET MISSING)     *
                               4335 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               4336 *        B     BSTRLT                                                 *
                               4337 *        B     BSTRAS                                                 *
                               4338 *                                                                     *
                               4339 *INPUT                                                                *
                               4340 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               4341 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               4342 *     KEYWORD LET, OR THE FIRST CHARACTER IN THE ASSIGNMENT LIST      *
                               4343 *     IF THE KEYWORD, LET, IS MISSING.                                *
                               4344 *   * A TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE     *
                               4345 *     FIRST CHARACTER IN THE LEADING KEYWORD, LET, OR THE FIRST       *
                               4346 *     CHARACTER IN THE ASSIGNMENT LIST IF THE KEYWORD, LET, IS        *
                               4347 *     MISSING.                                                        *
                               4348 *                                                                     *
                               4349 *OUTPUT                                                               *
                               4350 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               4351 *     GENERATED BY BSTRLT IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               4352 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               4353 *     SEQUENCES.                                                      *
                               4354 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               4355 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               4356 *                                                                     *
                               4357 *EXTERNAL REFERENCES                                                  *
                               4358 *   * B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.   *
                               4359 *   * B$PUTC - (B$PCAD, B$PNBY, B$PVAD) - ENTRY TO COMPILER VIRTUAL   *
                               4360 *              MEMORY OUTPUT ROUTINE.                                 *
                               4361 *   * B$LIST - (B$LSTR, B$LVSV, B$LRTN, B$LBAS) - ENTRY TO BASIC      *
                               4362 *              COMPILER LIST ADDRESS ROUTINE.                         *
                               4363 *   * B$SCAN - ENTRY TO COMPILER ARITHMETIC EXPRESSION SCAN ROUTINE.  *
                               4364 *   * BSCSCN - (BSCSTP, B$CRAD, B$CDAS, B$CRBS) - COMPILER CHARACTER  *
                               4365 *              EXPRESSION SCAN ROUTINE.                               *
                               4366 *   * B$BTAB - (B$BRVA, B$BRLN) - ENTRY TO BASIC COMPILER BRANCH      *
                               4367 *              TABLE ROUTINE.                                         *
                               4368 *   * BSDIST - (B$DST2) - ENTRY TO BASIC COMPILER DISTRIBUTOR ROUTINE.*
                               4369 *   * BSCOMN - (B$PRM1, B$RTRN, B$BROP, B$CADR) - COMPILER CORE       *
                               4370 *              RESIDENT COMMON SECTION                                *
                               4371 *   * B$SYMB - (B$CRSW, B$BCKT) - COMPILER SYMBOL TRANSLATION ROUTINE.*
                               4372 *                                                                     *
                               4373 *EXITS, NORMAL                                                        *
                               4374 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR                      *
                               4375 *                                                                     *
                               4376 *EXITS, ERROR                                                         *
                               4377 *   N/A                                                               *
                               4378 *                                                                     *
                               4379 *TABLES/WORK AREAS                                                    *
                               4380 *   N/A                                                               *
                               4381 *                                                                     *
                               4382 *ATTRIBUTES                                                           *
                               4383 *   BSTRLT IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               4384 *                                                                     *
                               4385 *CHARACTER CODE DEPENDENCY                                            *
                               4386 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               4387 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               4388 *                                                                     *
                               4389 *NOTES                                                                *
                               4390 *   ERROR PROCEDURES                                                  *
                               4391 *      N/A                                                            *
                               4392 *                                                                     *
                               4393 *   REGISTER USAGE                                                    *
                               4394 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               4395 *                                                                     *
                               4396 *   SAVED/RESTORED AREAS                                              *
                               4397 *      N/A                                                            *
                               4398 *                                                                     *
                               4399 *   MODIFICATION CONSIDERATIONS                                       *
                               4400 *      BSTRLT IS DIVIDED INTO THREE SECTIONS.  OCCUPYING THREE        *
                               4401 *      SECTORS.  ANY MODIFICATIONS MUST MAINTAIN LINKAGE BETWEEN      *
                               4402 *      THE THREE SECTORS AND REALIZE THE LIMITATION OF THE SECTOR     *
                               4403 *      BOUNDARY ON THE SIZE OF EACH SECTION.                          *
                               4404 *                                                                     *
                               4405 *   REQUIRE MODULES                                                   *
                               4406 *        @SYSEQ - COMMON SYSTEM EQUATES.                              *
                               4407 *        @FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUE EQUATES. *
                               4408 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.      *
                               4409 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               4410 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               4411 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               4412 *        $B$EQU - COMPILER FIXED ADDRESS EQUATES.                     *
                               4413 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               4414 *                                                                     *
                               4415 *   OTHER                                                             *
                               4416 *      BSTRLT IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               4417 ***********************************************************************

    0C00                       4419        ORG   *,256,0                    PLACE MODULE AT PAGE BOUNDARY
                          0C00 4420        USING *,@BR                      ESTABLISH BASE ADDRESSING
                               4421 ***********************************************************************
                               4422 *            FIRST DETERMINE IF THIS SEGMENT HAS BEEN ACCESSED        *
                               4423 *            PREVIOUSLY IN THE PROCESSING OF THIS STATEMENT.          *
                               4424 ***********************************************************************
                          0C00 4425 BSTRLT EQU   *                          LET ENTRY POINT ADDRESS
    0C00 74 01 F6              4426        ST    CNTCA2(,@BR),@BR           SAVE THE CADDR OF THIS SECTION
    0C03 3D 00 1AF5            4427        CLI   B$RTRN,@ZERO               IF THIS FIELD IS ZERO WE ARE
    0C07 F2 81 09              4428        JE    BST020                     * ENTERING FOR THE ?INST TIME
    0C0A 4C 01 12 1AF5         4429        MVC   BST010+@OP1(@CADDR,@BR),B$RTRN  ELSE BRANCH TO THE SAVED
    0C0F C0 87 0000            4430 BST010 B     *-*                       * RETURN ADDRESS
                               4431 ***********************************************************************
                               4432 *            LET ENTRY POINT (KEYWORD, LET, IS PRESENT).  THIS ENTRY  *
                               4433 *            POINT WILL ADVANCE THE TEXT CHARACTER POINTER TO THE     *
                               4434 *            'T' IN THE KEYWORD LET.                                  *
                               4435 ***********************************************************************
    0C13 3C 03 0873            4436 BST020 MVI   B$NUMC,B@LLET             SET GET ROUTINE TO SKIP KEYWORD
    0C17 C0 87 0867            4437        B     B$GETC                    ADVANCE TEXT CHARACTER POINTER
                               4438 ***********************************************************************
                               4439 *            ASSIGNMENT ENTRY POINT (KEYWORD, LET, IS MISSING).  THIS *
                               4440 *            ENTRY POINT WILL ADVANCE THE TEXT CHARACTER POINTER TO   *
                               4441 *            THE LEADING CHARACTER OF THE FIRST ASSIGNMENT LIST       *
                               4442 *            ELEMENT.                                                 *
                               4443 ***********************************************************************
                          0C1B 4444 BSTRAS EQU   *                         ASSIGNMENT ENTRY POINT ADDRESS
    0C1B 74 01 F6              4445        ST    CNTCA2(,@BR),@BR          SAVE THE CADDR OF THIS SECTION
                               4446 ***********************************************************************
                               4447 *            THE TEXT CHARACTER POINTER IS POSITIONED. NOW INITIALIZE *
                               4448 *            ALL SWITCHES AND GENERATE A BRANCH INSTRUCTION IMAGE SO  *
                               4449 *            THAT AT EXECUTION TIME THE RIGHT SIDE OF THE EQUAL SIGN  *
                               4450 *            HILL BE PROCESSED FIRST AND THE RESULT SAVED IN THE      *
                               4451 *            TEMPORARY VARIABLE, ECWRK.                               *
                               4452 ***********************************************************************
    0C1E D2 02 E9              4453 BST080 LA    CNTBRA(,@BR),@XR          LOAD CADDR OF BRANCH INSTR
    0C21 D0 87 CF              4454        B     BST150(,@BR)              GO GENERATE BRANCH INSTR IMAGE
    0C24 0C 01 1AF7 0A43       4455        MVC   B$BROP(@VADDR),B$PVAD     SAVE RETURN ADDR FOR RTRN BRNCH
    0C2A 35 02 0878            4456        L     B$GPTR,@XR                LOAD THE TEXT CHARACTER POINTER
                               4457 ***********************************************************************
                               4458 *            INITIALIZE MODULE SWITCHES AND BEGIN PROCESSING          *
                               4459 *            ASSIGNMENT LIST ELEMENTS IN SEQUENCE.                    *
                               4460 ***********************************************************************
    0C2E 3C 01 1BAC            4461 BST100 MVI   B$SSTA,@B1                ENABLE BDSYMB DETECTION OF 'STR'
    0C32 C0 87 0DBC            4462        B     B$SYMB                    TRANSLATE CURRENTLY REED SYMBOL
    0C36 3C 00 159E            4463        MVI   B$KWSW,@ZERO              TURN OFF KEYWCOK SWITCH
                               4464 ***********************************************************************
                               4465 *            IF SYMBOL JUST TRANSLATED WAS A CHARACTER REFERENCE.     *
                               4466 *            THE SWITCH, BSCRSW, WILL BE ON AND THE VADDR OF THE      *
                               4467 *            REFERENCE WILL BE AT BSBCKT.  THE TEXT CHARACTER POINTER *
                               4468 *            REFERENCES THE CHARACTER FOLLOWING THE CHARACTER         *
                               4469 *            REFERENCE(THE OPENING PARENTHESIS OF AN ARRAY REFERENCE).*
                               4470 *            IF THE SYMBOL WAS A STRING REFERENCE, THE TEXT CHARACTER *
                               4471 *            POINTER REFERENCES THE 'T' IN STR.                       *
                               4472 ***********************************************************************
    0C3A 3D 00 0E42            4473 BST120 CLI   B$CRSW,@ZERO              IF THE SYMBOL WAS A CHAR REF
    0C3E D0 01 4B              4474        BNE   BST130(,@BR)              * GO ACCESS CHAR PROCESSOR SEG
                               4475 ***********************************************************************
                               4476 *            THE SYMBOL JUST PROCESSED WAS A STRING FUNCTION          *
                               4477 *            SET UP TO ACCESS STR PROCESSOR SEGMENT                   *
                               4478 ***********************************************************************
    0C41 7C 14 F7              4479        MVI   CNTSAD(,@BR),CNTSTR       SET DISK ADDR PARM FOR STR PROC
    0C44 5C 01 F9 F6           4480        MVC   CNTWRK(@CADDR,@BR),CNTCA2(,@BR)  SET UP CORE RES TEST
    0C48 F2 87 25              4481        J     BST132                    GO TO ACCESSING ROUTINE
                               4482 ***********************************************************************
                               4483 *            THE SYMBOL JUST PROCESSED WAS A CHARACTER REFERENCE.     *
                               4484 *            SET UP TO ACCESS CHAR PROCESSOR SEGMENT.                 *
                               4485 ***********************************************************************
    0C4B D2 02 62              4486 BST130 LA    BST131(,@BR),@XR          LOAD RETURN ADDR
    0C4E 34 02 18EB            4487        ST    B$LRTN,@XR                SAVE RETURN ADDRESS IN BLISTA
    0C52 34 01 18E7            4488        ST    B$LBSV,@BR                SAVE BASE REG IN BLISTA
    0C56 C2 01 185E            4489        LA    B$LBAS,@BR                LOAD BLISTA BASE ADDRESS
    0C5A 35 02 0878            4490        L     B$GPTR,@XR                LOAD TEXT CHARACTER POINTER
    0C5E C0 87 1862            4491        B     B$LSTR                    GO GENERATE CHAR ADDR STACK PMC
                               4492 ***********************************************************************
                               4493 *            COMPLETE CHARACTER REFERENCE PROCESSING BY STACKING      *
                               4494 *            THE CONTENT OF &CWRK.                                    *
                               4495 ***********************************************************************
    0C62 D2 02 EE              4496 BST131 LA    CNTCWR(,@BR),@XR          LOAD CADDR OF 'STC' ECWRK INSTR
    0C65 4C 00 EF 159F         4497        MVC   CNTCWR+@B1(,@BR),B$WORK-@B1(@B1)  SET VADDR OF &CWRK
    0C6A D0 87 CF              4498        B     BST150(,@BR)              GO GENERATE PMC
    0C6D F2 87 36              4499        J     BST140                    GO CHECK NEXT LIST ELEMENT
```

```
                               4501 ***********************************************************************
                               4502 *            NEXT REQUIRED SEGMENT ACCESSING SECTION.                 *
                               4503 ***********************************************************************
    0C70 1C 01 1AF9 F6         4504 BST132 MVC   B$CADR(@CADDR),CNTCA2(,@BR)  SAVE CADDR OF CNTRL SECTION
    0C75 D2 02 A6              4505        LA    BST140(,@BR),@XR          SAVE THE RETURN ADDRESS FOR
    0C78 34 02 1AF5            4506        ST    B$RTRN,@XR                * RE-ENTERING THE CNTRL SECTION
    0C7C 5D 01 F6 F4           4507 BST134 CLC   CNTCA2(@CADDR,@BR),CNTPBA(,@BR)  IF CURR SEG CAME FR DISK
    0C80 F2 81 0F              4508        JE    BST136                    * GO LOAD & EXEC SEG FR DISK
                               4509 ***********************************************************************
                               4510 *            CONTROL SECTION WAS CORE RESIDENT - TEST WHETHER THE     *
                               4511 *            REQUIRED SECTION IS ALSO CORE RESIDENT.                  *
                               4512 ***********************************************************************
    0C83 7C 1F FC              4513        MVI   CNTFCP-@B1(,@BR),CNTFPE   SET FINAL CORE PAGE
    0C86 4E 00 FC 043B         4514        ALC   CNTFCP-1(,@BR),$EXFTR(@B1)  CALC MAX PROCESSOR CORE PAGE
    0C8B 5D 01 F9 FD           4515        CLC   CNTWRK(,@BR),CNTFCP(@CADDR,@BR)  IF NEXT SEGMENT IN CORE
    0C8F F2 82 0B              4516        JL    BST138                    * GO SET TO EXEC NEXT SEGMENT
                               4517 ***********************************************************************
                               4518 *            REQUIRED SECTION IS DISK RESIDENT - ESTABLISH            *
                               4519 *            DISTRIBUTOR PARAMETERS FOR CORELOADING AND EXECUTING     *
                               4520 *            THE REQUIRED SECTION.                                    *
                               4521 ***********************************************************************
    0C92 5C 01 F6 F4           4522 BST136 MVC   CNTCA2(,@BR),CNTPBA(@CADDR,@BR)  SET UP DISKLOAD CADDR
    0C96 D2 02 F5              4523        LA    CNTAD2(,@BR),@XR          LOAD DIST PARAMETERS CADDR
    0C99 C0 87 073A            4524        B     B$DST2                    GO LOAD & EXEC NEXT SEGMENT
                               4525 ***********************************************************************
                               4526 *            REQUIRED SEGMENT IS CORE RESIDENT - BRANCH TO THE        *
                               4527 *            REQUIRED SEGMENT'S ENTRY POINT.                          *
                               4528 ***********************************************************************
    0C9D 75 01 F9              4529 BST138 L     CNTWRK(,@BR),@BR          LOAD THE BASE ADDRESS FOR
    0CA0 76 01 F2              4530        A     CNTBLS(,@BR),@BR          * NEXT SEGMENT
    0CA3 D0 87 00              4531        B     CNTENT(,@BR)              GO EXECUTE NEXT SEGMENT
```

```
                               4533 ***********************************************************************
                               4534 *            LIST ELEMENT HAS BEEN PROCESSED, NOW CHECK TO SEE        *
                               4535 *            IF THE ENTIRE LIST HAS BEEN PROCESSED.  IF NOT GO GET    *
                               4536 *            THE NEXT LIST ELEMENT, IF IT HAS BEEN, GO PROCESS        *
                               4537 *            THE RIGHT SIDE.                                          *
                               4538 ***********************************************************************
    0CA6 D2 02 EC              4539 BST140 LA    CNTUSC(,@BR),@XR          LOAD CADDR OF 'USC' INSTRUCTION
    0CA9 7C 01 D7              4540        MVI   BST160+@Q(,@BR),B@LUSC-1  SET LNGTH PARM FOR PUT RTN
    0CAC D0 87 CF              4541        B     BST150(,@BR)              GO GENERATE PMC
    0CAF BD 7E 00              4542        CLI   B@CHAR(,@XR),B@EQUL       IF THE NEXT CHARACTER IS AN'+.
    0CB2 F2 81 07              4543        JE    BST145                    * THEN GO ACCESS TERM SECTION
    0CB5 C0 87 0867            4544        B     B$GETC                    ELSE ADVANCE TEXT POINTER AND
    0CB9 D0 87 2E              4545        B     BST100(,@BR)              * PROCESS NEXT LIST ELEMENT
                               4546 ***********************************************************************
                               4547 *            THE ENTIRE ASSIGNMENT LIST HAS BEEN PROCESSED, NOW       *
                               4548 *            SET UP TO ACCESS THE TERMINATION SECTION                 *
                               4549 ***********************************************************************
    0CBC 7C 18 F7              4550 BST145 MVI   CNTSAD(,@BR),CNTTRM       SET DISK ADDR PARR FOR TAM SCTN
    0CBF 5C 01 F9 F6           4551        MVC   CNTWRK(@CADDR,@BR),CNTCA2(,@BR)  SET UP CORE RES TEST
    0CC3 5E 01 F9 FB           4552        ALC   CNTWRK(@CADDR,@BR),CNTBL1(,@BR)  INCREMENT TO CADDR-1 PAGE
    0CC7 1C 01 1AF5 EB         4553        MVC   B$RTRN(@CADDR),CNTBOP(,@BR)  CLEAR RETURN ADDRESS
    0CCC D0 87 7C              4554        B     BST134(,@BR)              GO ACCRDS TERMINATION SECTION
                               4555 ***********************************************************************
                               4556 *            THIS SUBROUTINE WILL GENERATE, IN VIRTUAL MEMORY,        *
                               4557 *            THE PSEUDO INSTRUCTION POINTED TO BY @XR.                *
                               4558 *            THE INPUT PARAMETERS ARE AS FOLLOWS:                     *
                               4559 *                  1. XR REFERENCES THE INSTRUCTION TO BE             *
                               4560 *                     GENERATED.                                      *
                               4561 *                  2. IF THE LENGTH OF THE INSTRUCTION IS NOT         *
                               4562 *                     THREE, THE LENGTH MUST BE STORED IN A           *
                               4563 *                     MVI INSTRUCTION (BST160+@Q).                    *
                               4564 ***********************************************************************
    0CCF 74 08 E8              4565 BST150 ST    BST170+@OP1(,@BR),@ARR    SAVE THE RETURN ADDRESS
    0CD2 34 02 0A40            4566        ST    B$PCAD,@XR                SET CADDR PARM FOR THE PUT RTN
    0CD6 3C 02 0A41            4567 BST160 MVI   B$PNBY,B@LLET-1           SET LENGTH FARAH FOR THE PUT RTN
    0CDA C0 87 093A            4568        B     B$PUTC                    GENERATE PMC IN VIRTUAL MEMORY
    0CDE 7C 02 D7              4569        MVI   BST160+@Q(,@BR),B@LLET-1  MAKE SUBROUTINE REUSABLE
    0CE1 35 02 0878            4570        L     B$GPTR,@XR                LOAD THE TEXT CHARACTER POINTER
    0CE5 C0 87 0000            4571 BST170 B     *-*                       RETURN TO CALLING SECTION
```

```
                               4573 ***********************************************************************
                               4574 *            SUBSTRING ASSIGNMENT - CONTROL SECTION CONSTANTS          *
                               4575 *            AND WORKAREAS.                                            *
                               4576 ***********************************************************************
    0CE9 46               0CE9 4577 CNTBRA DC    AL(B@LCOP)(B@CBRA)        BRANCH OPCODE
    0CEA 0000             0CEB 4578 CNTBOP DC    AL(@VADDR)(@ZERO)         BRANCH OPERAND
                               4579 *
    0CEC 2C               0CEC 4580 CNTUSC DC    AL(B@LCOP)(B@CUSC)        UNSTACK CHAR OPCODE
    0CED 01               0CED 4581        DC    XL1'01'                   UNSTACK CHAR OPERAND
    0CEE 28               0CEE 4582 CNTCWR DC    AL(B@LCOP)(B@CSTC)        STACK CHAR OPCODE
    0CEF F500             0CF0 4583        DC    AL2(B$CWRK)               STACK CHAR OPERAND
                          0004 4584 CNTPSI EQU   X'04'                     PHYSICAL SECTOR INCREMENT
                          0000 4585 CNTENT EQU   0                         DISP TO ENTRY PTS OF OTHER SCTNS
                          0014 4586 CNTSTR EQU   B@DSML+CNTPSI             STR PROC SECTION-PHYS SCTR ADDR
                          0018 4587 CNTTRM EQU   CNTSTR+CNTPSI             TERM SECTION-PHYS SCTR ADDR
                               4588 *
    0CF1 0100             0CF2 4589 CNTBLS DC    AL(@CADDR)(B@BLSZ)        LENGTH OF CORE PAGE
    0CF3 0600             0CF4 4590 CNTPBA DC    AL(@CADDR)(B$CSBF)        PROCESSOR DISK BUFFER CORE ADDR
                               4591 *
                               4592 *
                          0CF5 4593 CNTAD2 EQU   *                         DIST PARMS FOR EXEC NEXT SECTION
    0CF5                  0CF6 4594 CNTCA2 DS    CL(@CADDR)                CONTROL SECTION CORE ADDRESS
    0CF7                  0CF7 4595 CNTSAD DS    CL1                       PHYSICAL SECTOR ADDRESS
    0CF8                  0CF9 4596 CNTWRK DS    CL2                       CONTROL SECTION WORKAREA
    0CFA 0200             0CFB 4597 CNTBL1 DC    AL(@CADDR)(2*B@BLSZ)      LENGTH OF 2 CORE PAGES
    0CFC 0000             0CFD 4598 CNTFCP DC    AL(@CADDR)(@ZERO)         FINAL AVAILABLE CORE PAGE ADDR
                          001F 4599 CNTFPE EQU   X'1F'                     FINAL PAGE BEFORE EXTENSION
                               4600 ***********************************************************************
                               4601 *            END OF LET-CONTROL SECTION                                *
                               4602 ***********************************************************************
    0D00                       4604        ORG   BSTRLT+B@BLSZ             PLACE SEGMENT AT PAGE BOUNDARY
                          0D00 4605        USING *,@BR                     ESTABLISH BASE ADDRESS
                               4606 ***********************************************************************
                               4607 *            SYMBOL JUST TRANSLATED WAS A STRING FUNCTION - ADVANCE   *
                               4608 *            TEXT CHARACTER POINTER TO OPENING CHARACTER OF CHAR      *
                               4609 *            REFERENCE WITHIN THE STRING FUNCTION.                    *
                               4610 ***********************************************************************
    0D00 3C 03 0873            4611 BST200 MVI   B$NUMC,B@LLET             SKIP TO LEADING CHAR IN STRING
    0D04 C0 87 0867            4612        B     B$GETC                    * FUNCTION CHARACTER REFERENCE
    0D08 C0 87 0DBC            4613        B     B$SYMB                    TRANSLATE THE CHAR REFERENCE
                               4614 ***********************************************************************
                               4615 *            THE VADDR OF THE TRANSLATED CHARACTER REFERENCE IS       *
                               4616 *            AT B$BCKT.                                               *
                               4617 ***********************************************************************
    0D0C 4C 01 DF 1590         4618        MVC   STRAOP(@VADDR,@BR),B$BCKT  SAVE VADDR IN 'STA' OPERAND
    0D11 BD 4D 00              4619        CLI   B@CHAR(,@XR),B@LPAR       IF CHAR REF IS AN ARRAY REF
    0D14 D0 81 55              4620        BE    BST240(,@BR)              * GO PROCESS ARRAY REFERENCE
                               4621 ***********************************************************************
                               4622 *            STRING FUNCTION CHARACTER REFERENCE IS A CHARACTER       *
                               4623 *            VARIABLE.                                                *
                               4624 ***********************************************************************
    0D17 D2 02 DD              4625        LA    STRSTA(,@BR),@XR          LOAD CADDR OF STA INSTRUCTION
    0D1A D0 87 83              4626        B     BST300(,@BR)              GO GENERATE PMC
    0D1D 5C 01 E2 DF           4627        MVC   STRCOP(@VADDR,@BR),STRAOP(,@BR)  SET VADDR OPERND OF 'STC'
    0D21 D2 02 E0              4628        LA    STRSTC(,@BR),@XR          LOAD CADDR OF 'STC' INSTRUCTION
    0D24 D0 87 83              4629        B     BST300(,@BR)              GO GENERATE PMC
    0D27 C0 87 1514            4630 BST210 B     B$SCAN                    PROCESS 1ST 'STR' ARITH OPERAND
    0D2B BD 5D 00              4631        CLI   B@CHAR(,@XR),B@RPAR       IF LENGTH PARM IS NOT PRESENT
    0D2E D0 81 38              4632        BE    BST220(,@BR)              * GO GENERATE 'STX' INSTRUCTION
    0D31 C0 87 1514            4633        B     B$SCAN                    ELSE PROCESS LENGTH PARAMETER
    0D35 D0 87 41              4634        B     BST230(,@BR)              GO COMPLETE 'STR' PROCESSING
    0D38 D2 02 E3              4635 BST220 LA    STRSTX(,@BR),@XR          LOAD CADDR OF 'STX' INSTRUCTION
    0D3B 7C 01 8B              4636        MVI   BST310+@Q(,@BR),B@LSTX-1  SET LENGTH PARM FOR PUT ROUTINE
    0D3E D0 87 83              4637        B     BST300(,@BR)              GO GENERATE PMC
                               4638 ***********************************************************************
                               4639 *            STRING FUNCTION IS PROCESSED.  NOW GENERATE CHARACTER    *
                               4640 *            STACKING FOR ECWRK AND FUNCTION CALL THEN RETURN TO      *
                               4641 *            PROCESS NEXT ASSIGNMENT LIST ELEMENT.                    *
                               4642 ***********************************************************************
    0D41 D2 02 E5              4643 BST230 LA    STRCWR(,@BR),@XR          LOAD CADDR OF 'STC' ECWRK INSTR
    0D44 4C 00 E6 159F         4644        MVC   STRWOP-@B1(,@BR),B$WORK-@B1(@B1)  SET VADDR OF &CWRK
    0D49 D0 87 83              4645        B     BST300(,@BR)              GO GENERATE PMC
    0D4C D2 02 E8              4646        LA    STRFN2(,@BR),@XR          LOAD CADDR OF 'FNO' #2 INSTR
    0D4F D0 87 83              4647        B     BST300(,@BR)              GO GENERATE PMC
    0D52 D0 87 5F              4648        B     BST250(,@BR)              RETURN TO PROCESS NEXT LIST ELMT
                               4649 ***********************************************************************
                               4650 *            STRING FUNCTION CHARACTER REFERENCE IS A CHARACTER       *
                               4651 *            ARRAY REFERENCE.                                         *
                               4652 ***********************************************************************
    0D55 D0 87 99              4653 BST240 B     BST340(,@BR)              GO PROCESS STR CHAR ARRAY REF
    0D58 C0 87 0867            4654        B     B$GETC                    LINK TO ADVANCE TEXT CHAR PNTR
    0D5C D0 87 27              4655        B     BST210(,@BR)              GO PROCESS 'STR' ARITH OPERANDS
                               4656 ***********************************************************************
                               4657 *            WHEN THE STRING OPERAND HAS BEEN PROCESSED.              *
                               4658 *            THIS SECTION WILL RETURN TO THE CONTROL SECTION TO       *
                               4659 *            CONTINUE PROCESSING THE ASSIGNMENT LIST ELEMENTS.        *
                               4660 ***********************************************************************
    0D5F 1D 01 1AF9 F9         4661 BST250 CLC   B$CADR(@CADDR),STRPBA(,@BR)  IF CONTROL SECTION CAME FROM
    0D64 F2 81 0D              4662        JE    BST270                    * DISK-GO LD & EXEC CNTL SECTION
                               4663 ***********************************************************************
                               4664 *            CONTROL SECTION IS CORE RESIDENT - LOAD BASE REGISTER   *
                               4665 *            AND RETURN.                                              *
                               4666 ***********************************************************************
    0D67 4C 01 73 1AF5         4667        MVC   BST260+@OP1(@CADDR,@BR),B$RTRN  SET UP RETURN BRANCH ADDR
    0D6C 35 01 1AF9            4668        L     B$CADR,@BR                LOAD CONTROL SECTION BASE ADDR
    0D70 C0 87 0000            4669 BST260 B     *-*                       RETURN TO CONTROL SECTION

                               4671 ***********************************************************************
                               4672 *            CONTROL SECTION IS DISK RESIDENT - SET DISTRIBUTOR       *
                               4673 *            PARAMETERS TO LOAD AND EXECUTE CONTROL SECTION.          *
                               4674 ***********************************************************************
    0D74 5C 01 F6 F9           4675 BST270 MVC   STRCA2(@CADDR,@BR),STRPBA(,@BR)  SET UP DISKLOAD CADDR
    0D78 C0 87 0867            4676        B     B$GETC                    ADVANCE THE TEXT CHAR POINTER
    0D7C D2 02 F5              4677        LA    STRAD2(,@BR),@XR          LOAD DIST PARAMETERS CADDR
    0D7F C0 87 073A            4678        B     B$DST2                    GO LOAD & EXEC CONTROL SECTION
```

```
                               4680 ***********************************************************************
                               4681 *            THIS SUBROUTINE WILL GENERATE, IN VIRTUAL MEMORY,        *
                               4682 *            THE PSEUDO INSTRUCTION POINTED TO BY @XR.                *
                               4683 *            THE INPUT PARAMETERS ARE AS FOLLOWS:                     *
                               4684 *                  1. XR REFERENCES THE INSTRUCTION TO BE             *
                               4685 *                     GENERATED.                                      *
                               4686 *                  2. IF THE LENGTH OF THE INSTRUCTION IS NOT         *
                               4687 *                     THREE. THE LENGTH MUST BE STORED IN             *
                               4688 *                     MVI INSTRUCTION (BST310+@Q).                    *
                               4689 ***********************************************************************
    0D83 74 08 98              4690 BST300 ST    BST320+@OP1(,@BR),@ARR    SAVE THE RETURN ADDRESS
    0D86 34 02 0A40            4691        ST    B$PCAD,@XR                SET CADDR PARM FOR THE PUT RTN
    0D8A 3C 02 0A41            4692 BST310 MVI   B$PNBY,B@LLET-1           SET LENGTH PARM FOR THE PUT RTN
    0D8E C0 87 093A            4693        B     B$PUTC                    GENERATE PMC IN VIRTUAL MEMORY
    0D92 7C 02 8B              4694        MVI   BST310+@Q(,@BR),B@LLET-1  MAKE THE SUBROUTINE REUSABLE
    0D95 C0 87 0000            4695 BST320 B     *-*                       RETURN TO CALLING SECTION

                               4697 ***********************************************************************
                               4698 *            THIS SUBROUTINE WILL GENERATE PSEUDO INSTRUCTIONS        *
                               4699 *            TO PROCESS A CHARACTER ARRAY REFERENCE.  THE INPUT       *
                               4700 *            PARAMETERS ARE AS FOLLOWS:                               *
                               4701 *                  1. THE VIRTUAL ADDRESS OF THE ARRAY DESCRIPTOR     *
                               4702 *                     IS AT BSBCKT.                                   *
                               4703 *                  2. THE TEXT CHARACTER POINTER REFERENCES THE       *
                               4704 *                     OPENING PARERTHESIS OF THE ARRAY INDEX.         *
                               4705 ***********************************************************************
    0D99 74 08 DC              4706 BST340 ST    BST360+@OP1(,@BR),@ARR    SAVE THE RETURN ADDRESS
    0D9C 4C 01 ED 1590         4707        MVC   STR1OP(@VADDR,@BR),B$BCKT  SAVE VADDR OF ARRAY DESCRIPTOR
    0DA1 4C 01 DF 15A0         4708        MVC   STRAOP(@VADDR,@BR),B$WORK  SET VADDR OF @WRK IN 'STA' PMC
    0DA6 D2 02 DD              4709        LA    STRSTA(,@BR),@XR          LOAD CADDR OF 'STA' INSTR
    0DA9 D0 87 83              4710        B     BST300(,@BR)              GO GENERATE 'STA' PMC
    0DAC C0 87 1514            4711        B     B$SCAN                    GO PROCESS ARRAY INDEX
    0DB0 7C 00 8B              4712        MVI   BST310+@Q(,@BR),B@LUSF-1  SET LENGTH PARM OF PUT ROUTINE
    0DB3 D2 02 F4              4713        LA    STRUSF(,@BR),@XR          LOAD CADDR OF 'USF' INSTR
    0DB6 D0 87 83              4714        B     BST300(,@BR)              GO GENERATE 'USF' INSTR
    0DB9 5C 01 F3 DF           4715        MVC   STRFOP(@VADDR,@BR),STRAOP(,@BR)  SET VADDR OPRND FOR 'STF'
    0DBD D2 02 F1              4716        LA    STRSTF(,@BR),@XR          LOAD CADDR OF 'STF' INSTR
    0DC0 D0 87 83              4717        B     BST300(,@BR)              GO GENERETE 'STF' INSTR
    0DC3 5C 01 F0 ED           4718        MVC   STRBOP(@VADDR,@BR),STR1OP(,@BR)  SET VADDR OPRND FOR 'STF'
    0DC7 D2 02 EE              4719        LA    STRSB1(,@BR),@XR          LOAD CADDR OF 'SB1' INSTR
    0DCA D0 87 83              4720        B     BST300(,@BR)              GO GENERATE 'SB1' INSTR
    0DCD D2 02 F1              4721        LA    STRSTF(,@BR),@XR          LOAD CADDR OF 'STF' INSTR
    0DD0 D0 87 83              4722        B     BST300(,@BR)              GO GENERATE 'STF &WRK' PMC
    0DD3 D2 02 EB              4723        LA    STRSC1(,@BR),@XR          LOAD CADDR OF 'SC1' INSIR
    0DD6 D0 87 83              4724        B     BST300(,@BR)              GO GENERATE 'SC1' INSTR
    0DD9 C0 87 0000            4725 BST360 B     *-*                       RETURN TO CALLING SECTION
```

```
                               4727 ***********************************************************************
                               4728 *            STRING PROCESSOR SECTION EQUANS,CONSTANTS, AND           *
                               4729 *            WORKAREAS.                                               *
                               4730 ***********************************************************************
    0DDD 34                0DDD 4731 STRSTA DC    AL(B@LCOP)(B@CSTA)        STACK ADDRESS OPCODE
    0DDE                   0DDF 4732 STRAOP DS    CL2                       STACK ADDRESS OPERAND
                               4733 *
    0DE0 28                0DE0 4734 STRSTC DC    AL(B@LCOP)(B@CSTC)        STACK CHARACTER FIELD OPCODE
    0DE1                   0DE2 4735 STRCOP DS    CL2                       STACK CHARACTER FIELD OPERAND
                               4736 *
    0DE3 3C                0DE3 4737 STRSTX DC    AL(B@LCOP)(B@CSTX)        STACK EXEC CTRL CODE OPCODE
    0DE4 FF                0DE4 4738 STRXOP DC    XL1'FF'                   STACK EXEC CTRL CODE OPERAND
                               4739 *
    0DE5 28                0DE5 4740 STRCWR DC    AL(B@LCOP)(B@CSTC)        STACK CHAR OF CWRK OPCODE
    0DE6 F500              0DE7 4741 STRWOP DC    AL2(B$CWRK)               STACK CHAR OF CWRK OPERAND
                               4742 *
    0DE8 12                0DE8 4743 STRFN2 DC    AL(B@LCOP)(B@CFN0)        FUNCT CALL-NO ARGUMENT OPCODE
    0DE9 5120              0DEA 4744        DC    AL2(V$CCON)               FUNCT CALL-NO ARGUMENT OPERAND
                               4745 *
    0DEB 2A                0DEB 4746 STRSC1 DC    AL(B@LCOP)(B@CSC1)        STACK CHAR ARRAY ELEMENT OPCODE
    0DEC                   0DED 4747 STR1OP DS    CL2                       STACK CHAR ARRAY ELEMENT OPERAND
                               4748 *
    0DEE 3A                0DEE 4749 STRSB1 DC    AL(B@LCOP)(B@CSB1)        STACK CHAR ARRAY ADDR OPCODE
    0DEF                   0DF0 4750 STRBOP DS    CL2                       STACK CHAR ARRAY ADDR OPERAND
                               4751 *
    0DF1 20                0DF1 4752 STRSTF DC    AL(B@LCOP)(B@CSTF)        STACK FLOATING PT VALUE OPCODE
    0DF2                   0DF3 4753 STRFOP DS    CL2                       STACK FLOATING PT VALUE OPERAND
                               4754 *
    0DF4 26                0DF4 4755 STRUSF DC    AL(B@LCOP)(B@CUSF)        UNSTACK FLTING PT VALUE OPCODE
                               4756 *
                          0DF5 4757 STRAD2 EQU   *                         DIST PARAMETER ADDR
    0DF5                   0DF6 4758 STRCA2 DS    CL(@CADDR)                CONTROL SECTION CORE ADDRESS
    0DF7 10                0DF7 4759        DC    AL1(B@DSML)               PHYSICAL SECTOR ADDRESS
    0DF8 0600              0DF9 4760 STRPBA DC    AL(@CADDR)(B$CSBF)        PROCESSOR DISK BUFFER CADDR
                               4761 ***********************************************************************
                               4762 *            END OF LET-STRING PROCESSOR SECTION                      *
                               4763 ***********************************************************************
    0E00                       4765        ORG   BSTRLT+2*B@BLSZ           PLACE SEGMENT AT PAGE BOUNDARY
                          0E00 4766        USING *,@BR                     ESTABLISH BASE ADDRESS
                               4767 ***********************************************************************
                               4768 *            THE ASSIGNMENT LIST HAS BEEN PROCESSED. NOW GENERATE A   *
                               4769 *            BRANCH INSTRUCTION IMAGE IN VIRTUAL MEMORY(AT EXECUTION  *
                               4770 *            TIME THIS BRANCH WILL TRANSFER CONTROL BEYOND THE SET    *
                               4771 *            UP FOR THE RIGHT SIDE TO THE NEXT SEQUENTIAL STATEMENT.  *
                               4772 ***********************************************************************
    0E00 D2 02 D8              4773 BST400 LA    TRMBIC(,@BR),@XR          LOAD CADDR OF 'BRA' INSTRUCTION
    0E03 D0 87 C2              4774        B     BST550(,@BR)              GO GENERATE PMC
                               4775 ***********************************************************************
                               4776 *            ESTABLISH CONDITIONS TO RESOLVE THE ADDRESS OPERAND      *
                               4777 *            IN THE FIRST BRANCH INSTRUCTION IMAGE (BST080)           *
                               4778 ***********************************************************************
    0E06 0C 01 19EF 1AF7       4779        MVC   B$BRVA,B$BROP(@VADDR)     SET BRANCH TABLE VADDR PARM
    0E0C 1F 01 19EF DC         4780        SLC   B$BRVA,TRMBN1(@VADDR,@BR) * FOR THE BRA IMAGE OPERAND
    0E11 0C 01 19F1 0A43       4781        MVC   B$BRLN,B$PVAD(@VADDR)     SET BRANCH TABLE LINE NO. PARM
    0E17 C0 87 1996            4782        B     B$BTAB                    LINK TO SET UP RESOLUTION
                               4783 ***********************************************************************
                               4784 *            GENERATE PSEUDO INSTRUCTIONS TO UNSTACK THE SOURCE       *
                               4785 *            CHARACTERS INTO ECWRK.  THE FIRST BRANCH INSTRUCTION     *
                               4786 *            PASSES CONTROL TO THIS INSTRUCTION SEQUENCE.             *
                               4787 ***********************************************************************
    0E1B D2 02 DD              4788        LA    TRMSTA(,@BR),@XR          LOAD CADDR OF 'STA' INSTRUCTION
    0E1E 4C 00 DE 159F         4789        MVC   TRMAOP-@B1(,@BR),B$WORK-@B1(@B1) SET VADDR OF &CWRK
    0E23 D0 87 C2              4790        B     BST550(,@BR)              GO GENERATE PMC
    0E26 C0 87 0867            4791        B     B$GETC                    ADVANCE TEXT CHARACTER POINTER
    0E2A BD 7D 00              4792        CLI   B@CHAR(,@XR),B@SQUO       IF THE OPERAND IS A LITERAL
    0E2D F2 01 0B              4793        JNE   BST410                    * BYPASS BDSYMB CALL
    0E30 3C 00 0873            4794        MVI   B$NUMC,B@GETS             DISABLE THE GET ROUTINE
    0E34 C0 87 14B0            4795        B     B$CSCN                    GO PROCESS CHAR LITERAL OPERAND
    0E38 F2 87 5C              4796        J     BST600                    CONTINUE PROCESSING
    0E3B 3C 01 1BAC            4797 BST410 MVI   B$SSTA,@B1                ENABLE BDSYMB DETECTION OF 'STR'
    0E3F C0 87 0DBC            4798        B     B$SYMB                    TRANSLATE SOURCE SYMBOL
    0E43 3C 00 159E            4799        MVI   B$KWSW,@ZERO              TURN OFF KEYWORD SWITCH
    0E47 3D 00 0E42            4800        CLI   B$CRSW,@ZERO              IF SOURCE SYMBOL IS NOT A CHAR
    0E4B D0 81 65              4801        BE    BST500(,@BR)              * REF GO SET UP 'STR' PROCESSING
                               4802 ***********************************************************************
                               4803 *            SOURCE SYMBOL IS A CHARACTER REFERENCE (ARRAY, VARIABLE, *
                               4804 *            OR CONSTANT).                                            *
                               4805 ***********************************************************************
    0E4E D2 02 97              4806 BST440 LA    BST600(,@BR),@XR          LOAD CADDR OF RETURN ADDR
    0E51 34 02 150D            4807 BST460 ST    B$CRAD,@XR                SET RETURN ADDR IN BECSCN
    0E55 34 01 1509            4808        ST    B$CRBS,@BR                SAVE BASE REG CONTENT IN BECSCN
    0E59 C2 01 14BB            4809        LA    B$CBAS,@BR                LOAD BECSCN BASE ADDRESS
    0E5D 35 02 0878            4810        L     B$GPTR,@XR                LOAD TEXT CHARACTER POINTER
    0E61 C0 87 14CC            4811        B     B$CSTR                    GO TO CHAR EXPRSSN SCAN ROUTINE
```

```
                               4813 ************************************************************************
                               4814 *            SOURCE SYMBOL IS A STRING FUNCTION                        *
                               4815 ************************************************************************
    0E65 3C 03 0873            4816 BST500 MVI   B$NUMC,B@LLET             SET GET RTN TO SKIP 'STRC'
    0E69 C0 87 0867            4817        B     B$GETC                    ADVANCE TEXT CHARACTER POINTER
    0E6D C0 87 0DBC            4818        B     B$SYMB                    TRANSLATE STRING CHARACTER REF
    0E71 D2 02 77              4819        LA    BST540(,@BR),@XR          LOAD RETURN ADDRESS
    0E74 D0 87 51              4820        B     BST460(,@BR)              GO TO CHAR EXPRSSN SCAN ROUTINE
    0E77 C0 87 1514            4821 BST540 B     B$SCAN                    PROCESS 1ST ARITH OPERAND
    0E7B BD 5D 00              4822        CLI   B@CHAR(,@XR),B@RPAR       IF NEXT OPERAND IS PRESENT THEN
    0E7E D0 01 8D              4823        BNE   BST545(,@BR)              * PROCESS IT
    0E81 D2 02 E0              4824        LA    TRMSTX(,@BR),@XR          ELSE LOAD CADDR OF STX INSTRUCTN
    0E84 7C 01 CA              4825        MVI   BST560+@Q(,@BR),B@LSTX-1  SET LENGTH PARM FOR PUT RTN
    0E87 D0 87 C2              4826        B     BST550(,@BR)              * AND GEN PMC
    0E8A D0 87 91              4827        B     BST547(,@BR)              GO FINISH STR PROCESSING
    0E8D C0 87 1514            4828 BST545 B     B$SCAN                    PROCESS LAST OPERAND
    0E91 D2 02 E5              4829 BST547 LA    TRMFN1(,@BR),@XR          LOAD CADDR OF PMC FOR 'FNO' #1
    0E94 D0 87 C2              4830        B     BST550(,@BR)              GO GENERATE PMC
    0E97 D2 02 E8              4831 BST600 LA    TRMUSC(,@BR),@XR          LOAD CADDR OF UNSTACK PMC
    0E9A 7C 01 CA              4832        MVI   BST560+@Q(,@BR),B@LUSC-1  SET LENGTH PARM FOR PUT RTN
    0E9D D0 87 C2              4833        B     BST550(,@BR)              GO GENERATE PMC

                               4835 ************************************************************************
                               4836 *            INSTRUCTIONS TO PROCESS THE SOURCE VALUE ARE COMPLETE.   *
                               4837 *            NON GENERATE THE RETURN BRANCH INSTRUCTION. THIS         *
                               4838 *            INSTRUCTION WILL TRANSFER CONTROL TO THE LIST            *
                               4839 *            ASSIGNMENT SEQUENCE AFTER THE SOURCE VALUE IS STORED     *
                               4840 *            INTO ECNRK.                                              *
                               4841 ************************************************************************
    0EA0 4C 01 E4 1AF7         4842        MVC   TRMBOP(@VADDR,@BR),B$BROP  SET VADDR OPRND OF RTRN BRANCH
    0EA5 D2 02 E2              4843        LA    TRMBRC(,@BR),@XR          LOAD CADDR OF 'BRA' INSTRUCTION
    0EA8 D0 87 C2              4844        B     BST550(,@BR)              GO GENERATE PMC
                               4845 ************************************************************************
                               4846 *            RESOLVE SECOND BRANCH INSTRUCTION IMAGE (BST500).        *
                               4847 ************************************************************************
    0EAB 0C 01 19EF 19F1       4848        MVC   B$BRVA,B$BRLN(@VADDR)     SET BRANCH TABLE VADDR PARM
    0EB1 1F 01 19EF DC         4849        SLC   B$BRVA,TRMBN1(@VADDR,@BR) * FOR 'BRA' IMAGE INSTR
    0EB6 3A 07 071D            4850        SBN   B$NXSW,B$NXMK             SET NXT STMT SWCH ON TO ESTBLSH
                               4851 *                                      * LINE NO. PARM
                               4852 ************************************************************************
                               4853 *            RETURN TO COMPILER DISTRIBUTOR                           *
                               4854 ************************************************************************
    0EBA C0 87 0867            4855        B     B$GETC                    LINK TO ADVANCE TEXT CHAR PTR
    0EBE C0 87 0700            4856        B     B$DIST
```

```
                               4858 ************************************************************************
                               4859 *            THIS SUBROUTINE WILL GENERATE, IN VIRTUAL MEMORY,         *
                               4860 *            THE PSEUDO INSTRUCTION POINTED TO BY @XR.  THE            *
                               4861 *            INPUT PARAMETERS ARE AS FOLLOWS:                          *
                               4862 *                  1. XR REFERENCES THE INSTRUCTION TO BE              *
                               4863 *                     GENERATED.                                      *
                               4864 *                  2. IF THE LENGTH OF THE INSTRUCTION IS NOT          *
                               4865 *                     THREE, THE LENGTH MUST BE STORED IN A            *
                               4866 *                     MVI INSTRUCTION (BST560+@Q).                     *
                               4867 ************************************************************************
    0EC2 74 08 D7              4868 BST550 ST    BST570+@OP1(,@BR),@ARR    SAVE THE RETURN ADDRESS
    0EC5 34 02 0A40            4869        ST    B$PCAD,@XR                SET CADDR PARM FOR THE PUT RTN
    0EC9 3C 02 0A41            4870 BST560 MVI   B$PNBY,B@LLET-1           SET LENGTH PARM FOR THE PUT RTN
    0ECD C0 87 093A            4871        B     B$PUTC                    GENERATE PMC IN VIRTUAL MEMORY
    0ED1 7C 02 CA              4872        MVI   BST560+@Q(,@BR),B@LLET-1  MAKE SUBROUTINE REUSABLE
    0ED4 C0 87 0000            4873 BST570 B     *-*                       RETURN TO CALLING SECTION

                               4875 ************************************************************************
                               4876 *            TERMINATION SECTION CONSTANTS, EQUATES AND WORKAREAS     *
                               4877 ************************************************************************
    0ED8 46               0ED8 4878 TRMBIC DC    AL(B@LCOP)(B@CBRA)        UNCONDITIONAL BRANCH OPCODE
    0ED9 0000             0EDA 4879        DC    AL(@VADDR)(@ZERO)         BRANCH IMAGE OPERAND
                               4880 *
    0EDB 0001             0EDC 4881 TRMBN1 DC    IL(@VADDR)'1'             BINARY ONE
                               4882 *
    0EDD 34               0EDD 4883 TRMSTA DC    AL(B@LCOP)(B@CSTA)        STACK ADDRESS OPCODE
    0EDE F500             0EDF 4884 TRMAOP DC    AL2(B$CWRK)               STACK ADDRESS OPERAND
                               4885 *
    0EE0 3C               0EE0 4886 TRMSTX DC    AL(B@LCOP)(B@CSTX)        STACK EXEC CTRL CODE OPCODE
    0EE1 FF               0EE1 4887        DC    XL1'FF'                   STACK EXEC CTRL CODE OPERAND
                               4888 *
    0EE2 46               0EE2 4889 TRMBRC DC    AL(B@LCOP)(B@CBRA)        UNCONDITIONAL BRANCH OPCODE
    0EE3                  0EE4 4890 TRMBOP DS    CL2                       UNCONDITIONAL BRANCH OPERAND
                               4891 *
    0EE5 12               0EE5 4892 TRMFN1 DC    AL(B@LCOP)(B@CFN0)        FUNCT CALL-NO ARGUMENT OPCODE
    0EE6 5100             0EE7 4893        DC    AL(@VADDR)(V$CSSR)        FUNC CALL-NO ARGUMENT OPERAND
                               4894 *
    0EE8 2C               0EE8 4895 TRMUSC DC    AL(B@LCOP)(B@CUSC)        UNSTACK CHAR ELEMENT OPCODE
    0EE9 01               0EE9 4896        DC    XL1'01'                   UNSTACK CHAR ELEMENT OPERAND
                               4897 ************************************************************************
                               4898 *            END OF LET-TERMINATION SECTION                           *
                               4899 ************************************************************************
                               4901 ***********************************************************************
                               4902 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               4903 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               4904 *                                                                     *
                               4905 ***********************************************************************
                               4906 *STATUS                                                               *
                               4907 *   VERSION 1 MODIFICATION 4                                          *
                               4908 *                                                                     *
                               4909 *FUNCTION                                                             *
                               4910 *   BSTRIF IS EXECUTED TO TRANSLATE IF STATEMENTS WITH SUB-STRING     *
                               4911 *   OPERANDS AS THEY OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE    *
                               4912 *   PSEUDO INSTRUCTION SEQUENCE AND TO PLACE THE PSEUDO INSTRUCTION   *
                               4913 *   SEQUENCE IN VIRTUAL MEMORY.                                       *
                               4914 *                                                                     *
                               4915 *ENTRY POINTS                                                         *
                               4916 *   BSTRIF HAS ONLY ONE ENTRY POINT:                                  *
                               4917 *        BSTRIF - TRANSLATE IF STATEMENTS                             *
                               4918 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               4919 *        B     BSTRIF                                                 *
                               4920 *                                                                     *
                               4921 *INPUT                                                                *
                               4922 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               4923 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               4924 *     LEADING KEYWORD, IF.                                            *
                               4925 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               4926 *     FIRST CHARACTER IN THE LEADING KEYWORD, IF.                     *
                               4927 *                                                                     *
                               4928 *OUTPUT                                                               *
                               4929 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               4930 *     GENERATED BY BSTRIF IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               4931 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               4932 *     SEQUENCES.                                                      *
                               4933 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               4934 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               4935 *                                                                     *
                               4936 *EXTERNAL REFERENCES                                                  *
                               4937 *   * B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.   *
                               4938 *   * B$PUTC - (B$PCAD, B$PNBY, B$PVAD) - ENTRY TO COMPILER VIRTUAL   *
                               4939 *              MEMORY OUTPUT ROUTINE.                                 *
                               4940 *   * B$CSCN - (B$CSTR) - ENTRY TO COMPILER CHARACTER EXPRESSION      *
                               4941 *              SCAN ROUTINE.                                          *
                               4942 *   * B$SCAN - ENTRY TO COMPILER ARITHMETIC EXPRESSION SCAN ROUTINE.  *
                               4943 *   * B$DIST - (BSDST2) - ENTRY TO COMPILER DISTRIBUTOR ROUTINE.      *
                               4944 *   * B$SYMB - (BSCRSW, B$SSTA) - ENTRY TO COMPILER SYMBOL            *
                               4945 *              TRANSLATION ROUTINE.                                   *
                               4946 *   * B$ZDBN - ENTRY TO COMPILER DECIMAL TO BINARY CONVERSION         *
                               4947 *              ROUTINE.                                               *
                               4948 *   * B$BTAB - (BSBRVA, B$BRLN) - ENTRY TO COMPILER BRANCH ADDRESS    *
                               4949 *              TABLE ROUTINE.                                         *
                               4950 *   * B$COMN - (B$PRM1, B$CADR) - COMPILER CORE RESIDENT COMMON SCTN. *
                               4951 *                                                                     *
                               4952 *EXITS, NORMAL                                                        *
                               4953 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               4954 *                                                                     *
                               4955 *EXITS, ERROR                                                         *
                               4956 *   N/A                                                               *
                               4957 *                                                                     *
                               4958 *TABLES/WORK AREAS                                                    *
                               4959 *   * RELATIONAL OPERATOR - CONDITION CODE TABLE - EXTERNAL TO        *
                               4960 *     BSTRIF, THIS 14-BYTE TABLE IS USED TO DETERMINE THE ONE BYTE    *
                               4961 *     BRANCH-ON-CONDITION CONDITION CODE WHICH CORRESPONDS TO THE     *
                               4962 *     RELATIONAL OPERATOR PRESENT IN THE SOURCE STATEMENT.  THE       *
                               4963 *     ENTRIES ARE TWO BYTES IN LENGTH, EACH TWO-BYTE ENTRY CONSISTS   *
                               4964 *     ONE-BYTE HEXIDECIMAL REPRESENTATION OF THE RELATIONAL           *
                               4965 *     OPERATOR AND A ONE-BYTE BRANCH-ON-CONDITION CONDITION CODE.     *
                               4966 *     THE TABLE IS LOCATED IN THE COMPILER CORE RESIDENT COMMON       *
                               4967 *     SECTION, BZCOMN.                                                *
                               4968 *                                                                     *
                               4969 *ATTRIBUTES                                                           *
                               4970 *   BSTRIF IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               4971 *                                                                     *
                               4972 *CHARACTER CODE DEPENDENCY                                            *
                               4973 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A               *
                               4974 *   PARTICULAR INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER      *
                               4975 *   SET.                                                              *
                               4976 *                                                                     *
                               4977 *NOTES                                                                *
                               4978 *   ERROR PROCEDURES                                                  *
                               4979 *      N/A                                                            *
                               4980 *                                                                     *
                               4981 *   REGISTER USAGE                                                    *
                               4982 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               4983 *                                                                     *
                               4984 *   SAVED/RESTORED AREAS                                              *
                               4985 *      N/A                                                            *
                               4986 *                                                                     *
                               4987 *   MODIFICATION CONSIDERATIONS                                       *
                               4988 *      BSTRIF CROSSES A SECTOR BOUNDARY AND RESIDES ON TWO SECTORS.   *
                               4989 *      CO-RESIDENT ON THE SECOND ONE WITH BKSUBG.  ANY MODIFICATIONS  *
                               4990 *      MUST MAINTAIN LINKAGE BETWEEN THE TWO SECTORS.  CONSIDER       *
                               4991 *      CHANGE IN THE ENTRY ADDRESS OF BKSUBG, AND REALIZE THE         *
                               4992 *      LIMITATION OF THE SECTOR BOUNDARY UPON SIZE.                   *
                               4993 *                                                                     *
                               4994 *   REQUIRED MODULES                                                  *
                               4995 *        @SYSEQ - COMMON SYSTEM EQUATES.                              *
                               4996 *        @FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUE EQUATES. *
                               4997 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE THE NUCLEUS EQUATES.  *
                               4998 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               4999 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               5000 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               5001 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               5002 *        $B$EQU - COMPILER FIXED ADDRESS EQUATES.                     *
                               5003 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               5004 *                                                                     *
                               5005 *   OTHER                                                             *
                               5006 *      BSTRIF IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               5007 ***********************************************************************

    0F00                       5009       ORG   *,256,0                   PLACE MODULE AT PAGE BOUNDARY
                          0F00 5010       USING *,@BR                     ESTABLISH BASE ADDRESSING
                          0F00 5011 BSTRIF EQU   *                         ENTRY POINT
                               5012 ***********************************************************************
                               5013 *            THIS IS THE RETURN ENTRY POINT FROM PROCESSING THE       *
                               5014 *            RELATIONAL OPERATOR IN SECTION TWO OF BSTRIF. RESET      *
                               5015 *            THE LOOP COUNTER AND CONTINUE TO PROCESS THE SECOND      *
                               5016 *            OPERAND OF THE SUB-STRING IF STATEMENT.                  *
                               5017 ***********************************************************************
    0F00 7C 01 F4              5018        MVI   BITLSW(,@BR),@B1          RESTORE LOOP COUNTER
    0F03 D0 87 0D              5019        B     BIT100(,@BR)              GO PROCESS SECOND OPERAND
                               5020 ***********************************************************************
                               5021 *            ADVANCE TEXT CHARACTER POINTER TO FIRST CHARACTER        *
                               5022 *            OF IDENTIFIER AND INITIALIZE LOOP COUNTER TO ZERO.       *
                               5023 ***********************************************************************
                          0F06 5024 BITRE1 EQU   *                         PRIMARY ENTRY POINT
    0F06 3C 02 0873            5025        MVI   B$NUMC,B@LKIF             SET GET ROUTINE TO SKIP KEYWORD
    0F0A 7C 00 F4              5026        MVI   BITLSW(,@BR),@ZERO        INITIALIZE LOOP SWITCH TO ZERO
    0F0D 74 01 E8              5027 BIT100 ST    BITCA2(,@BR),@BR          SAVE BSTRIF CORE ADDRESS
    0F10 C0 87 0867            5028        B     B$GETC                    SET TEXT CHARACTER POINTER
                               5029 ***********************************************************************
                               5030 *            PROCESS THE IDENTIFIER VIA A CALL TO BDSYMB. IF THE      *
                               5031 *            IDENTIFIER IS A CHARACTER REF. THE SWITCH BSCRSW WILL    *
                               5032 *            BE ON AND THE VADDR OF THE REF WILL BE LOCATED AT        *
                               5033 *            B$BCKT.                                                  *
                               5034 ***********************************************************************
    0F14 BD 7D 00              5035        CLI   B@CHAR(,@XR),B@SQUO       IF THE OPERAND IS A LITERAL
    0F17 F2 01 0B              5036        JNE   BIT110                    * BYPASS BDSYMB CALL
                               5037 ***********************************************************************
                               5038 *            OPERAND IS A CHARACTER LITERAL, DON'T USE BDSYMB         *
                               5039 ***********************************************************************
    0F1A 3C 00 0873            5040        MVI   B$NUMC,B@GETS             DISABLE THE GET ROUTINE
    0F1E C0 87 14B0            5041        B     B$CSCN                    GO PROCESS CHAR LITERAL OPERAND
    0F22 F2 87 70              5042        J     BIT200                    CONTINUE PROCESSING
    0F25 3C 01 1BAC            5043 BIT110 MVI   B$SSTA,@B1                ENABLE DETECTION OF 'STR'
    0F29 C0 87 0DBC            5044        B     B$SYMB                    TRANSLATE THE IDENTIFIER
    0F2D 3C 00 159E            5045        MVI   B$KWSW,@ZERO              CLEAR KEYWORD SWITCH
    0F31 3D 00 0E42            5046        CLI   B$CRSW,@ZERO              IS CHARACTER REF SWITCH ON ?
    0F35 D0 01 7E              5047        BNE   BIT160(,@BR)              YES-GO PROCESS CHAR REF
```

```
                               5049 ***********************************************************************
                               5050 *            THE IDENTIFIER IS A STR FUNCTION. SO ADVANCE TEXT        *
                               5051 *            CHARACTER POINTER TO THE LEADING CHARACTER OF THE        *
                               5052 *            CHARACTER REF IN THE STR FUNCTION AND PROCESS THE        *
                               5053 *            REST OF THE STRING.                                      *
                               5054 ***********************************************************************
    0F38 3C 02 0873            5055        MVI   B$NUMC,B@LLET-1           SET GET ROUTINE TO SKIP 'STR'
    0F3C C0 87 0867            5056        B     B$GETC                    ADVANCE TEXT CHARACTER POINTER
    0F40 C0 87 14B0            5057        B     B$CSCN                    PROCESS CHAR REF WITHIN 'STR'
    0F44 3C 00 159E            5058        MVI   B$KWSW,@ZERO              TURN OFF KETWORK SWITCH
    0F48 C0 87 1514            5059        B     B$SCAN                    PROCESS FIRST 'STR' PARAMETER
    0F4C BD 5D 00              5060        CLI   @ZERO(,@XR),B@RPAR        IS 2ND PARAMETER MISSING ?
    0F4F D0 01 64              5061        BNE   BIT120(,@BR)              NO-GO PROCESS 2ND PARAMETER
    0F52 D2 02 F6              5062        LA    BITSTX(,@BR),@XR          SET CADDR PARAMETER FOR PUT RTN
    0F55 34 02 0A40            5063        ST    B$PCAD,@XR                * WITH 'STX' INSTR ADDR
    0F59 3C 01 0A41            5064        MVI   B$PNBY,B@LSTX-1           SET LNGTH PARAMETER FOR PUT RTN
    0F5D C0 87 093A            5065        B     B$PUTC                    GO GENERATE PMC
    0F61 D0 87 68              5066        B     BIT140(,@BR)              GO CONTINUE PROCESSING
    0F64 C0 87 1514            5067 BIT120 B     B$SCAN                    PROCESS LAST 'STR' PARAMETER
    0F68 D2 02 F8              5068 BIT140 LA    BITFNO(,@BR),@XR          LOAD CADDR OF 'FNO' INSTRUCTION
    0F6B 34 02 0A40            5069        ST    B$PCAD,@XR                SET CADDR PARM FOR PUT ROUTINE
    0F6F 3C 02 0A41            5070        MVI   B$PNBY,B@LFN0-1           SET LENGTH PARM FOR PUT ROUTINE
    0F73 C0 87 093A            5071 BIT150 B     B$PUTC                    LINK TO GENERATE PMC
    0F77 C0 87 0867            5072        B     B$GETC                    ADVANCE TEXT CHARACTER POINTER
    0F7B D0 87 95              5073        B     BIT200(,@BR)              GO SET LOOP SWITCH VALUE
```

```
                               5075 ***********************************************************************
                               5076 *            THE OPERAND JUST PROCESSED BY BDSYMB WAS A CHARACTER     *
                               5077 *            REFERENCE. THE VADDR OF THE REFERENCE IS CONTAINED       *
                               5078 *            AT RSBCKT AND THE TEXT CHARACTER POINTER REFERENCES      *
                               5079 *            THE CHARACTER FOLLOWING THE LAST CHARACTER OF THE        *
                               5080 *            IDENTIFIER.                                              *
                               5081 ***********************************************************************
    0F7E D2 02 95              5082 BIT160 LA    BIT200(,@BR),@XR          SAVE RETURN ADDRESS
    0F81 34 02 150D            5083        ST    B$CRAD,@XR                * IN BECSCN
    0F85 34 01 1509            5084        ST    B$CRBS,@BR                SAVE BASE REGISTER FOR RETURN
                               5085 *                                      * FROM BECSCN
    0F89 C2 01 14BB            5086        LA    B$CBAS,@BR                LOAD BECSCN BASE REGISTER
    0F8D 35 02 0878            5087        L     B$GPTR,@XR                LOAD TEXT CHARACTER POINTER
    0F91 C0 87 14CC            5088        B     B$CSTR                    GO PROCESS CHAR REF
                               5089 ***********************************************************************
                               5090 *            THE OPERAND HAS BEEN PROCESSED. NOW INCREMENT THE        *
                               5091 *            LOOP SWITCH AND DETERMINE IF PROCESSING IS FINISHED.     *
                               5092 ***********************************************************************
    0F95 5E 00 F4 F5           5093 BIT200 ALC   BITLSW(@B1,@BR),BIT001(,@BR)  INCREMENT LOOP SNITCH BY 1
    0F99 7D 02 F4              5094        CLI   BITLSW(,@BR),@CADDR       IS LOOP SWITCH * 2 ?
    0F9C D0 81 A9              5095        BE    BIT300(,@BR)              YES-GO TO TERMINATION CODE
                               5096 ***********************************************************************
                               5097 *            LOOP SWITCH = 1, SO WE NOW MUST COMPUTE THE CONDITION    *
                               5098 *            CODE WHICH CORRESPONDS TO THE RELATIONAL OPERATOR(S)     *
                               5099 *            IN THE BASIC STATEMENT. WE MUST ACCESS SECTION TWO       *
                               5100 *            IN ORDER TO PROCESS THE RELATIONAL OPERATOR.             *
                               5101 ***********************************************************************
    0F9F 34 01 1AF9            5102        ST    B$CADR,@BR                SAVE OPERAND PROC SECTION CADDR
    0FA3 7C 00 E6              5103        MVI   BIT390+@D1(,@BR),@ZERO    SAVE DISP INTO SEGMENT 2
    0FA6 F2 87 0F              5104        J     BIT340                    GO ACCESS SEGMENT 2
                               5105 ***********************************************************************
                               5106 *            SET PARAMETER TO SKIP EMBEDDED KEYWORD 'GOTO' OR 'THEN'  *
                               5107 *            TO ADVANCE THE TEXT CHARACTER POINTER TO THE LINE NO.    *
                               5108 ***********************************************************************
    0FA9 3C 04 0873            5109 BIT300 MVI   B$NUMC,B@LTHN             SET GET RTN TO SKIP KEYWORD
    0FAD C0 87 0867            5110        B     B$GETC                    ADVANCE TEXT CHAR POINTER
    0FB1 7C 4A E6              5111        MVI   BIT390+@D1(,@BR),BITTRM   SAVE TERMINATION DISPLACEMENT
                               5112 ***********************************************************************
                               5113 *            CONVERT LINE NO. FROM DECMAL TO BINARY                   *
                               5114 ***********************************************************************
    0FB4 C0 87 19F2            5115        B     B$ZDBN                    LINK TO CONVERT LINE NUMBER
```

```
                               5117 ***********************************************************************
                               5118 *            ACCESS PART 2 OF IF STATEMENT PROCESSOR TO              *
                               5119 *            COMPLETE PSEUDOCODE GENERATION.                          *
                               5120 ***********************************************************************
    0FB8 5D 01 E8 F1           5121 BIT340 CLC   BITCA2(,@BR),BITPBA(@CADDR,@BR) IF CURR SEG CAME FR DISK
    0FBC F2 81 10              5122        JE    BIT360                    * GO LOAD & EXEC 2ND SEGMENT
                               5123 ***********************************************************************
                               5124 *            CURRENT SEGMENT WAS CORE RESIDENT - TEST WHETHER 2ND    *
                               5125 *            SEGMENT HAS ALSO BEEN LOADED INTO CORE.                  *
                               5126 ***********************************************************************
    0FBF 5C 01 EB ED           5127 BIT350 MVC   BITFCP(,@BR),BITFPE(@CADDR,@BR)  SET FINAL CORE PAGE
    0FC3 4E 00 EA 043B         5128        ALC   BITFCP-1(,@BR),$EXFTR(@B1)  CALC MAX PROCESSOR CORE PAGE
    0FC8 5D 01 E8 EB           5129        CLC   BITCA2(,@BR),BITFCP(@CADDR,@BR)  IF 2ND SEGMENT IN CORE
    0FCC F2 82 0F              5130        JL    BIT380                      * GO SET TO EXEC 2ND SEGMENT
                               5131 ***********************************************************************
                               5132 *            2ND SEGMENT IS DISK RESIDENT - ESTABLISH DISTRIBUTOR    *
                               5133 *            PARAMETERS FOR CORELOADING & EXECUTING 210 SEGMENT      *
                               5134 ***********************************************************************
    0FCF 5C 01 E8 F1           5135 BIT360 MVC   BITCA2(,@BR),BITPBA(@CADDR,@BR)  SET UP DISKLOAD CADDR
                               5136 ***********************************************************************
                               5137 *            EXIT TO DISTRIBUTOR TO ACCESS 2ND SEGMENT                *
                               5138 ***********************************************************************
    0FD3 D2 02 E7              5139 BIT370 LA    BITAD2(,@BR),@XR          LOAD DIST PARM CADDR
    0FD6 5C 00 E8 E6           5140        MVC   BITCA2(@B1,@BR),BIT390+@D1(,@BR)  SET CADDR TERM SECTION
    0FDA C0 87 073A            5141        B     B$DST2                    GO LOAD & EXEC 2ND SEGMENT
                               5142 ***********************************************************************
                               5143 *            2ND SEGMENT IS CORE RESIDENT - BRANCH TO NEST            *
                               5144 *            CONSECUTIVE CORE APGE & CONTINUE EXECUTION               *
                               5145 ***********************************************************************
    0FDE 75 01 E8              5146 BIT380 L     BITCA2(,@BR),@BR          LOAD THE BASE ADDRESS FOR
    0FE1 76 01 EF              5147        A     BITBLS(,@BR),@BR          * 2ND SEGMENT
    0FE4 D0 87 00              5148 BIT390 B     BITSG2(,@BR)              GO EXECUTE THE 2ND SEGMENT
```

```
                               5150 *********************************************************************
                               5151 *            CONSTANTS & MORKAREAS TO ACCESS THE 2ND SEGMENT          *
                               5152 *********************************************************************
                          0000 5153 BITSG2 EQU   0                          PAGE 2 ENTRY PT DISP
                          004A 5154 BITTRM EQU   X'4A'                      DISP TO TERM ENTRY IN SECTION 2
                          0FE7 5155 BITAD2 EQU   *                          DISTR PARMS FOR SEG-2 EXEC
    0FE7                  0FE8 5156 BITCA2 DS    CL(@CADDR)                 IF SEGMENT CORE ADDRESS
    0FE9 20               0FE9 5157        DC    AL1(B@DSIF+4)              BSTRIF SEG-2 PHYS SECTOR ADDR
    0FEA                  0FEB 5158 BITFCP DS    CL(@CADDR)                 FINAL AVAILABLE CORE PAGE ADDR
    0FEC 1F00             0FED 5159 BITFPE DC    AL(@CADDR)(B$CSXA-B@BLSZ) FINAL PAGE BEFORE EXTENSION
    0FEE 0100             0FEF 5160 BITBLS DC    AL(@CADDR)(B@BLSZ)         LENGTH OF CORE PAGE
    0FF0 0600             0FF1 5161 BITPBA DC    AL(@CADDR)(B$CSBF)         PROCESSOR DISK BUFFER CADDR
    0FF2 0001             0FF3 5162 BITBN1 DC    IL(@VADDR)'1'              BINARY 1
                               5163 *********************************************************************
                               5164 *            CONSTANTS, PSUEDO INSTRUCTION IMAGES AND WORKAREAS       *
                               5165 *********************************************************************
    0FF4                  0FF4 5166 BITLSW DS    CL1                        LOOP SWITCH
    0FF5 01               0FF5 5167 BIT001 DC    XL1'01'                    INCR FOR LOOP SWITCH VALUE
                               5168 *
    0FF6 3C               0FF6 5169 BITSTX DC    AL(B@LCOP)(B@CSTX)         STACK EXEC CTRL CODE OPCODE
    0FF7 FF               0FF7 5170        DC    XL1'FF'                    STACK EXEC CTRL CODE OPERAND
                               5171 *
    0FF8 12               0FF8 5172 BITFNO DC    AL(B@LCOP)(B@CFN0)         FUNCTION CALL-NO ARGUMENT OPCODE
    0FF9 5100             0FFA 5173 BITOOP DC    AL2(V$CSSR)                FUNCTION CALL-NO ARGUMENT OPERAND
```

```
                               5175 ************************************************************************
                               5176 *            ESTABLISH IF SEGMENT 2 ADDRESSABILITY                     *
                               5177 ************************************************************************
    1000                       5178        ORG   BSTRIF+B@BLSZ             BEGIN SEGMENT 2 AT PAGE BNDRY
                          1000 5179        USING *,@BR                     DEFINE SEGMENT 2 BASE ADDRESS
                               5180 ************************************************************************
                               5181 *            THIS SECTION WILL SEARCH THE RELATIONAL OPERATOR          *
                               5182 *            TABLE FOR THE CONDITION CODE AND SAVE THE CONDITION       *
                               5183 *            CODE AT ISPRM1 FOR LATER USE IN A BRANCH-ON-CONDITION     *
                               5184 *            PSEUDO INSTRUCTION.                                       *
                               5185 ************************************************************************
                          1000 5186 BITREL EQU   *                         RELATIONAL OPERATOR ENTRY POINT
    1000 35 02 0878            5187        L     B$GPTR,@XR                LOAD TEXT CHARACTER POINTER
                               5188 ************************************************************************
                               5189 *            STORE 1ST RELATIONAL OPERATOR IN OPERAND OF CLI INSTR     *
                               5190 ************************************************************************
    1004 6C 00 2B 00           5191        MVC   BIT280+@Q(@B1,@BR),B@CHAR(,@XR)  STORE 1ST RELATNL OPTR
                               5192 ************************************************************************
                               5193 *            CHECK FOR COMPOUND RELATIONAL OPERATOR                    *
                               5194 ************************************************************************
    1008 C0 87 0867            5195        B     B$GETC                    ADVANCE TEXT CHARACTER PTR
    100C BD 7E 00              5196        CLI   B@CHAR(,@XR),B@EQUL       IF CHARACTER IS '='
    100F D0 81 1F              5197        BE    BIT240(,@BR)              * GO COMPUTE OPERATOR
    1012 BD 6E 00              5198        CLI   B@CHAR(,@XR),B@GRTR       IF CHARACTER IS '>'
    1015 D0 81 1F              5199        BE    BIT240(,@BR)              * GO COMPUTE OPERATOR
                               5200 ************************************************************************
                               5201 *            THE OPERATOR IS NOT COMPOUND-DISABLE GET ROUTINE          *
                               5202 ************************************************************************
    1018 3C 00 0873            5203        MVI   B$NUMC,B@GETS             DISABLE THE GET ROUTINE
    101C D0 87 23              5204        B     BIT260(,@BR)              GO SEARCH OPERATOR TABLE
                               5205 ************************************************************************
                               5206 *            IF THE RELATIONAL OPERATOR IS COMPOUND. ADD TIE TWO       *
                               5207 *            RELATIONAL OPERATORS TO DERIVE A CHARACTER CODE           *
                               5208 ************************************************************************
    101F 6E 00 2B 00           5209 BIT240 ALC   BIT280+@Q(@B1,@BR),B@CHAR(,@XR)  ADD OPERATORS
                               5210 ************************************************************************
                               5211 *            SEARCH THE RELATIONAL OPERATOR TABLE FOR THE              *
                               5212 *            CORRESPONDING CONDITION CODE TO BE PLACED IN THE          *
                               5213 *            BRANCH ON CONDITION PSEUDO INSTRUCTION                    *
                               5214 ************************************************************************
    1023 C2 02 1AF8            5215 BIT260 LA    B$TOTB,@XR                LOAD TABLE BASE ADDRESS
    1027 E2 02 02              5216 BIT270 LA    B$TLTH(,@XR),@XR          ADD LENGTH TO ADDR
    102A BD 00 00              5217 BIT280 CLI   B$TOD1(,@XR),*-*          IF TEXT OPERATOR = TABLE ENTRY
    102D D0 01 27              5218        BNE   BIT270(,@BR)              * FALL THROUGH
                               5219 ************************************************************************
                               5220 *            SAVE CONDITION CODE IN OPERAND FIELD OF 'BRC' INSTR       *
                               5221 ************************************************************************
    1030 2C 00 1AF3 01         5222        MVC   B$PRM1(@B1),B$TCD2(,@XR)  SAVE BRC CONDITION CODE
```

```
                               5224 ***********************************************************************
                               5225 *            RETURN TO PROCESS NEXT CHARACTER EXPRESSION              *
                               5226 ***********************************************************************
    1035 3D 06 1AF8            5227        CLI   B$CADR-@B1,BITEN2         IF OPERAND SECTION IS ON DISK
    1039 F2 81 07              5228        JE    BIT290                    * GO LOAD AND EXEC FROM DISK
                               5229 ***********************************************************************
                               5230 *            OPERAND PROCESSOR SECTION IS CORE RESIDENT - RESTORE     *
                               5231 *            STATUS AND BRANCH TO OPERAND PROCESSOR SECTION.          *
                               5232 ***********************************************************************
    103C 35 01 1AF9            5233        L     B$CADR,@BR                RESTORE OPERAND SECTN BASE ADDR
    1040 D0 87 00              5234        B     @ZERO(,@BR)               GO TO OPERAND PROC SECTION
                               5235 ***********************************************************************
                               5236 *            OPERAND PROCESSOR SECTION IS DISK RESIDENT - LOAD        *
                               5237 *            AND RETURN.                                              *
                               5238 ***********************************************************************
    1043 D2 02 8D              5239 BIT290 LA    TWOAD2(,@BR),@XR          LOAD DIST PARAMETER CADDR
    1046 C0 87 073A            5240        B     B$DST2                    LOAD & RTRN TO OPRND PROC SECTN
```

```
                               5242 ***********************************************************************
                               5243 *            THIS SECTION WILL GENERATE A COMPARE CHARACTERS         *
                               5244 *            PSEUDO INSTRUCTION, A BRANCH ON CONDITION PSEUDO        *
                               5245 *            INSTRUCTION, ESTABLISH CONDITIONS FOR BRANCH TABLE      *
                               5246 *            RESOLUTION AND EXIT TO THE COMPILER DISTRIBUTOR.        *
                               5247 ***********************************************************************
                          104A 5248 BITERM EQU   *                         TERMINATION SECTION ENTRY POINT
    104A D2 02 8C              5249        LA    BITCMC(,@BR),@XR          LOAD CADDR OF 'CMC' INSTRUCTION
    104D 34 02 0A40            5250        ST    B$PCAD,@XR                SET CADDR PARM FOR PUT RTN
    1051 3C 00 0A41            5251        MVI   B$PNBY,B@LCMC-1           SET LENGTH PARM FOR PUT RTN
    1055 C0 87 093A            5252        B     B$PUTC                    LINK TO GENERATE PMC
                               5253 ***********************************************************************
                               5254 *            GENERATE BRANCH ON CONDITION INSTRUCTION IMAGE          *
                               5255 ***********************************************************************
    1059 4C 00 89 1AF3         5256        MVC   BITB02(@B1,@BR),B$PRM1    GET CONDITION CODE 'FRM' SEG-1
    105E D2 02 86              5257        LA    BITBRC(,@BR),@XR          LOAD CADDR OF 'BRC' INSTRUCTION
    1061 34 02 0A40            5258        ST    B$PCAD,@XR                SET CADDR PARM FOR PUT RTN
    1065 3C 03 0A41            5259        MVI   B$PNBY,B@LBRC-1           SET LENGTH PARAMETER FOR PUT RTN
    1069 C0 87 093A            5260        B     B$PUTC                    LINK TO GENERATE PMC
                               5261 ***********************************************************************
                               5262 *            ESTABLISH ADDRESS AND LINE NUMBER PARAMETERS FOR        *
                               5263 *            BRANCH TABLE RESOLUTION                                 *
                               5264 ***********************************************************************
    106D 0C 01 19EF 0A43       5265        MVC   B$BRVA,B$PVAD(@VADDR)     SET VADDR PARAMETER
    1073 1F 01 19EF 8B         5266        SLC   B$BRVA,BITLNG(@VADDR,@BR) SET PARM FOR VADDR OF 'BRC'
    1078 0C 01 19F1 1A6A       5267        MVC   B$BRLN,B$BINO(B@LCLN)     SET LINE NO. PARM
    107E C0 87 1996            5268        B     B$BTAB                    LINK TO SET RESOLUTION COND.
                               5269 ***********************************************************************
                               5270 *            PROCESSING IS FINISHED RETURN TO DISTRIBUTOR            *
                               5271 ***********************************************************************
    1082 C0 87 0700            5272        B     B$DIST                    RETURN TO DISTRIBUTOR
                               5273 ***********************************************************************
                               5274 *            SEGMENT2 CONSTANTS ANC WORK AREAS                       *
                               5275 ***********************************************************************
    1086 44               1086 5276 BITBRC DC    AL(B@LCOP)(B@CBRC)        BRANCH ON CONDITION OPCODE
    1087 0000             1088 5277 BITB01 DC    XL(B@LCVA)'00'            BRANCH ON COND VADDR OPERAND
    1089                  1089 5278 BITB02 DS    CL(B@LCCC)                BRANCH ON COND COND CODE OPERAND
    108A 0002             108B 5279 BITLNG DC    AL(@VADDR)(B@LCCC+1)      LENGTH OF COND CODE + 1
    108C 42               108C 5280 BITCMC DC    AL(B@LCOP)(B@CCMC)        COMPARE CHARACTER OPCODE
                          0006 5281 BITEN2 EQU   X'06'                     CORE PGE NO. OF DISK BUFFER
                          108D 5282 TWOAD2 EQU   *                         CONSTANTS AND WORK AREAS USED
    108D 0600             108E 5283 TWOCA2 DC    AL(@CADDR)(B$CSBF)        * BY THE RELATIONAL OPERATOR
    108F 1C               108F 5284        DC    AL1(B@DSIF)               * SECTION TO RETURN TO THE
                               5285 *                                      * OPERAND PROCESSOR SECTION
                               5286 ***********************************************************************
                               5287 *            END OF SUBSTRING IF STATEMENT PROCESSOR                 *
                               5288 ***********************************************************************
                               5290 ************************************************************************
                               5291 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                   *
                               5292 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083       *
                               5293 *                                                                      *
                               5294 ************************************************************************
                               5295 *STATUS                                                                *
                               5296 *   VERSION 1 MODIFICATION 0                                           *
                               5297 *                                                                      *
                               5298 *FUNCTION                                                              *
                               5299 *   BKSUBG IS EXECUTED TO TRANSLATE GOSUB STATEMENTS AS THEY OCCUR     *
                               5300 *   IN A BASIC PROGRAM INTO THE APROPRIATE PSEUDOCODE AND TO PLACE     *
                               5301 *   THE PSEUDOCODE IN VIRTUAL MEMORY.                                  *
                               5302 *                                                                      *
                               5303 *ENTRY POINTS                                                          *
                               5304 *   BKSUBG HAS ONLY ONE ENTRY POINT:                                   *
                               5305 *        BKSUBG - TRANSLATE GOSUB STATEMENT                            *
                               5306 *   THE FORMAT OF THE CALLING SEQUENCE IS:                             *
                               5307 *        B     BKSUBG                                                  *
                               5308 *                                                                      *
                               5309 *INPUT                                                                 *
                               5310 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING   *
                               5311 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE        *
                               5312 *     LEADING KEYWORD, GOSUB.                                          *
                               5313 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST  *
                               5314 *     CHARACTER IN THE LEADING KEYWORD, GOSUB.                         *
                               5315 *                                                                      *
                               5316 *OUTPUT                                                                *
                               5317 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE         *
                               5318 *     GENERATED BY BKSUBG IS STORED IN THE NEXT AVAILABLE VIRTUAL      *
                               5319 *     MEMORY LOCATION, FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               5320 *     SEQUENCES.                                                       *
                               5321 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE        *
                               5322 *     CHARACTER WHICH TERMINATES THE STATEMENT.                        *
                               5323 *   * BSBRVA - CONTAINS THE VIRTUAL ADDRESS OF THE RIGHT BYTE OF THE   *
                               5324 *     ADDRESS OPERAND FIELD IN THE RETURN-ADDRESS STACKING             *
                               5325 *     INSTRUCTION.                                                     *
                               5326 *   * BSNXSW - SET TO ON STATUS TO CAUSE RESOLUTION OF THE RETURN-     *
                               5327 *     ADDRESS STACKING INSTRUCTION OPERAND ADDRESS.                    *
                               5328 *                                                                      *
                               5329 *EXTERNAL REFERENCES                                                   *
                               5330 *   * B$GETC - (B$NUMC) - ENTRY TO BASIC RETRIEVAL ROUTINE.            *
                               5331 *   * B$PUTC - (B$PCAD, B$PNBY, B$PVAD) - ENTRY TO COMPILER VIRTUAL    *
                               5332 *              MEMORY OUTPUT ROUTINE.                                  *
                               5333 *   * B$BTAB - (B$BRVA, B$BRLN) - ENTRY TO BASIC COMPILER BRANCH       *
                               5334 *              TABLE ROUTINE.                                          *
                               5335 *   * B$ZDBN - (B$BINO) - ENTRY TO BASIC COMPILER ZONED DECIMAL TO     *
                               5336 *              BINARY CONVERSION ROUTINE.                              *
                               5337 *   * B$DIST - (B$NXSW) - ENTRY TO BASIC COMPILER DISTRIBUTOR          *
                               5338 *                                                                      *
                               5339 *EXITS, NORMAL                                                         *
                               5340 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR                       *
                               5341 *                                                                      *
                               5342 *EXITS, ERROR                                                          *
                               5343 *   N/A                                                                *
                               5344 *                                                                      *
                               5345 *TABLES/WORK AREAS                                                     *
                               5346 *   N/A                                                               *
                               5347 *                                                                     *
                               5348 *ATTRIBUTES                                                           *
                               5349 *   BKSUBG IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               5350 *                                                                     *
                               5351 *CHARACTER CODE DEPENDENCY                                            *
                               5352 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               5353 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               5354 *                                                                     *
                               5355 *NOTES                                                                *
                               5356 *   ERROR PROCEDURES                                                  *
                               5357 *      N/A                                                            *
                               5358 *                                                                     *
                               5359 *   REGISTER USAGE                                                    *
                               5360 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               5361 *                                                                     *
                               5362 *   SAVED/RESTORED AREAS                                              *
                               5363 *      N/A                                                            *
                               5364 *                                                                     *
                               5365 *   MODIFICATION CONSIDERATIONS                                       *
                               5366 *      BKSUBG IS CO-RESIDENT ON A SECTOR WITH BSTRIF.  ANY         1-4*
                               5367 *      MODIFICATION SHOULD CONSIDER THE CO-RESIDENCY AND THE       1-4*
                               5368 *      LIMITATION OF THE SECTOR BOUNDARY ON SIZE.                  1-4*
                               5369 *                                                                     *
                               5370 *   REQUIRED MODULES                                                  *
                               5371 *        @SYSEQ - COMMON SYSTEM EQUATES                               *
                               5372 *        @FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUE EQUATES  *
                               5373 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES       *
                               5374 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES                    *
                               5375 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES                         *
                               5376 *        @ERMEQ - ERROR MESSAGE EQUATES                               *
                               5377 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES                       *
                               5378 *        $B$EQU - COMPILER FIXED EQUATES                              *
                               5379 *        $B@EQU - COMPILER SYSTEM EQUATES                             *
                               5380 *                                                                     *
                               5381 *   OTHER                                                             *
                               5382 *      BKSUBG IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS       *
                               5383 ***********************************************************************

                               5385 *
                               5386 * ENTER BKSUBG - 'GOSUB' STATEMENT ROUTINE
                               5387 *
                          1090 5388 BKSUBG EQU   *                         BKSUBG ENTRY POINT
                               5389 *
                               5390 * SET INPUT PARAMETER TO SKIP KEYWORD 'GOSUB'
                               5391 *
    1090 3C 05 0873            5392 BKS010 MVI   B$NUMC,B@LGSB             SET GET RTN TO SKIP 'GOSUB'
    1094 C0 87 0867            5393        B     B$GETC                    LINK TO ADVANCE POINTER
                               5394 *
                               5395 * CONVERT 'GOSUB' LINE NUMBER TO BINARY FROM ITS DECIMAL FORM
                               5396 *
    1098 C0 87 19F2            5397 BKS020 B     B$ZDBN                    LINK TO CONVERT LINE NUMBER
```

```
                               5399 *
                               5400 * GENERATE AN ADDRESS STACKING INSTRUCTION IMAGE FOR RETURN ADDRESS
                               5401 *
    109C D2 02 E6              5402 BKS030 LA    BKSTAC(,@BR),@XR          LOAD CADDR OF 'STA' INSTR
    109F 34 02 0A40            5403        ST    B$PCAD,@XR                SET PUT RTN FOR VADDR OF 'STA'
    10A3 3C 02 0A41            5404        MVI   B$PNBY,B@LSTA-1           SET PUT RTN FOR LENGTH OF 'STA'
    10A7 C0 87 093A            5405        B     B$PUTC                    LINK TO GENERATE 'STA' IMAGE
    10AB 4C 01 EF 0A43         5406        MVC   BKSVAS(,@BR),B$PVAD(@VADDR)  SAVE VADDR AFTER 'STA' INST
                               5407 *
                               5408 * GENERATE AN UNCONDITIONAL BRANCH INSTRUCTION IMAGE IN VIRTUAL MEMORY
                               5409 *
    10B0 D2 02 E9              5410 BKS040 LA    BKSBRC(,@BR),@XR          LOAD CADDR OF 'BRA' INSTR
    10B3 34 02 0A40            5411        ST    B$PCAD,@XR                SET PUT RTN FOR VADDR OF 'BRA'
    10B7 3C 02 0A41            5412        MVI   B$PNBY,B@LBRA-1           SET PUT RTN FOR LENGTH OF 'BRA'
    10BB C0 87 093A            5413        B     B$PUTC                    LINK TO GENERATE 'BRA' IMAGE
                               5414 *
                               5415 * ESTABLISH LINE NUMBER AND VIRTUAL ADDRESS FOR RESOLUTION OF 'BRA'
                               5416 * INSTRUCTION OPERAND
                               5417 *
    10BF 0C 01 19F1 1A6A       5418 BKS050 MVC   B$BRLN,B$BINO(@VADDR)     ESTABLISH BRANCH LINE NUMBER
    10C5 0C 01 19EF 0A43       5419        MVC   B$BRVA,B$PVAD(@VADDR)     SET BRANCH TABLE VADDR
    10CB 1F 01 19EF ED         5420        SLC   B$BRVA,BKSBN1(@VADDR,@BR) ADJUST VADDR FOR 'BRA' OPERAND
    10D0 C0 87 1996            5421        B     B$BTAB                    LINK TO RESOLVE 'BRA' OPERAND
                               5422 *
                               5423 * ESTABLISH VIRTUAL ADDRESS PARAMETER FOR 'STA' BRANCH TABLE RESOLUTION
                               5424 *
    10D4 1C 01 19EF EF         5425 BKS060 MVC   B$BRVA,BKSVAS(@VADDR,@BR)  SET BRANCH TABLE VADDR
    10D9 1F 01 19EF ED         5426        SLC   B$BRVA,BKSBN1(@VADDR,@BR)  ADJUST VADDR FOR 'STA' OPERAND
                               5427 *
                               5428 * SET SWITCH ON TO CAUSE THE DISTRIBUTOR TO SET UP ADDR RESOLUTION
                               5429 * CONDITIONS
                               5430 *
    10DE 3A 07 071D            5431 BKS070 SBN   B$NXSW,B$NXMK             SET SW TO RESOLVE 'STA' ADDR
                               5432 *
                               5433 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               5434 *
    10E2 C0 87 0700            5435 BKS080 B     B$DIST                    RETURN TO DISTRIBUTOR

                               5437 ***********************************************************************
                               5438 * 'GOSUB' STMT ROUTINE PMC AND STORAGE PARAMETERS
                               5439 ***********************************************************************
                               5440 *
    10E6 34               10E6 5441 BKSTAC DC    AL(B@LCOP)(B@CSTA)        STACK ADDRESS INSTR OPCODE
    10E7 0000             10E8 5442 BKSTAO DC    XL(B@LCVA)'00'            STACK ADDRESS INSTR OPERAND
                               5443 *
    10E9 46               10E9 5444 BKSBRC DC    AL(B@LCOP)(B@CBRA)        'BRA' INSTR OPCODE
    10EA 0000             10EB 5445 BKSBRO DC    XL(B@LCVA)'00'            'BRA' INSTR OPERAND

                               5447 ***********************************************************************
                               5448 * 'GOSUB' STATEMENT ROUTINE CONSTANTS
                               5449 ***********************************************************************
                               5450 *
    10EC 0001             10ED 5451 BKSBN1 DC    IL(@VADDR)'1'             BINARY 1

                               5453 ***********************************************************************
                               5454 * 'GOSUB' STMT ROUTINE WORK AREAS
                               5455 ***********************************************************************
                               5456 *
    10EE                  10EF 5457 BKSVAS DS    CL(@VADDR)                VIRTUAL ADDRESS SAVE AREA
                               5458 ***********************************************************************
                               5459 *
                               5460 * END OF 'GOSUB' STATEMENT ROUTINE CODING
                               5461 *
                               5463 ************************************************************************
                               5464 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                   *
                               5465 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083       *
                               5466 *                                                                      *
                               5467 ************************************************************************
                               5468 *STATUS                                                                *
                               5469 *   VERSION 1 MODIFICATION 0                                           *
                               5470 *                                                                      *
                               5471 *FUNCTION                                                              *
                               5472 *   BNDATA IS EXECUTED TO TRANSLATE DATA STATEMENTS AS THEY OCCUR      *
                               5473 *   IN A BASIC PROGRAM INTO APPROPRIATE PSEUDOCODE AND TO PLACE        *
                               5474 *   THE PSEUDOCODE INTO VIRTUAL MEMORY.                                *
                               5475 *                                                                      *
                               5476 *ENTRY POINTS                                                          *
                               5477 *   BNDATA HAS ONLY ONE ENTRY POINT:                                   *
                               5478 *        BNDATA - TRANSLATE DATA STATEMENT.                            *
                               5479 *   THE FORMAT OF THE CALLING SEQUENCE IS:                             *
                               5480 *        B     BNDATA                                                  *
                               5481 *                                                                      *
                               5482 *INPUT                                                                 *
                               5483 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING   *
                               5484 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE        *
                               5485 *     LEADING KEYWORD, DATA.                                           *
                               5486 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST  *
                               5487 *     CHARACTER IN THE LEADING KEYWORD, DATA.                          *
                               5488 *   * $INLNO - CONTAINS A VALUE OF ZERO WHEN NO PREVIOUS DATA          *
                               5489 *     STATEMENTS HAVE BEEN PROCESSED.                                  *
                               5490 *   * B$CLNK - CONTAINS THE VIRTUAL ADDRESS OF THE RIGHT BYTE OF THE   *
                               5491 *     ADDRESS OPERAND FIELD IN THE LAST GENERATED DDL INSTRUCTION:     *
                               5492 *     THIS IS ONLY REQUIRED WHEN $INLNO IS NON-ZERO.                   *
                               5493 *                                                                      *
                               5494 *OUTPUT                                                                *
                               5495 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE         *
                               5496 *     GENERATED BY BNDATA IS STORED IN THE NEXT AVAILABLE VIRTUAL      *
                               5497 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION          *
                               5498 *     SEQUENCES.                                                       *
                               5499 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE        *
                               5500 *     CHARACTER WHICH TERMINATES THE STATEMENT.                        *
                               5501 *   * $INLNO - CONTAINS THE VIRTUAL ADDRESS OF THE FIRST DCA           *
                               5502 *     INSTRUCTION GENERATED FOR THE DATA STATEMENT WHEN THIS IS THE    *
                               5503 *     FIRST SUCH STATEMENT TO BE PROCESSED IN THE PROGRAM.             *
                               5504 *   * B$DLNK - CONTAINS THE VIRTUAL ADDRESS OF THE RIGHT BYTE OF THE   *
                               5505 *     ADDRESS OPERAND FIELD IN THE DDL INSTRUCTION GENERATED FOR THE   *
                               5506 *     CURRENT STATEMENT.                                               *
                               5507 *   * B$BRVA - CONTAINS THE VIRTUAL ADDRESS OF THE RIGHT BYTE OF THE   *
                               5508 *     ADDRESS OPERAND FIELD IN THE BYPASS BRANCH INSTRUCTION           *
                               5509 *     GENERATED FOR THE CURRENT STATEMENT.                             *
                               5510 *   * BSNXSW - SET TO ON STATUS TO CAUSE RESOLUTION OF THE BYPASS      *
                               5511 *     BRANCH INSTRUCTION OPERAND ADDRESS.                              *
                               5512 *                                                                      *
                               5513 *EXTERNAL REFERENCES                                                   *
                               5514 *      B$GETC - (B$NUNC) - ENTRY TO BASIC RETRIEVAL ROUTINE.           *
                               5515 *      B$PUTC - (B$PCAD, B$PNBY, B$PVAD) - ENTRY TO COMPILER VIRTUAL   *
                               5516 *               MEMORY OUTPUT ROUTINE.                                 *
                               5517 *      B$FCON - (B$CTYP, B$BCKT) - ENTRY TO BASIC COMPILER CONSTANT    *
                               5518 *               ROUTINE.                                               *
                               5519 *      B$BTAB - (B$BRVA, B$BRIN) - ENTRY TO BASIC COMPILER BRANCH     *
                               5520 *               TABLE ROUTINE.                                        *
                               5521 *      B$SCAN - (B$FVPP, B$FVPP, B$FVPS, BIFVME, B$FVMP, B$FVMS) -    *
                               5522 *               ENTRY TO BASIC COMPILER SCAN ROUTINE.                 *
                               5523 *      B$DLNK - AREA CONTAINING VIRTUAL ADDRESS OF THE RIGHT BYTE OF  *
                               5524 *               ADDRESS OPERAND FIELD OF 'DCA' INSTRUCTIONS.          *
                               5525 *      $INLNO - AREA CONTAINING VIRTUAL ADDRESS OF 'DCA' INSTRUCTIONS.*
                               5526 *      B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                  *
                               5527 *                                                                     *
                               5528 *EXITS, NORMAL                                                        *
                               5529 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               5530 *                                                                     *
                               5531 *EXITS, ERROR                                                         *
                               5532 *   N/A                                                               *
                               5533 *                                                                     *
                               5534 *TABLES/WORK AREAS                                                    *
                               5535 *   * INTERNAL CONSTANT BUCKET - 2 BYTES. INTERNAL TO BNDATA; FOR     *
                               5536 *     ACCUMULATING INTERNAL CONSTANT SYMBOL CHARACTERS IN PREPARATION *
                               5537 *     FOR A TABLE SEARCH.                                             *
                               5538 *   * INTERNAL CONSTANT TABLE - INTERNAL TO BNDATA, THIS TABLE        *
                               5539 *     CONTAINS THE CORE ADDRESSES OF VIRTUAL ADDRESS VALUES           *
                               5540 *     ASSOCIATED WITH EACH INTERNAL CONSTANT, AND A LENGTH CODE WHICH *
                               5541 *     REPRESENTS ONE LESS THAN THE CONSTANT SYMBOL LENGTH.  SYMBOL    *
                               5542 *     MATCHING IS BASED ON THE SIGN CR THE CONSTANT AND THE LETTER    *
                               5543 *     CHARACTER FOLLOWING THE '&' IDENTIFIER.                         *
                               5544 *                                                                     *
                               5545 *ATTRIBUTES                                                           *
                               5546 *   BNDATA IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               5547 *                                                                     *
                               5548 *CHARACTER CODE DEPENDENCY                                            *
                               5549 *   THE OPERATION OF THIS MODULE DEPENDS UPON AN INTERNAL REPRESENTA- *
                               5550 *   TION OF THE EXTERNAL CHARACTER SET WHICH IS EQUIVALENT TO THE     *
                               5551 *   ONE USED AT ASSEMBLY TIME.  THE CODING HAS BEEN ARRANGED SO THAT  *
                               5552 *   REDEFINITION OF CHARACTER CONSTANIS, BY REASSEMBLY, WILL RESULT IN*
                               5553 *   A CORRECT MODULE FOR THE NEW DEFINITIONS.                         *
                               5554 *                                                                     *
                               5555 *NOTES                                                                *
                               5556 *   ERROR PROCEDURES                                                  *
                               5557 *      N/A                                                            *
                               5558 *                                                                     *
                               5559 *   REGISTER USAGE                                                    *
                               5560 *      BOTH THE INDEX AND BASE REGISTERS ARE USER DURING EXECUTION.   *
                               5561 *                                                                     *
                               5562 *   SAVED/RESTORED AREAS                                              *
                               5563 *      N/A                                                            *
                               5564 *                                                                     *
                               5565 *   MODIFICATION CONSIDERATIONS                                       *
                               5566 *      BNDATA MUST RESIDE ON ONE SECTOR OR BE LINKED PROPERLY IF IT   *
                               5567 *      CROSSES A SECTOR BOUNDARY.  AS IT APPROACHES THE SECTOR        *
                               5568 *      LIMITATION, EXCEEDING THIS SIZE MUST BE A CONSIDERATION IN ANY *
                               5569 *      MODIFICATIONS.                                                 *
                               5570 *                                                                     *
                               5571 *   REQUIRED MODULES                                                  *
                               5572 *        @SYSEQ - COMMON SYSTEM EQUATES                               *
                               5573 *        @FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUE EQUATES  *
                               5574 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS               *
                               5575 *        @VMDEQ - VIRTUAL NEWRY DIRECTORY EQUATES                     *
                               5576 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES                         *
                               5577 *        @ERMEQ - ERROR MESSAGE EQUATES                               *
                               5578 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES                       *
                               5579 *        $B$EQU - COMPILER FIXED EQUATES                              *
                               5580 *        $B@EQU - COMPILER SYSTEM EQUATES                             *
                               5581 *                                                                     *
                               5582 *   OTHER                                                             *
                               5583 *      BNDATA IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               5584 ***********************************************************************

    1100                       5586       ORG   *,256,0                   BEGIN AT CORE PAGE BOUNDARY
                          1100 5587       USING *,@BR                     DEFINE USE ADDR FOR CORE PAGE
                               5588 *
                               5589 * ENTER BNDATA 'DATA' STATEMENT ROUTINE
                               5590 *
                          1100 5591 BNDATA EQU   *
                               5592 *
                               5593 * SET GET ROUTINE PARAMETER TO SKIP TO 1ST CHARACTER FOLLOWING 'DATA'
                               5594 *
    1100 3C 04 0873            5595 BND010 MVI   B$NUMC,B@LDAT             SET GET TO SKIP 'DATA'
                               5596 *
                               5597 * GENERATE A 'BRA' IMAGE IN VIRTUAL MEMORY
                               5598 *
    1104 D2 02 D1              5599 BND020 LA    BNDBRC(,@BR),@XR          LOAD CADDR OF 'BRA' INSTR
    1107 34 02 0A40            5600        ST    B$PCAD,@XR                SET PUT RTN VADDR FOR 'BRA'
                               5601 *
                               5602 * SET THE LENGTH PARAMETER IN PUT TO BE USED IN THE GENERATION OF THE
                               5603 * FOLLOWING INSTRUCTIONS: 'BRA', 'DCA' AND 'DDL'.
                               5604 *
    110B 3C 02 0A41            5605        MVI   B$PNBY,B@LCOP+B@LCVA-1    SET LENGTH PARM OF PUT RTN
    110F C0 87 093A            5606        B     B$PUTC                    LINK TO GENERATE 'BRA' PMC
                               5607 *
                               5608 * SAVE THE NEXT AVAILABLE VADDR IN THE BRANCH TABLE LINE NUMBER PARM
                               5609 *
    1113 0C 01 19F1 0A43       5610 BND030 MVC   B$BRLN,B$PVAD(@VADDR) SAVE THE NEXT AVAILABLE VADDR
                               5611 *
                               5612 * TEST THE CURRENT STATEMENT FOR BEING THE FIRST DATA STATEMENT
                               5613 *
    1119 3D 56 03CE            5614 BND040 CLI   $INLNO-1,B@DVC1           IF THIS IS NOT 1ST DATA STMT
    111D F2 02 09              5615        JNL   BND060                    * GO SET ADDR RESOLUTION COND
                               5616 *
                               5617 * IF THIS IS THE FIRST DCA ESTABLISH THE NEXT AVAILABLE VADDR AS THE
                               5618 * VALUE OF THE LINE NUMBER COMMUNICATION PARAMETER
                               5619 *
    1120 0C 01 03CF 0A43       5620 BND050 MVC   $INLNO,B$PVAD(@VADDR)     SAVE NEXT VADDR IN LN NO PARM
    1126 F2 87 0A              5621        J     BND070                    JUMP TO SET PUT RTN PARAMETERS
                               5622 *
                               5623 * SET UP ADDRESS RESOLUTION CONDITIONS TO LINK PREVIJUS ADOR DEFINITII
                               5624 * SEQUENCE WITH THE SEQUENCE FOR THE CURRENT STATEMENT
                               5625 *
    1129 0C 01 19EF 1B37       5626 BND060 MVC   B$BRVA,B$DLNK(@VADDR)     SET VADDR OF LAST DOL OPND AS
                               5627 *                                      * INPUT PARM
    112F C0 87 1996            5628        B     B$BTAB                    LINK TO RESOLVE BRANCH ADDRESS
                               5629 *
                               5630 * SET INPUT PARAMETERS FOR THE PUT ROUTINE
                               5631 *
    1133 D2 02 D4              5632 BND070 LA    BNDDAC(,@BR),@XR          LOAD CADDR OF 'DCA' INSTR
    1136 34 02 0A40            5633        ST    B$PCAD,@XR                SET PUT RTN VADDR FOR 'DCA'
                               5634 *
                               5635 * ADVANCE THE TEXT POINTER TO THE 1ST CHAR OF DATA LIST ITEM
                               5636 *
    113A C0 87 0867            5637 BND080 B     B$GETC                    LINK TO GET 1ST ITEM CHAR
                               5638 *
                               5639 * TEST FOR CHARACTER DATA
                               5640 *
    113E BD 7D 00              5641 BND090 CLI   B@CHAR(,@XR),B@SQUO       IF ELEMENT IS NOT CHAR DATA
    1141 F2 01 07              5642        JNE   BND100                    * GO TEST FOR INTERNAL CONSTANT
    1144 3C 1F 0A5F            5643        MVI   B$CTYP,B$CCON             SET CONSTANT RTN FOR CHAR DATA
    1148 F2 87 4A              5644        J     BND170                    GO PROCESS DATA CONSTANT
                               5645 *
                               5646 * TEST FOR INTERNAL CONSTANT DATA ELEMENT
                               5647 *
    114B 7C 4E DA              5648 BND100 MVI   BNDBKT+BNDBK0(,@BR),B@PLUS  SET SIGN OF CONSTANT TO PLUS
    114E BD 50 00              5649        CLI   B@CHAR(,@XR),B@ICON       IF CHAR IS NOT INTERNAL CON
    1151 F2 01 2C              5650        JNE   BND130                    * GO SET BUCKET SIGN BYTE
                               5651 *
                               5652 * SET 2ND BYTE OF COMPARE BUCKET AND SEARCH TABLE FOR INTERNAL CONSTANT
                               5653 *
    1154 C0 87 0867            5654 BND110 B     B$GETC                    LINK TO GET NEXT CHAR
    1158 6C 00 DB 00           5655        MVC   BNDBKT+BNDBK1(,@BR),B@CHAR(1,@XR)  SET 2ND BUCKET BYTE
    115C D2 02 D7              5656        LA    BNDTAB-BNDTEL(,@BR),@XR   LOAD TABLE BASE ADDR IN XR
    115F E2 02 05              5657 BND120 LA    BNDTEL(,@XR),@XR          INCREMENT POINTER TO NEXT ENTRY
    1162 6D 01 DB 01           5658        CLC   BNDBKT+BNDBK1(,@BR),BNDTB1(BNDBKL,@XR)  IF ICON NOT = ENT
    1166 D0 01 5F              5659        BNE   BND120(,@BR)              * GO SEARCH TABLE AGAIN
    1169 2C 00 0873 04         5660        MVC   B$NUMC,BNDTB4(1,@XR)      SET GET TO ADVANCE POINTER
    116E B5 02 03              5661        L     BNDTB3(,@XR),@XR          LOAD INTERNAL CON VADDR CADDR
    1171 6C 01 D6 00           5662        MVC   BNDDAO(,@BR),BNDICA(@VADDR,@XR)  SET 'DCA' INST OPERAND
    1175 C0 87 093A            5663        B     B$PUTC                    LINK TO GENERATE 'DCA' PMC
    1179 C0 87 0867            5664        B     B$GETC                    LINK TO GET CONSTANT DELIMITER
    117D F2 87 22              5665        J     BND190                    GO TEST FOR END OF DATA LIST
                               5666 *
                               5667 * MOVE CHAR TO 1ST BUCKET BYTE AND TEST FOR INTERNAL CONSTANT
                               5668 *
    1180 6C 00 DA 00           5669 BND130 MVC   BNDBKT+BNDBK0(,@BR),B@CHAR(1,@XR)  SET BUCKET SIGN BYTE
    1184 C0 87 0867            5670        B     B$GETC                    LINK TO GET NEXT CHAR
    1188 BD 50 00              5671        CLI   B@CHAR(,@XR),B@ICON       IF ELEMENT IS AN INTERNAL CON
    118B D0 81 54              5672        BE    BND110(,@BR)              * GO GET NEXT CHAR IN SEARCH TBL
                               5673 *
                               5674 * DISABLE BAGETC TO GET NEXT CHAR AND RESTORE TEXT POINTER
                               5675 *
    118E D2 02 DA              5676        LA    BNDBKT+BNDBK0(,@BR),@XR   RESTORE TEXT POINTER
    1191 3C 00 0873            5677        MVI   B$NUMC,B@GETS             DISABLE GET RTN TO GET CHARS
                               5678 *
                               5679 * CALL CONSTANT SCAN ROUTINE TO PROCESS THE DATA ELEMENT
                               5680 *
    1195 C0 87 0A46            5681 BND170 B     B$FCON                    LINK TO PROCESS DATA CONSTANT
                               5682 *
                               5683 * GENERATE A 'DCA' PMC WITH THE VADDR OF THE DATA CONSTANT AS OPERAND
                               5684 * IN VIRTUAL MEMORY
                               5685 *
    1199 4C 01 D6 1590         5686 BND180 MVC   BNDDAO(,@BR),B$BCKT(@VADDR)  SET DATA CON VADDR 'DCA' OPND
    119E C0 87 093A            5687        B     B$PUTC                    LINK TO GENERATE 'DCA' PMC
                               5688 *
                               5689 * TEST FOR A STATEMENT TERMINATOR
                               5690 *
    11A2 BD 1E 00              5691 BND190 CLI   B@CHAR(,@XR),B@EOST       IF THERE IS ANOTHER ELEMENT
    11A5 D0 01 3A              5692        BNE   BND080(,@BR)              * GO REPEAT PROCESSING
                               5693 *
                               5694 * GENERATE A SEQUENCE LINKAGE INSTR (DDL) IN VIRTUAL MEMORY
                               5695 *
    11A8 D2 02 D7              5696 BND200 LA    BNDDLC(,@BR),@XR          LOAD CADDR OF 'DDL' INSTR
    11AB 34 02 0A40            5697        ST    B$PCAD,@XR                SET PUT RTN VADDR FOR 'DDL'
    11AF C0 87 093A            5698        B     B$PUTC                    LINK TO GENERATE 'DDL' PMC
                               5699 *
                               5700 * SAVE THE VADDR OF THE OPERAND FIELD OF THE DDL INSTR
                               5701 *
    11B3 0C 01 1B37 0A43       5702 BND210 MVC   B$DLNK,B$PVAD(@VADDR)     SET PARM WITH NEXT VADDR
    11B9 1F 00 1B37 FA         5703        SLC   B$DLNK,BNDBN1(@VADDR-1,@BR)  ADJUST VADDR TO OPND OF 'DDL'
                               5704 *
                               5705 * SET UP ADDRESS RESOLUTION CONDITIONS FOR THE BYPASS BRANCH INSTR
                               5706 *
    11BE 0C 01 19EF 19F1       5707 BND220 MVC   B$BRVA,B$BRLN(@VADDR)     SET PARM WITH VADDR AFTER BRA
    11C4 1F 00 19EF FA         5708        SLC   B$BRVA,BNDBN1(@VADDR-1,@BR)  ADJUST VADDR TO OPND OF 'BRA'
    11C9 3A 07 071D            5709        SBN   B$NXSW,B$NXMK             SET SW FOR LINE RESOLUTION
                               5710 *
                               5711 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               5712 *
    11CD C0 87 0700            5713 BND230 B     B$DIST                    RETURN TO DISTRIBUTOR
```

```
                               5715 ***********************************************************************
                               5716 * 'DATA' STATEMENT ROUTINE PARAMETER AND STORAGE AREAS
                               5717 ***********************************************************************
                               5718 *
    11D1 46               11D1 5719 BNDBRC DC    AL(B@LCOP)(B@CBRA)        'BRA' INSTR OPCODE
    11D2 0000             11D3 5720 BNDBRO DC    XL(B@LCVA)'00'            'BRA' INSTR OPERAND
                               5721 *
    11D4 6A               11D4 5722 BNDDAC DC    AL(B@LCOP)(B@CDCA)        'DCA' INSTR OPCODE
    11D5                  11D6 5723 BNDDAO DS    CL(B@LCVA)                'DCA' INSTR OPERAND
                               5724 *
    11D7 6C               11D7 5725 BNDDLC DC    AL(B@LCOP)(B@CDDL)        'DDL' INSTR OPCODE
    11D8 0000             11D9 5726 BNDDLO DC    XL(B@LCVA)'00'            'DDL' INSTR OPERAND

                               5728 ***********************************************************************
                               5729 * 'DATA' STATEMENT INTERNAL CONSTANT TABLE
                               5730 ***********************************************************************
                               5731 *
                          0000 5732 BNDBK0 EQU   0                         LENGTH TO 1ST BUCKET BYTE
                          0001 5733 BNDBK1 EQU   1                         LENGTH TO 2ND BUCKET BYTE
                               5734 *
                          0005 5735 BNDTEL EQU   5                         LNG OF INTERNAL CON TBL ENTRY
                          0001 5736 BNDTB1 EQU   1                         DISP TO FIELD FOR BUCKET COMP
                          0003 5737 BNDTB3 EQU   3                         DISP TO CADDR OF CON VADDR
                          0004 5738 BNDTB4 EQU   4                         DISP TO CONSTANT LENGTH
                               5739 *
                          0000 5740 BNDICA EQU   0                         DISP FOR INTERNAL CON VADDR
                          0002 5741 BNDBKL EQU   2                         LNG OF INT CON COMP AREA
                               5742 *
                          11DA 5743 BNDBKT EQU   *                         INTERNAL CON COMPARE AREA ADDR
    11DA                  11DB 5744        DS    CL(BNDBKL)                COMPARE AREA FOR INTERNAL CON
                               5745 *
                          11DC 5746 BNDTAB EQU *
    11DC 4E               11DC 5747        DC    AL1(B@PLUS)               POSITIVE SIGNED INTERNAL CON
    11DD C5               11DD 5748        DC    AL1(B@CIEX)               2ND CHAR IN &E
    11DE 15A8             11DF 5749        DC    AL(@CADDR)(B$FVPE)        CADDR OF VADDR OF +&E
    11E0 01               11E0 5750        DC    AL1(B@LIEX-1)             LENGTH OF &E-1
                               5751 *
    11E1 4E               11E1 5752        DC    AL1(B@PLUS)               POSITIVE SIGNED INTERNAL CON
    11E2 D7               11E2 5753        DC    AL1(B@CIPI)               2ND CHAR IN &PI
    11E3 15AA             11E4 5754        DC    AL(@CADDR)(B$FVPP)        CADDR OF VADDR OF +$PI
    11E5 02               11E5 5755        DC    AL1(B@LIPI-1)             LENGTH OF &PI-1
                               5756 *
    11E6 4E               11E6 5757        DC    AL1(B@PLUS)               POSITIVE SIGNED INTERNAL CON
    11E7 E2               11E7 5758        DC    AL1(B@CIS2)               2ND CHAR IN &SQR2
    11E8 15AC             11E9 5759        DC    AL(@CADDR)(B$FVPS)        CADDR OF VADDR OF +&SQR2
    11EA 04               11EA 5760        DC    AL1(B@LIS2-1)             LENGTH OF &SQR2-1
                               5761 *
    11EB 60               11EB 5762        DC    AL1(B@MINS)               NEGATIVE SIGNED INTERNAL CON
    11EC C5               11EC 5763        DC    AL1(B@CIEX)               2ND CHAR IN &E
    11ED 15A2             11EE 5764        DC    AL(@CADDR)(B$FVME)        CADDR OF VADDR OF -&E
    11EF 01               11EF 5765        DC    AL1(B@LIEX-1)             LENGTH OF &E-1
                               5766 *
    11F0 60               11F0 5767        DC    AL1(B@MINS)               NEGATIVE SIGNED INTERNAL CON
    11F1 D7               11F1 5768        DC    AL1(B@CIPI)               2ND CHAR IN &PI
    11F2 15A4             11F3 5769        DC    AL(@CADDR)(B$FVMP)        CADDR OF VADDR OF -&PI
    11F4 02               11F4 5770        DC    AL1(B@LIPI-1)             LENGTH OF &PI-1
```

```
                               5771 *
    11F5 60               11F5 5772        DC    AL1(B@MINS)               NEGATIVE SIGNED INTERNAL CON
    11F6 E2               11F6 5773        DC    AL1(B@CIS2)               2ND CHAR IN &SQR2
    11F7 15A6             11F8 5774        DC    AL(@CADDR)(B$FVMS)        CADDR OF VADDR OF -&SQR2
    11F9 04               11F9 5775        DC    AL1(B@LIS2-1)             LENGTH OF &SQR2-1

                               5777 ***********************************************************************
                               5778 * 'DATA' STATEMENT ROUTINE CONSTANTS
                               5779 ***********************************************************************
                               5780 *
    11FA 01               11FA 5781 BNDBN1 DC    IL(@VADDR-1)'1'           BINARY 1
                               5782 *
                               5783 ***********************************************************************
                               5784 *
                               5785 * END OF 'DATA' STATEMENT ROUTINE CODING
                               5786 *
                               5788 ***********************************************************************
                               5789 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               5790 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               5791 *                                                                     *
                               5792 ***********************************************************************
                               5793 *STATUS                                                               *
                               5794 *   VERSION 1 MODIFICATION 0                                          *
                               5795 *                                                                     *
                               5796 *FUNCTION                                                             *
                               5797 *   BKFORX IS EXCUTED TO TRANSLATE FOR STATEMENTS AS THEY OCCUR IN A  *
                               5798 *   BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE    *
                               5799 *   PSEUDOCODE IN VIRTUAL MEMORY.                                     *
                               5800 *                                                                     *
                               5801 *INPUT                                                                *
                               5802 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               5803 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               5804 *     LEADING KEYWORD, FOR.                                           *
                               5805 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               5806 *     FIRST CHARACTER IN THE LEADING KEYWORD, FOR.                    *
                               5807 *   * FOR TABLE - CONTAINS TEN 4-BYTE ENTRIES, EACH CONTAINING THE    *
                               5808 *     VIRTUAL ADDRESSES OF A FOR-LOOP CONTROL VARIABLE AND OF THE     *
                               5809 *     NXT INSTRUCTION IN THE ASSOCIATED FOR OBJECT CODE SEQUENCE.     *
                               5810 *   * B$FTPT - CONTAINS THE CORE ADDRESS OF THE FIRST BYTE OF THE     *
                               5811 *     ENTRY LAST PLACED IN THE FOR TABLE.                             *
                               5812 *   * B$FTND - CONTAINS THE CORE ADDRESS OF THE FINAL BYTE IN THE     *
                               5813 *     FOR TABLE.                                                      *
                               5814 *                                                                     *
                               5815 *OUTPUT                                                               *
                               5816 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               5817 *     GENERATED BY BKFORX IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               5818 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               5819 *     SEQUENCES.                                                      *
                               5820 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               5821 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               5822 *   * FOR TABLE - UPDATED WITH THE CURRENT STATEMENT FOR-LOOP ENTRY.  *
                               5823 *     THE TABLE IS NOT AFFECTED WHEN AN ERROR OCCURS.                 *
                               5824 *   * B$FTPT - CONTAINS THE CORE ADDRESS OF THE FIRST BYTE IN THE     *
                               5825 *     FOR TABLE ENTRY GENERATED FOR THE CURRENT STATEMENT.  THIS      *
                               5826 *     IS NOT AFFECTED WHEN A COMPILER ERROR OCCURS.                   *
                               5827 *                                                                     *
                               5828 *EXTERNAL REFERENCES                                                  *
                               5829 *        B$GETC - (B$NUMC) - ENTRY TO BASIC RETRIEVAL ROUTINE.        *
                               5830 *        B$PUTC - (B$PFNC, B$PCAD, B$PNBY, B$PVAD, B$PCDL, B$PERC) -  *
                               5831 *                 ENTRY TO COMPILER VIRTUAL MEMORY OUTPUT ROUTINE.    *
                               5832 *        B$ECON - (B$BCKT) - ENTRY TO BASIC COMPILER CONSTANT ROUTINE.*
                               5833 *        B$SYKI - (B$BCKT) - ENTRY TO BASIC SYMBOL TRANSLATION        *
                               5834 *                 ROUTINE                                             *
                               5835 *        B$SCAN - ENTRY TO BASIC COMPILER ARITHMETIC EXPRESSION SCAN  *
                               5836 *                 ROUTINE                                             *
                               5837 *        B$FTPT - FOR TABLE POINTER TO LAST BYTE PLACED IN TABLE.     *
                               5838 *        $XIND1 - INDICATOR FOR LONG OR SHORT PRECISION.              *
                               5839 *        B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                *
                               5840 *                                                                     *
                               5841 *EXITS, NORMAL                                                        *
                               5842 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               5843 *                                                                     *
                               5844 *EXITS, ERROR                                                         *
                               5845 *   N/A                                                               *
                               5846 *                                                                     *
                               5847 *TABLES/WORK AREAS                                                    *
                               5848 *   * FOR TABLE - EXTERNAL TO BKFORX, THIS 'PUSH-DOWN' TABLE          *
                               5849 *     CONTAINS TEN 4-BYTE ENTRY LOCATIONS. THE FIRST ENTRY LOCATION   *
                               5850 *     IS ALWAYS SET TO ZEROS, AND IS USED TO GUARD AGAINST A TABLE    *
                               5851 *     REFERENCE WHEN THE TABLE IS EMPTY.  THE FOLLOWING NINE ENTRY    *
                               5852 *     LOCATIONS IN THE TABLE MAY EACH CONTAIN VIRTUAL ADDRESSES OF AN *
                               5853 *     UNFINISHED FOR-LOOP CONTROL VARIABLE AND ITS ASSOCIATED NXT     *
                               5854 *     INSTRUCTION, DEPENDING ON THE CURRENT LOOP NESTING DEPTH.       *
                               5855 *                                                                     *
                               5856 *ATTRIBUTES                                                           *
                               5857 *   BKFORX IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               5858 *                                                                     *
                               5859 *CHARACTER CODE DEPENDENCY                                            *
                               5860 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               5861 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               5862 *                                                                     *
                               5863 *NOTES                                                                *
                               5864 *   ERROR PROCEDURES                                                  *
                               5865 *      IF MORE THAN NINE LEVELS OF FOR-LOOP NESTING ARE ATTEMPTED,    *
                               5866 *      THE FOR TABLE STATUS REMAINS UNCHANGED AND THE ERROR CONDITION *
                               5867 *      CODE FOR MORE THAN 9 NESTED FOR/NXT LOOPS, IS LOGGED IN        *
                               5868 *      VIRTUAL MEMORY USING OUTPUT ROUTINE BBPUTC, BKFORX EXECUTION   *
                               5869 *      IS OTHERWISE UNAFFECTED.                                       *
                               5870 *                                                                     *
                               5871 *   REGISTER USAGE                                                    *
                               5872 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               5873 *                                                                     *
                               5874 *   SAVED/RESTORED AREAS                                              *
                               5875 *      N/A                                                            *
                               5876 *                                                                     *
                               5877 *   MODIFICATION CONSIDERATIONS                                       *
                               5878 *      BKFORX RESIDES ON ONE SECTOR AND MUST NOT EXCEED ITS BOUNDARY. *
                               5879 *      ANY MODIFICATIONS MUST CONSIDER THIS SIZE LIMITATION.          *
                               5880 *                                                                     *
                               5881 * REQUIRED MODULES                                                    *
                               5882 *        @SYSEQ - COMMON SYSTEM EQUATES                               *
                               5883 *        @FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUE EQUATES  *
                               5884 *        @CANEQ - COMMON CORE LOCATIONS                               *
                               5885 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES                    *
                               5886 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES                         *
                               5887 *        @ERMEQ - ERROR MESSAGE EQUATES                               *
                               5888 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES                       *
                               5889 *        $B$EQU - COMPILER FIXED EQUATES                              *
                               5890 *        $B@EQU - COMPILER SYSTEM EQUATES                             *
                               5891 *                                                                     *
                               5892 *   OTHER                                                             *
                               5893 *      BKFORX IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               5894 ***********************************************************************

    1200                       5896       ORG   *,256,0                   BEGIN AT CORE PAGE BOUNDARY
                          1200 5897       USING *,@BR                     DEFINE BASE ADDR FOR CORE PAGE
                               5898 *
                               5899 * ENTER BKFORX - FOR STATEMENT ROUTINE
                               5900 *
                          1200 5901 BKFORX EQU   *                         BKFORX ENTRY POINT
                               5902 *
                               5903 * SET INPUT PARAMETER TO SKIP KEYWORD 'FOR'
                               5904 *
    1200 3C 03 0873            5905 BKF010 MVI   B$NUMC,B@LKFR             SET PARAMETER TO SKIP 'FOR'
    1204 C0 87 0867            5906        B     B$GETC                    LINK TO ADVANCE POINTER
                               5907 *
                               5908 * STORE CONTROL VARIABLE VIRTUAL ADDRESS
                               5909 *
    1208 C0 87 0DBC            5910 BKF020 B     B$SYMB                    LINK TO GET VADDR OF CTRL VAR
    120C 4C 01 BA 1590         5911        MVC   BKFOFO(,@BR),B$BCKT(@VADDR)  SAVE CTRL VARIABLE VADDR
                               5912 *
                               5913 * GENERATE PMC'S FOR INITIAL AND FINAL CONTROL VALUES
                               5914 *
    1211 C0 87 1514            5915 BKF030 B     B$SCAN                    LINK TO PROCESS INITIAL VALUE
    1215 3C 00 0873            5916        MVI   B$NUMC,B@LKTO-2           SET GET RTN NOT TO SKIP CHAR
    1219 C0 87 0867            5917        B     B$GETC                    LINK NOT TO SKIP CHARACTERS
    121D 3C 01 1BAC            5918        MVI   B$SSTA,@B1                SET SW TO ALLOW 'STEP' PARM
    1221 C0 87 1514            5919        B     B$SCAN                    LINK TO PROCESS FINAL VALUE
    1225 3C 00 1BAC            5920        MVI   B$SSTA,@ZERO              SET SWITCH OFF FOR 'STEP'
    1229 BD 1E 00              5921        CLI   B@CHAR(,@XR),@EOS         IF INCREMENT NOT SPECIFIED
    122C F2 81 0F              5922        JE    BKF050                    * SKIP TO SET INCREMENT = 1
                               5923 *
                               5924 * GENERATE PMC FOR SPECIFIED INCREMENT VALUE
                               5925 *
    122F 3C 02 0873            5926 BKF040 MVI   B$NUMC,BKFLSP+1           SET PARAMETER TO SKIP 'EP'
    1233 C0 87 0867            5927        B     B$GETC                    LINK TO ADVANCE POINTER
    1237 C0 87 1514            5928        B     B$SCAN                    LINK TO PROCESS INCREMENT
    123B F2 87 1F              5929        J     BKF060                    JUMP TO TEST PRECISION
                               5930 *
                               5931 * GENERATE PMC FOR DEFAULT INCREMENT VALUE
                               5932 *
    123E D2 02 E8              5933 BKF050 LA    BKFOC1(,@BR),@XR          LOAD CADDR OF DECIMAL ONE
    1241 3C 00 0873            5934        MVI   B$NUMC,B@GETS             SET GETC NOT TO GET NEXT CHAR
    1245 C0 87 0A46            5935        B     B$FCON                    LINK TO GET VADDR OF ONE
    1249 4C 01 E3 1590         5936        MVC   BKFOSO(,@BR),B$BCKT(@VADDR)  MOVE VADDR OF 1 TO PMC STRING
    124E D2 02 E1              5937        LA    BKFOSC(,@BR),@XR          LOAD CADDR OF 'STF' INSTR
    1251 34 02 0A40            5938        ST    B$PCAD,@XR                SET PUT RTN FOR VADDR OF 'STF'
    1255 3C 02 0A41            5939        MVI   B$PNBY,B@LSTF-1           SET PUT RTN FOR LENGTH OF 'STF'
    1259 C0 87 093A            5940        B     B$PUTC                    LINK TO WRITE INCREMENT PMC
                               5941 *
                               5942 * TEST FOR PRECISION BEFORE GENERATING FOR/NXT PMC SEQUENCE
                               5943 *
    125D 38 40 03D0            5944 BKF060 TBN   $XIND1,$XPREC             IF PRECISION IS STANDARD
    1261 F2 90 06              5945        JF    BKF070                    * SKIP TO GENERATE FOR/NEXT PMC
    1264 7C 27 E0              5946        MVI   BKFOFA(,@BR),BKFLLP       SET LENGTH FOR LONG PRECISION
    1267 7C 20 BF              5947        MVI   BKFDAN(,@BR),2*B@LELP     SET 'DWA' OPERAND FOR LONG PREC
                               5948 *
                               5949 * GENERATE FOR/NXT LOOP CONTROL PMC SEQUENCE
                               5950 *
    126A 1C 00 0A41 E0         5951 BKF070 MVC   B$PNBY,BKFOFA(1,@BR)      SET PUT RTN FOR FOR LOOP LNG
    126F D2 02 B8              5952        LA    BKFOFC(,@BR),@XR          LOAD CADDR FOR FOR LOOP INSTR
    1272 34 02 0A40            5953        ST    B$PCAD,@XR                SET PUT BIN - FOR LOOP VADDR
    1276 C0 87 093A            5954        B     B$PUTC                    LINK TO GENERATE FOR/NXT STRING
                               5955 *
                               5956 * INCREMENT 'FOR' TABLE POINTER FOR CURRENT NEST DEPTH LEVEL
                               5957 *
    127A 35 02 1B0D            5958 BKF080 L     B$FTPT,@XR                LOAD THE 'FOR' TABLE POINTER
    127E E2 02 04              5959        LA    B@LFRT(,@XR),@XR          INCR POINTER TO NEXT LEVEL
    1281 34 02 1B0D            5960        ST    B$FTPT,@XR                STORE THE 'FOR' TABLE POINTER
    1285 0D 01 1B0D 1B0B       5961        CLC   B$FTPT,B$FTND(@CADDR)     IF NESTING LIMIT NOT EXCEEDED
    128B F2 04 14              5962        JNH   BKF100                    * SKIP TO STORE CURRENT LEVEL
                               5963 *
                               5964 * GENERATE ERROR CODE FOR 'FOR' NESTING DEPTH EXCEPTION
                               5965 *
    128E 1F 01 1B0D E5         5966 BKF090 SLC   B$FTPT,BKFOTL(@CADDR,@BR) SET 'FOR' PT TO ORIGINAL ENTRY
    1293 3C 33 094E            5967        MVI   B$PFNC,B$PFAE             SET PUT RTN FOR ERROR OUTPUT
    1297 3C AD 0A39            5968        MVI   B$PERC,@@E608             SET ERROR CODE
    129B C0 87 093A            5969        B     B$PUTC                    LINK TO OUTPUT CHARACTER STRING
    129F F2 87 12              5970        J     BKF120                    JUMP TO BKFORX EXIT
                               5971 *
                               5972 * STORE CURRENT LOOP VALUES IN FOR TABLE
                               5973 *
    12A2 9C 01 01 BA           5974 BKF100 MVC   BKFOCV(,@XR),BKFOFO(@VADDR,@BR)  STORE CTRL VARIABLE VADDR
    12A6 8C 01 03 0A43         5975        MVC   BKFOND(,@XR),B$PVAD(@VADDR)    MOVE NEXT PMC VADDR TO TBL
    12AB 8F 00 03 09D3         5976        SLC   BKFOND(,@XR),B$PCDL(@VADDR-1)  SUBTRACT LENGTH OF LIST PMC
    12B0 9E 01 03 EA           5977        ALC   BKFOND(,@XR),BKFOX3(@VADDR,@BR)  SET NEXT PMC VADDR IN TBL
                               5978 *
                               5979 * RETURN CONTROL TO THE DISTRIBUTOR
                               5980 *
    12B4 C0 87 0700            5981 BKF120 B     B$DIST                    RETURN TO DISTRIBUTOR

                               5983 ***********************************************************************
                               5984 * 'FOR' STATEMENT ROUTINE PMC AND STORAGE PARAMETERS
                               5985 ***********************************************************************
                               5986 *
    12B8 4E               12B8 5987 BKFOFC DC    AL(B@LCOP)(B@CFOR)        'BEGIN LOOP' OPCODE
    12B9                  12BA 5988 BKFOFO DS    CL(B@LCVA)                CONTROL VARIABLE VADDR
                               5989 *
    12BB 50               12BB 5990 BKFONC DC    AL(B@LCOP)(B@CNXT)        'CONTINUE LOOP' OPCODE
    12BC 0000             12BD 5991 BKFONO DC    XL(@VADDR)'00'            LOOP EXIT BRANCH ADDR FIELD
                               5992 *
    12BE 6E               12BE 5993 BKFDAC DC    AL(B@LCOP)(B@CDWA)        'DWA' INSTRUCTION OPCODE
    12BF                  12BF 5994 BKFDAN DS    CL(B@LCNN)                'DWA' INSTRUCTION OPERAND
    12BF                       5995        ORG   BKFDAN                    INITIALIZE 'DMA' OPERAND FOR
    12BF 10               12BF 5996        DC    AL(B@LCNN)(2*B@LESP)      * STANDARD PREC UNPACKED FLT PT
    12C0 0000000000000000 12DF 5997 BKFOPR DC    XL(2*B@LELP)'00'          LOOP CONTROL PARAMETERS FIELD
    12E0                  12E0 5998 BKFOFA DS    CL1                       'FOR LOOP' PMC LENGTH - 1
    12E0                       5999        ORG   BKFOFA                    LENGTH SET FOR SHORT PRECISION
    12E0 17               12E0 6000        DC    AL1(B@LFOR+B@LNXT+B@LDWA+2*B@LESP-1) CHANGE FOR LENGTH PR
                               6001 *
    12E1 20               12E1 6002 BKFOSC DC    AL(B@LCOP)(B@CSTF)        STACK FLT VALUE OPCODE
    12E2                  12E3 6003 BKFOSO DS    CL(B@LCVA)                STACK FLT VALUE OPERAND

                               6005 ***********************************************************************
                               6006 * 'FOR' STATEMENT ROUTINE CONSTANTS AND EQUATES
                               6007 ***********************************************************************
                               6008 *
                               6009 * CONSTANTS
                               6010 *
    12E4 0004             12E5 6011 BKFOTL DC    AL(@CADDR)(B@LFRT)        'FOR' TABLE ENTRY LENGTH
    12E6 0002             12E7 6012 BKFBN2 DC    IL(@VADDR)'2'             BINARY INTEGER *2
    12E8 F1               12E8 6013 BKFOC1 DC    CL1'1'                    EBCDIC 1
    12E9 0003             12EA 6014 BKFOX3 DC    AL(@VADDR)(B@LFOR)        BINARY INTEGER *3
                               6015 *
                               6016 * EQUATES
                               6017 *
                          0027 6018 BKFLLP EQU   B@LFOR+B@LNXT+B@LDWA+2*B@LELP-1  LONG PREC 'FOR' SEQ LNG
                          0001 6019 BKFLSP EQU   1                         LENGTH OF 'STEP'-2
                          0001 6020 BKFOCV EQU   1                         DISP FOR 'FOR' TABLE CTRL VAR.
                          0003 6021 BKFOND EQU   3                         DISP FOR 'FOR' TABLE NXT VADDR
                               6022 *
                               6023 ***********************************************************************
                               6024 *
                               6025 * END OF 'FOR' STATEMENT ROUTINE CODING
                               6026 *
                               6028 ***********************************************************************
                               6029 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               6030 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               6031 *                                                                     *
                               6032 ***********************************************************************
                               6033 *STATUS                                                               *
                               6034 *   VERSION 1 MODIFICATION 0                                          *
                               6035 *                                                                     *
                               6036 *FUNCTION                                                             *
                               6037 *   BXDPRT IS EXECUTED TO TRANSLATE PRINT STATEMENTS AS THEY OCCUR,   *
                               6038 *   A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE  *
                               6039 *   PSEUDOCODE IN VIRTUAL MEMORY.                                     *
                               6040 *                                                                     *
                               6041 *ENTRY POINTS                                                         *
                               6042 *   BXDPRT HAS ONLY ONE ENTRY POINT:                                  *
                               6043 *        BXDPRT - TRANSLATE PRINT STATEMENT                           *
                               6044 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               6045 *        B     BXDPRT                                                 *
                               6046 *                                                                     *
                               6047 *INPUT                                                                *
                               6048 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING, *
                               6049 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               6050 *     LEADING KEYWORD, PRINT.                                         *
                               6051 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST,*
                               6052 *     CHARACTER IN THE LEADING KEYWORD, PRINT.                        *
                               6053 *                                                                     *
                               6054 *OUTPUT                                                               *
                               6055 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               6056 *     GENERATED BY BXDPRT IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               6057 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               6058 *     SEQUENCES.                                                      *
                               6059 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               6060 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               6061 *                                                                     *
                               6062 *EXTERNAL REFERENCES                                                  *
                               6063 *        B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.*
                               6064 *        B$PUTC - (B$PCAD, B$PNBY, B$ARSW) - ENTRY TO COMPILER        *
                               6065 *                 VIRTUAL MEMORY OUTPUT ROUTINE.                      *
                               6066 *        B$FCON - (B$CTYP, B$BCKT, B$@PCT) - ENTRY TO BASIC COMPILER  *
                               6067 *                 CONSTANT ROUTINE.                                   *
                               6068 *        B$CSCN - (B$CSSW) - ENTRY TO BASIC COMPILER CHARACTER SCAN   *
                               6069 *                 ROUTINE.                                            *
                               6070 *        B$SCAN - ENTRY TO BASIC COMPILER ARITHMETIC EXPRESSION SCAN  *
                               6071 *                 ROUTINE.                                            *
                               6072 *        B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                *
                               6073 *                                                                     *
                               6074 *EXITS, NORMAL                                                        *
                               6075 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               6076 *                                                                     *
                               6077 *EXITS, ERROR                                                         *
                               6078 *   N/A                                                               *
                               6079 *                                                                     *
                               6080 *TABLES/WORK AREAS                                                    *
                               6081 *   * PRINT CODE TABLE - INTERNAL TO BXDPRT, THIS TABLE CONTAINS PRS  *
                               6082 *     INSTRUCTION CODES ASSOCIATED WITH PRINT LIST DELIMITERS.        *
                               6083 *     DELIMITERS REQUIRE DIFFERENT CODES DEPENDING ON THE CLASS OF    *
                               6084 *     THE PRECEDING LIST ELEMENT.                                     *
                               6085 *                                                                     *
                               6086 *ATTRIBUTES                                                           *
                               6087 *   * BXDPRT IS NATURALLY RELOCATABLE AND REUSABLE.                   *
                               6088 *                                                                     *
                               6089 *CHARACTER CODE DEPENDENCY                                            *
                               6090 *   THE OPERATION OF THIS MODULE DEPENDS UPON AN INTERNAL REPRESENTA- *
                               6091 *   TION OF THE EXTERNAL CHARACTER SET WHICH IS EQUIVALENT TO THE ONE *
                               6092 *   USED AT ASSEMBLY TIME.  THE CODING HAS BEEN ARRANGED SO THAT      *
                               6093 *   REDEFINITION OF CHARACTER CONSTANTS, BY REASSEMBLY, WILL RESULT   *
                               6094 *   IN A CORRECT MODULE FOR THE NEW DEFINITIONS.                      *
                               6095 *                                                                     *
                               6096 *NOTES                                                                *
                               6097 *   ERROR PROCEDURES                                                  *
                               6098 *      N/A                                                            *
                               6099 *                                                                     *
                               6100 *   REGISTER USAGE                                                    *
                               6101 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               6102 *                                                                     *
                               6103 *   SAVED/RESTORED AREAS                                              *
                               6104 *      N/A                                                            *
                               6105 *                                                                     *
                               6106 *   MODIFICATION CONSIDERATIONS                                       *
                               6107 *      BXDPRT RESIDES ON ONE SECTOR AND HAS ONLY 9 BYTES AVAILABLE    *
                               6108 *      FOR MODIFICATION.  IF A SIGNIFICANT CHANGE IN SIZE IS REQUIRED *
                               6109 *      LINKAGE WOULD HAVE TO BE ESTABLISHED TO A SECOND SECTOR.       *
                               6110 *                                                                     *
                               6111 *   REQUIRED MODULES                                                  *
                               6112 *        @SYSEQ - COMMON SYSTEM EQUATES.                              *
                               6113 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.   *
                               6114 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.      *
                               6115 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               6116 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               6117 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               6118 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               6119 *        $B$EQU - COMPILER FIXED EQUATES.                             *
                               6120 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               6121 *                                                                     *
                               6122 *   OTHER                                                             *
                               6123 *      BXDPRT IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS       *
                               6124 ***********************************************************************

    1300                       6126        ORG   *,256,0                   BEGIN AT CORE PAGE BOUNDARY
                          1300 6127        USING *,@BR                     DEFINE BASE ADDR FOR CORE PAGE
                               6128 *
                               6129 * ENTER BXDPRT - 'PRINT' STATEMENT ROUTINE
                               6130 *
                          1300 6131 BXDPRT EQU   *
                               6132 *
                               6133 * SKIP TO LETTER 'T' IN KEYWORD 'PRINT'
                               6134 *
    1300 3C 04 0873            6135 BXD010 MVI   B$NUMC,B@LPRT-1           SET GET RTN TO SKIP TO 'T'
    1304 C0 87 0867            6136        B     B$GETC                    LINK TO ADVANCE POINTER
                               6137 *
                               6138 * INITIALIZE THE SUBROUTINE
                               6139 *
    1308 7B 07 A8              6140 BXD020 SBF   BXDRS1(,@BR),BXDRM1       SET PRINT LIST SWITCH OFF
                               6141 *
                               6142 * SET THE 'PRINT AND SPACE' CODE TABLE MODE TO 1
                               6143 *
    130B 7C C8 2D              6144 BXD030 MVI   BXD090+@D1(,@BR),BXDMD1-BXDPRT-BXDLTH  SET NODE TO 1
    130E 7C 5B D6              6145        MVI   BXDM14(,@BR),BXD180-BXDPRT  SET MODE 1 BRANCH ADDRESS
                               6146 *
                               6147 * ATTEMPT TO GENERATE PMC FOR ARITH EXPR BY CALLING ARITH SCAN RTN
                               6148 *
    1311 C0 87 1514            6149 BXD040 B     B$SCAN                    LINK TO ATTEMPT PMC GENERATION
                               6150 *
                               6151 * TEST LIST ELEMENT FOR BEING A CHARACTER VARIABLE
                               6152 *
    1315 38 07 14BC            6153 BXD050 TBN   B$CSSW,B$CSMK             TEST FOR CHAR VARIABLE
                               6154 *
                               6155 * IF ELEMENT IS A CHARACTER VARIABLE CALL THE CHAR SCAN ROUTINE
                               6156 *
    1319 C0 10 14B0            6157 BXD060 BT    B$CSCN                    LINK TO PROCESS CHAR VARIABLE
                               6158 *
                               6159 * TEST FOR ANY PMC HAVING BEEN GENERATED FOR THIS ELEMENT
                               6160 *
    131D 38 01 0A45            6161 BXD065 TBN   B$ARSW,B$ARMK             IF PMC'S GENERATED
    1321 F2 10 03              6162        JT    BXD080                    * GO SEARCH TABLE
                               6163 *
                               6164 * SET THE 'PRINT AND SPACE' CODE TABLE MODE TO TWO
                               6165 *
    1324 7C D4 2D              6166 BXD070 MVI   BXD090+@D1(,@BR),BXDMD2-BXDPRT-BXDLTH  SET MODE TO 2
                               6167 *
                               6168 * SEARCH THE 'PRINT AND SPACE' CODE TABLE FOR PRS CODE AND BRANCH ADDR
                               6169 * FOR LIST DELIMITER
                               6170 *
    1327 6C 00 32 00           6171 BXD080 MVC   BXD100+@Q(,@BR),B@CHAR(1,@XR)  SAVE TEXT CHARACTER
    132B D2 02 00              6172 BXD090 LA    *-*(,@BR),@XR             LOAD ADDR OF PRS TABLE MODE
                               6173 *
    132E E2 02 03              6174 BXD095 LA    BXDLTH(,@XR),@XR          INCREMENT TABLE BY ENTRY LENGTH
    1331 BD 00 00              6175 BXD100 CLI   BXDDP0(,@XR),*-*          IF LIST AND TABLE DELIMITERS
    1334 D0 81 3D              6176        BE    BXD110(,@BR)              * GO SET CODE AND BRANCH ADDR
    1337 BD 00 00              6177        CLI   BXDDP0(,@XR),BXDDUM       IF DELIMITER IS NOT DUMMY ENTRY
    133A D0 01 2E              6178        BNE   BXD095(,@BR)              BRANCH TO NEXT COMPARE
                               6179 *
                               6180 * SET PRS CODE AND BRANCH TO THE ADDRESS LISTED IN THE TABLE
                               6181 *
    133D 6C 00 F3 01           6182 BXD110 MVC   BXDPRO(,@BR),BXDDP1(1,@XR)  SET PRS CODE IN PBS OPERAND
    1341 6C 00 47 02           6183        MVC   BXD120+@D1(,@BR),BXDDP2(1,@XR)  SET BRANCH DISPLACEMENT
    1345 D0 87 00              6184 BXD120 B     *-*(,@BR)                 BRANCH TO ADDR ACCORDING TO TBL
                               6185 *
                               6186 * GENERATE THE 'PRS' PMC INSTRUCTION IN VIRTUAL MEMORY
                               6187 *
    1348 D0 87 B1              6188 BXD140 B     BXD300(,@BR)              LINK TO GENERATE 'PRS' PMC
                               6189 *
                               6190 * SET THE PRINT LIST SWITCH ON
                               6191 *
    134B 7A 07 A8              6192 BXD150 SBN   BXDRS1(,@BR),BXDRM1       SET PRINT LIST SWITCH ON
    134E D0 87 0B              6193        B     BXD030(,@BR)              BRANCH TO PROCESS NEXT ELEMENT
                               6194 *
                               6195 * GENERATE THE 'PRS' INSTRUCTION IN VIRTUAL MEMORY
                               6196 *
    1351 D0 87 B1              6197 BXD160 B     BXD300(,@BR)              LINK TO GENERATE 'PRS' PMC
                               6198 *
                               6199 * DISABLE THE GET ROUTINE FOR THE NEXT EXECUTION OF ARM SCAN ROUTINE
                               6200 *
    1354 3C 00 0873            6201 BXD170 MVI   B$NUMC,B@GETS             SET GET RTN NOT TO SKIP CHAR
    1358 D0 87 0B              6202        B     BXD030(,@BR)              BRANCH TO PROCESS NEXT ELEMENT
                               6203 *
                               6204 * GENERATE THE 'PRS' INSTRUCTION IN VIRTUAL MEMORY
                               6205 *
    135B D0 87 B1              6206 BXD180 B     BXD300(,@BR)              LINK TO GENERATE 'PRS' PMC
                               6207 *
                               6208 * CALL CONSTANT ROUTINE TO GENERATE CHARACTER STRING IN V.M.
                               6209 *
    135E 3C 1B 0A5F            6210 BXD190 MVI   B$CTYP,B$SCON             SET CON RTN FOR CHAR STRING
    1362 35 02 0878            6211        L     B$GPTR,@XR                RESTORE TEXT POINTER
    1366 C0 87 0A46            6212        B     B$FCON                    LINK TO GENERATE CHAR STRING
                               6213 *
                               6214 * TEST FOR THIS ELEMENT BEING A NULL CHARACTER STRING
                               6215 *
    136A 7C E0 2D              6216 BXD200 MVI   BXD090+@D1(,@BR),BXDMD3-BXDPRT-BXDLTH  SET MODE TO 3
    136D 3D 00 0CA8            6217        CLI   B$CPCT,@ZERO              IF THIS IS A NULL STRING
    1371 D0 81 27              6218        BE    BXD080(,@BR)              * GO SEARCH 'PRS' BRANCH TABLE
                               6219 *
                               6220 * SET 'PRINT AND SPACE' CODE TABLE MODE TO FOUR
                               6221 *
    1374 7C C8 2D              6222 BXD210 MVI   BXD090+@D1(,@BR),BXDMD1-BXDPRT-BXDLTH  SET MODE TO 4
    1377 7C 51 D6              6223        MVI   BXDM14(,@BR),BXD160-BXDPRT  SET MODE 4 BRANCH ADDRESS
    137A 7C 08 F3              6224        MVI   BXDPRO(,@BR),B@PRRL       SET CODE FOR PRINT LONG
    137D BD 6B 00              6225        CLI   B@CHAR(,@XR),B@CMMA       IF DELIMITER IS A COMMA
    1380 D0 81 B1              6226        BE    BXD300(,@BR)              * LINK TO GENERATE FMC
    1383 7C 01 F3              6227        MVI   BXDPRO(,@BR),B@PRPN       SET CODE FOR PRINT AND NO SPACE
                               6228 *
                               6229 * MOVE THE VADDR OF THE 1ST STRING SEGMENT TO AN 'STC' INSTRUCTION
                               6230 *
    1386 4C 01 F6 1590         6231 BXD220 MVC   BXDSTO(,@BR),B$BCKT(@VADDR) MOVE VADDR OF 1ST CON TO OPKD
                               6232 *
                               6233 * GENERATE THE 'STC' INSTRUCTION IN VIRTUAL MEMORY
                               6234 *
    138B D2 02 F4              6235 BXD230 LA    BXDSTC(,@BR),@XR          LOAD CADDR OF 'STC' INSTR
    138E 3C 02 0A41            6236        MVI   B$PNBY,B@LSTC-1           SET PUT RTN LNG PARM FOR 'STC'
    1392 D0 87 B8              6237        B     BXD310(,@BR)              LINK TO GENERATE 'STC' PMC
                               6238 *
                               6239 * TEST FOR THE EXISTENCE OF ANOTHER SEGMENT IN THE CHARACTER STRING
                               6240 *
    1395 1F 00 0CA8 EF         6241 BXD240 SLC   B$CPCT,BXDBN1(1,@BR)      IF NO OTHER SEGMENTS EXIST
    139A D0 81 27              6242        BE    BXD080(,@BR)              * GO SEARCH PRS TABLE
                               6243 *
                               6244 * IF ANOTHER SEGMENT DOES EXIST GENERATE THE 'PRS' PMC IN V.M.
                               6245 *
    139D D0 87 B1              6246 BXD250 B     BXD300(,@BR)              LINK TO GENERATE 'PRS' PMC
                               6247 *
                               6248 * SUBTRACT THE LENGTH OF A STRING SEGMENT FROM THE 'STC' OPERAND
                               6249 *
    13A0 5F 01 F6 F1           6250 BXD260 SLC   BXDSTO(,@BR),BXDSUB(@VADDR,@BR)  SUB SEGMENT LENGTH
    13A4 D0 87 8B              6251        B     BXD230(,@BR)              BRANCH TO GENERATE 'STC' PMC
```

```
                               6252 *
                               6253 * TEST FOR THE PRINT LIST WNITCH BEING ON
                               6254 *
    13A7 F2 00 03              6255 BXD270 JC    BXD290,*-*                IF LIST SWITCH IS ON
    13A8                       6256        ORG   BXD270+@Q                 * GO BRANCH TO DIST
    13A8 80               13A8 6257        DC    AL1(@NOP)                 IF LIST SWITCH IS OFF
    13AA                       6258        ORG   BXD270+@INST3             * GO BRANCH TO GENERATE PMC
                               6259 *
                               6260 * GENERATE THE 'PRS' INSTRUCTION IN VIRTUAL MEMORY
                               6261 *
    13AA D0 87 B1              6262 BXD280 B     BXD300(,@BR)              LINK TO GENERATE 'PRS' PMC
                               6263 *
                               6264 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               6265 *
    13AD C0 87 0700            6266 BXD290 B     B$DIST                    RETURN TO THE DISTRIBUTOR
```

```
                               6268 ***********************************************************************
                               6269 * SUBROUTINE FOR THE GENERATION OF PSEUDOCODE IN VIRTUAL MEMORY -     *
                               6270 * * THE ENTIRE ROUTINE IS USED TO GENERATE THE 'PRS' INSTRUCTION      *
                               6271 * * AND A SECOND ENTRY POINT ALLOWS THE ROUTINE TO COMPLETE THE       *
                               6272 * * GENERATION OF THE 'STC' INSTRUCTION                               *
                               6273 ***********************************************************************
                               6274 *
                               6275 * ENTER GENERATE SUBROUTINE - FOR 'PRS' GENERATION
                               6276 *
    13B1 D2 02 F2              6277 BXD300 LA    BXDPRC(,@BR),@XR          LOAD CADDR OF 'PRS' INSTC
    13B4 3C 01 0A41            6278        MVI   B$PNBY,B@LPRS-1           SET PUT RTN LENGTH PARM
                               6279 *
                               6280 * SECONDARY ENTRY POINT TO GENERATE SUBROUTINE - FOR 'STC' GENERATION
                               6281 *
    13B8 74 08 CA              6282 BXD310 ST    BXD320+@OP1(,@BR),@ARR    STORE RETURN ADDRESS
    13BB 34 02 0A40            6283        ST    B$PCAD,@XR                SET PUT RTN VADDR PARM
    13BF C0 87 093A            6284        B     B$PUTC                    LINK TO GENERATE PMC
    13C3 35 02 0878            6285        L     B$GPTR,@XR                RESTORE TEXT POINTER
    13C7 C0 87 0000            6286 BXD320 B     *-*                       BRANCH TO RETURN ADDRESS

                               6288 ***********************************************************************
                               6289 * PRINT STATEMENT 'PRINT AND SPACE' CODE TABLE
                               6290 ***********************************************************************
                               6291 *
                          0003 6292 BXDLTH EQU   3                         LENGTH OF CODE TABLE ENTRY
                          0004 6293 BXDROM EQU   4                         NUMBER OF ENTRIES PER MODE
                               6294 *
                          0000 6295 BXDDUM EQU   X'00'                     TABLE DUMMY COMPARE

                               6297 ***********************************************************************
                               6298 * PRINT CODE TABLE MODE FOR LIST ELEMENT AND EXPRESSION PROCESSING
                               6299 ***********************************************************************
                               6300 *
                          13CB 6301 BXDMD1 EQU   *                         PRS TABLE - MODES 1 AND 4
    13CB 6B               13CB 6302        DC    AL1(B@CMMA)               DELIMITER - COMMA
    13CC 02               13CC 6303        DC    AL1(B@PRPL)               PRINT AND SPACE TO LONG ZONE
    13CD 48               13CD 6304        DC    AL1(BXD140-BXDPRT)        BRANCH ADDRESS
                               6305 *
    13CE 5E               13CE 6306        DC    AL1(B@SCLN)               DELIMITER - SEMI-COLON
    13CF 03               13CF 6307        DC    AL1(B@PRPS)               PRINT AND SPACE TO SHORT ZONE
    13D0 48               13D0 6308        DC    AL1(BXD140-BXDPRT)        BRANCH ADDRESS
                               6309 *
    13D1 1E               13D1 6310        DC    AL1(B@EOST)               DELIMITER - END OF STATEMENT
    13D2 04               13D2 6311        DC    AL1(B@PRPR)               PRINT AND RETURN CARRIAGE
    13D3 AA               13D3 6312        DC    AL1(BXD280-BXDPRT)        BRANCH ADDRESS
                               6313 *
    13D4 00               13D4 6314        DC    AL1(BXDDUM)               DELIMITER - NOT , OR ; OR CR
    13D5 01               13D5 6315        DC    AL1(B@PRPN)               PRINT AND NO SPACE
    13D6                  13D6 6316 BXDM14 DS    CL1                       BRANCH ADDRESS

                               6318 ***********************************************************************
                               6319 * PRINT CODE TABLE MODE FOR CHARACTER STRING PROCESSING
                               6320 ***********************************************************************
                               6321 *
                          13D7 6322 BXDMD2 EQU   *                         PRS TABLE - MODE 2
    13D7 6B               13D7 6323        DC    AL1(B@CMMA)               DELIMITER - COMMA
    13D8 05               13D8 6324        DC    AL1(B@PRSL)                SPACE TO LONG ZONE
    13D9 48               13D9 6325        DC    AL1(BXD140-BXDPRT)         BRANCH ADDRESS
                               6326 *
    13DA 5E               13DA 6327        DC    AL1(B@SCLN)                DELIMITER - SEMI-COLON
    13DB 06               13DB 6328        DC    AL1(B@PRSS)                SPACE TO SHORT ZONE
    13DC 48               13DC 6329        DC    AL1(BXD140-BXDPRT)         BRANCH ADDRESS
                               6330 *
    13DD 1E               13DD 6331        DC    AL1(B@EOST)                DELIMITER - END OF STATEMENT
    13DE 07               13DE 6332        DC    AL1(B@PRRC)                RETURN THE CARRIAGE
    13DF A7               13DF 6333        DC    AL1(BXD270-BXDPRT)         BRANCH ADDRESS
                               6334 *
    13E0 00               13E0 6335        DC    AL1(BXDDUM)                DELIMITER - NOT , OR ; OR CR
    13E1 01               13E1 6336        DC    AL1(B@PRPN)                PRINT AND NO SPACE
    13E2 5E               13E2 6337        DC    AL1(BXD190-BXDPRT)         BRANCH ADDRESS

                               6339 ***********************************************************************
                               6340 * PRINT CODE TABLE MODE FOR NULL STRING PROCESSING
                               6341 ***********************************************************************
                               6342 *
                          13E3 6343 BXDMD3 EQU   *                          PRS TABLE - MODE 3
    13E3 6B               13E3 6344        DC    AL1(B@CMMA)                DELIMITER - COMMA
    13E4 05               13E4 6345        DC    AL1(B@PRSL)                SPACE TO LONG ZONE
    13E5 48               13E5 6346        DC    AL1(BXD140-BXDPRT)         BRANCH ADDRESS
                               6347 *
    13E6 5E               13E6 6348        DC    AL1(B@SCLN)                DELIMITER - SEMI-COLON
    13E7 01               13E7 6349        DC    AL1(B@PRPN)                PRINT AND NO SPACE
    13E8 0B               13E8 6350        DC    AL1(BXD030-BXDPRT)         BRANCH ADDRESS
                               6351 *
    13E9 1E               13E9 6352        DC    AL1(B@EOST)                DELIMITER - END OF STATEMENT
    13EA 07               13EA 6353        DC    AL1(B@PRRC)                RETURN THE CARRIAGE
    13EB AA               13EB 6354        DC    AL1(BXD280-BXDPRT)         BRANCH ADDRESS
                               6355 *
    13EC 00               13EC 6356        DC    AL1(BXDDUM)                DELIMITER - NOT , OR ; OR CR
    13ED 01               13ED 6357        DC    AL1(B@PRPN)                PRINT AND NO SPACE
    13EE 54               13EE 6358        DC    AL1(BXD170-BXDPRT)         BRANCH ADDRESS

                               6360 ***********************************************************************
                               6361 * PRINT STATEMENT ROUTINE CONSTANTS AND EQUATES
                               6362 ***********************************************************************
                               6363 *
                               6364 * EQUATES
                               6365 *
                          0000 6366 BXDDP0 EQU   0                          PRS TABLE DISP FOR DELIMITER
                          0001 6367 BXDDP1 EQU   1                          PRS TABLE DISP FOR CODE
                          0002 6368 BXDDP2 EQU   2                          PRS TABLE DISP FOR BRANCH ADDR
                          0009 6369 BXDDMY EQU   BXDLTH*3                   PRS TABLE DISP TO DUMMY ENTRY
                               6370 *
                               6371 * CONSTANT
                               6372 *
    13EF 01               13EF 6373 BXDBN1 DC    IL(B@LCNN)'1'              BINARY 1
    13F0 0013             13F1 6374 BXDSUB DC    AL(@VADDR)(B@LCRV)         LENGTH OF SEGMENT TO SUB

                               6376 ***********************************************************************
                               6377 * PRINT STATEMENT ROUTINE STORAGE AND PARAMETER AREA
                               6378 ***********************************************************************
                               6379 *
    13F2 60               13F2 6380 BXDPRC DC    AL(B@LCOP)(B@CPRS)        PRINT AND SPACE OPCODE
    13F3                  13F3 6381 BXDPRO DS    CL(B@LCXX)                PRINT AND SPACE OPERAND
                               6382 *
    13F4 28               13F4 6383 BXDSTC DC    AL(B@LCOP)(B@CSTC)        STACK CHARACTER OPCODE
    13F5                  13F6 6384 BXDSTO DS    CL(@VADDR)                STACK CHARACTER OPERAND

                               6386 ***********************************************************************
                               6387 * PRINT STATEMENT ROUTINE PROGRAM SWITCHES
                               6388 ***********************************************************************
                               6389 *
                          13A8 6390 BXDRS1 EQU   BXD270+@Q                 PRINT LIST SWITCH
                          0007 6391 BXDRM1 EQU   @UCB-@NOP                 PRINT LIST SWITCH MASK
                               6392 *
                               6393 ***********************************************************************
                               6394 *
                               6395 * END OF 'PRINT' STATEMENT ROUTINE CODING
                               6396 *
                               6398 ***********************************************************************
                               6399 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               6400 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               6401 *                                                                     *
                               6402 ***********************************************************************
                               6403 *STATUS                                                               *
                               6404 *   VERSION 1 MODIFICATION 0                                          *
                               6405 *                                                                     *
                               6406 *FUNCTION                                                             *
                               6407 *   BXUPRT IS EXECUTED TO TRANSLATE PRINT USING STATEMENTS AS THEY    *
                               6408 *   OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO   *
                               6409 *   PLACE THE PSEUDOCODE IN VIRTUAL MEMORY.                           *
                               6410 *                                                                     *
                               6411 *ENTRY POINTS                                                         *
                               6412 *   BXUPRT HAS ONLY ONE ENTRY POINT:                                  *
                               6413 *        BXUPRT - TRANSLATE PRINT USING STATEMENT                     *
                               6414 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               6415 *        B     BXUPRT                                                 *
                               6416 *                                                                     *
                               6417 *INPUT                                                                *
                               6418 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               6419 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER OF THE       *
                               6420 *     LEADING KEYWORD, PRINT USING.                                   *
                               6421 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                               6422 *     CHARACTER IN THE LEADING KEYWORD PRINT USING.                   *
                               6423 *                                                                     *
                               6424 *OUTPUT                                                               *
                               6425 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               6426 *     GENERATED BY EXUPRT IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               6427 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               6428 *     SEQUENCES.                                                      *
                               6429 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               6430 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               6431 *                                                                     *
                               6432 *EXTERNAL REFERENCES                                                  *
                               6433 *      B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.  *
                               6434 *      B$PUTC - (B$PCAD, B$PNBY, B$PVAD, B$ARSW) - ENTRY TO COMPILER  *
                               6435 *               VIRTUAL MEMORY OUTPUT ROUTINE.                        *
                               6436 *      B$FCON - (B$CTYP, B$CKCT, B$CPCT) - ENTRY TO BASIC COMPILER    *
                               6437 *               CONSTANT ROUTINE.                                     *
                               6438 *      B$CSCN - (B$CSSW) - ENTRY TO BASIC COMPILER CHARACTER SCAN     *
                               6439 *               ROUTINE.                                              *
                               6440 *      B$SCAN - ENTRY TO BASIC COMPILER ARITHMETIC EXPRESSION SCAN    *
                               6441 *               ROUTINE.                                              *
                               6442 *      B$BTAB - (B$BRVA, B$IRLN) - ENTRY TO BASIC COMPILER BRANCH     *
                               6443 *               TABLE ROUTINE.                                        *
                               6444 *      B$ZDBN - (B$BINO) - ENTRY TO COMPILER ZONED DECIMAL TO BINARY  *
                               6445 *               CONVERSION ROUTINE.                                   *
                               6446 *      B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                  *
                               6447 *                                                                     *
                               6448 *EXITS, NORMAL                                                        *
                               6449 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               6450 *                                                                     *
                               6451 *EXITS, ERROR                                                         *
                               6452 *   N/A                                                               *
                               6453 *                                                                     *
                               6454 *TABLES/WORK AREAS                                                    *
                               6455 *   N/A                                                               *
                               6456 *                                                                     *
                               6457 *ATTRIBUTES                                                           *
                               6458 *   BXUPRT IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               6459 *                                                                     *
                               6460 *CHARACTER CODE DEPENDENCY                                            *
                               6461 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               6462 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               6463 *                                                                     *
                               6464 *NOTES                                                                *
                               6465 *   ERROR PROCEDURES                                                  *
                               6466 *      N/A                                                            *
                               6467 *                                                                     *
                               6468 *   REGISTER USAGE                                                    *
                               6469 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION    *
                               6470 *                                                                     *
                               6471 *   SAVED/RESTORED AREAS                                              *
                               6472 *      N/A                                                            *
                               6473 *                                                                     *
                               6474 *   MODIFICATION CONSIDERATIONS                                       *
                               6475 *      BXUPRT RESIDES ON ONE SECTOR. THE LIMITATION OF THE SECTOR     *
                               6476 *      BOUNDARY ON SIZE SHOULD BE CONSIDERED IN MAKING MODIFICATIONS. *
                               6477 *                                                                     *
                               6478 *   REQUIRED MODULES                                                  *
                               6479 *        @SYSEQ - COMMON SYSTEM EQUATES.                              *
                               6480 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.   *
                               6481 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.      *
                               6482 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               6483 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               6484 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               6485 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               6486 *        $B$EQU - COMPILER FIXED EQUATES.                             *
                               6487 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               6488 *                                                                     *
                               6489 *   OTHER                                                             *
                               6490 *      BXUPRT IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               6491 ***********************************************************************

    1400                       6493        ORG   *,256,0                   BEGIN AT CORE PAGE BOUNDARY
                          1400 6494        USING *,@BR                     DEFINE BASE ADDR FOR CORE PAGE
                               6495 *
                               6496 * ENTER BXUPRT - 'PRINT USING' STATEMENT ROUTINE
                               6497 *
                          1400 6498 BXUPRT EQU   *                         BXUPRT ENTRY POINT
                               6499 *
                               6500 * SKIP TO CHARACTER FOLLOWING KEYWORDS 'PRINT USING'
                               6501 *
    1400 3C 0A 0873            6502 BXU010 MVI   B$NUMC,B@LKPU             SET GET RTN TO SKIP KEYWORDS
    1404 C0 87 0867            6503        B     B$GETC                    LINK TO ADVANCE POINTER
                               6504 *
                               6505 * GENERATE AN 'STA' INSTRUCTION IMAGE IN VIRTUAL MEMORY
                               6506 *
    1408 D2 02 DC              6507 BXU020 LA    BXUSTC(,@BR),@XR          LOAD CADDR OF 'STA' INSTR
    140B 3C 02 0A41            6508        MVI   B$PNBY,B@LSTA-1           SET PUT RTN LNG PARM FOR STA
    140F D0 87 C9              6509        B     BXU360(,@BR)              LINK TO GENERATE 'STA' PMC
```

```
                               6510 *
                               6511 * ESTABLISH 'STA' OPERAND FOR ADDRESS RESOLUTION
                               6512 *
    1412 0C 01 19EF 0A43       6513 BXU025 MVC   B$BRVA,B$PVAD(@VADDR)      SET ADDR FOR BRANCH TABLE
    1418 1F 01 19EF E8         6514        SLC   B$BRVA,BXUBN1(@VADDR,@BR)  ADJUST VADDR TO 'STA' OPERAND
                               6515 *
                               6516 * GENERATE A 'BNX' INSTRUCTION IMAGE IN VIRTUAL MEMORY
                               6517 *
    141D D2 02 DF              6518 BXU030 LA    BXUBNC(,@BR),@XR          LOAD CADDR OF 'BNX' INSTR
    1420 3C 02 0A41            6519        MVI   B$PNBY,B@LBNX-1           SET PUT RTN LNG PARM FOR 'BNX'
    1424 D0 87 C9              6520        B     BXU360(,@BR)              LINK TO GENERATE 'BNX' PMC
                               6521 *
                               6522 * ESTABLISH THE NEXT VADDR IN V.M.(BEGINNING OF DATA OUTPUT SEQUENCE,
                               6523 * AS RESOLUTION ADDRESS
                               6524 *
    1427 0C 01 19F1 0A43       6525 BXU040 MVC   B$BRLN,B$PVAD(@VADDR)     SET ADDR FOR BR TBL RESOLUTION
                               6526 *
                               6527 * CALL BRANCH TABLE ROUTINE TO SET ADDRESS RESOLUTION CONDITIONS FOR
                               6528 * THE 'STA' OPERAND
                               6529 *
    142D C0 87 1996            6530 BXU050 B     B$BTAB                    LINK TO SET RESOLUTION COND
                               6531 *
                               6532 * ESTABLISH VADDR OF 'BNX' OPERAND FOR ADDRESS RESOLUTION
                               6533 *
    1431 0C 01 19EF 0A43       6534 BXU060 MVC   B$BRVA,B$PVAD(@VADDR)     SET ADDRESS FOR BR TABLE
    1437 1F 01 19EF E8         6535        SLC   B$BRVA,BXUBN1(@VADDR,@BR) ADJUST VADDP TO 'BNX' OPERAND
                               6536 *
                               6537 * CONVERT THE IMAGE LINE NUMBER TO BINARY FROM DECIMAL
                               6538 *
    143C C0 87 19F2            6539 BXU070 B     B$ZDBN                    LINK TO CONVERT LINE NO TO BIN
                               6540 *
                               6541 * ESTABLISH THE IMAGE LINE NUMBER AS RESOLUTION LINE NUMBER
                               6542 *
    1440 0C 01 19F1 1A6A       6543 BXU080 MVC   B$BRLN,B$BINO(@VADDR)     SET LN NO FOR BR TBL RESOLUTION
                               6544 *
                               6545 * CALL BRANCH TABLE ROUTINE TO SET ADDRESS RESOLUTION CONDITIONS FOR
                               6546 * THE 'BNX' OPERAND
                               6547 *
    1446 C0 87 1996            6548 BXU090 B     B$BTAB                    LINK TO SET RESOLUTION COND
                               6549 *
                               6550 * CHECK FOR THE PRESENCE OF LIST ELEMENTS
                               6551 *
    144A BD 1E 00              6552 BXU100 CLI   B@CHAR(,@XR),B@EOST       IF LIST ELEMENTS ARE PRESENT
    144D F2 01 10              6553        JNE   BXU170                    GO ATTEMPT PMC GENERATION
                               6554 *
                               6555 * SET CODE FOR NO LIST ELEMENTS IN THE 'PRU' INSTRUCTION
                               6556 *
    1450 7C 02 E3              6557 BXU110 MVI   BXUPRO(,@BR),B@PUNL       SET 'PRU' OPERAND FOR NO LIST
                               6558 *
                               6559 * SET TERMINATOR FLAG TO INDICATE LAST OUTPUT FOR LIST
                               6560 *
    1453 7A 10 E3              6561 BXU120 SBN   BXUPRO(,@BR),B@PUTM       SET LAST OUTPUT FOR LIST FLAG
                               6562 *
                               6563 * GENERATE THE 'PRU' INSTRUCTION IN VIRTUAL MEMORY
                               6564 *
    1456 D0 87 C2              6565 BXU130 B     BXU350(,@BR)              BRANCH TO GENERATE 'PRU' PMC
```

```
                               6566 *
                               6567 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               6568 *
    1459 C0 87 0700            6569 BXU140 B     B$DIST                   RETURN TO DISTRIBUTOR
                               6570 *
                               6571 * GENERATE THE 'PRU' INSTRUCTION IN VIRTUAL MEMORY
                               6572 *
    145D D0 87 C2              6573 BXU150 B     BXU350(,@BR)              BRANCH TO GENERATE 'PRU' PMC
                               6574 *
                               6575 * CALL ARITH SCAN ROUTINE TO ATTEMPT PMC GENERATION OF ARUN EXPRESSION
                               6576 *
    1460 C0 87 1514            6577 BXU170 B     B$SCAN                   LINK TO ATTEMPT PMC GENERATION
                               6578 *
                               6579 * TEST FOR THIS LIST ELEMENT BEING A CHARACTER VARIABLE
                               6580 *
    1464 38 07 14BC            6581 BXU180 TBN   B$CSSW,B$CSMK            IF ELEMENT IS NOT A CHAR VAR
    1468 F2 90 04              6582        JF    BXU200                   * GO SET 'PRU' OPERAND
                               6583 *
                               6584 * IF THIS LIST ELEMENT IS A CHARACTER VARIABLE CALL THE CHAR SCAN RTN
                               6585 *
    146B C0 87 14B0            6586 BXU190 B     B$CSCN                   LINK, GENERATE PMC FOR CHAR VAR
                               6587 *
                               6588 * SET 'PRU' OPERAND WITH CODE FOR ARITHMETIC OR CHARACTER EXPRESSION,
                               6589 * INCLUDING FIRST CONSTANT ESTABLISHED FOR A CHAR STRING BUT EXCLUDING
                               6590 * A NULL CHAR STRING
                               6591 *
    146F 7C 06 E3              6592 BXU200 MVI   BXUPRO(,@BR),B@PUD1       SET 'PRU' OPERAND CODE
                               6593 *
                               6594 * TEST FOR ANY PMC HAVING BEEN GENERATED FOR THIS ELEMENT
                               6595 *
    1472 38 01 0A45            6596 BXU210 TBN   B$ARSW,B$ARMK            IF NO PMC GENERATED
    1476 F2 90 0D              6597        JF    BXU230                   * GO BRANCH TO CONSTANT RTN
                               6598 *
                               6599 * TEST FOR DELIMITER BEING AN END OF STATEMENT
                               6600 *
    1479 35 02 0878            6601 BXU220 L     B$GPTR,@XR               RESTORE TEXT POINTER
    147D BD 1E 00              6602        CLI   B@CHAR(,@XR),B@EOST       IF DELIMITER IS NOT TERMINATOR
    1480 D0 01 5D              6603        BNE   BXU150(,@BR)             * GO GENERATE 'PRU' PMC
    1483 D0 87 53              6604        B     BXU120(,@BR)             GO SET LAST LIST OUTPUT FLAG
                               6605 *
                               6606 * CALL CONSTANT ROUTINE TO GENERATE CHARACTER STRING IN V.M.
                               6607 *
    1486 3C 1B 0A5F            6608 BXU230 MVI   B$CTYP,B$SCON            SET CON RTN FOR CHAR STRING
    148A C0 87 0A46            6609        B     B$FCON                   LINK TO GENERATE CHAR STRING
                               6610 *
                               6611 * TEST FOR THIS BEING A NULL STRING
                               6612 *
    148E 3D 00 0CA8            6613 BXU240 CLI   B$CPCT,@ZERO             IF THIS IS A NOT A NULL STRING
    1492 F2 01 06              6614        JNE   BXU260                   * MOVE 1ST SEGMENT VADDR TO STC
                               6615 *
                               6616 * IF THIS IS A NULL CHARACTER STRING SET CODE IN 'PRU' OPERAND
                               6617 *
    1495 7C 03 E3              6618 BXU250 MVI   BXUPRO(,@BR),B@PUNS       SET 'PRU' OPND FOR NULL STRING
    1498 D0 87 79              6619        B     BXU220(,@BR)             GO CHECK FOR OTHER ELEMENTS
                               6620 *
                               6621 * MOVE THE VADDR OF THE FIRST STRING SEGMENT TO AN 'STC' INSTR OPWD
                               6622 *
    149B 4C 01 E6 1590         6623 BXU260 MVC BXUSCO(,@BR),B$BCKT(@VADDR)  SET 1ST SEGMENT VADDR IN OPND
                               6624 *
                               6625 * SET THE 'PRU' OPND CODE FOR ARITH AND CHAR EXPRESSIONS
                               6626 *
    14A0 7C 06 E3              6627 BXU270 MVI   BXUPRO(,@BR),B@PUD1       SET 'PRU' OPERAND CODE
                               6628 *
                               6629 * GENERATE THE 'STU PMC INSTRUCTION IN VIRTUAL MEMORY
                               6630 *
    14A3 D2 02 E4              6631 BXU280 LA    BXUSCC(,@BR),@XR          LOAD CADDR OF 'STC' INSTR
    14A6 3C 02 0A41            6632        MVI   B$PNBY,B@LSTC-1           SET PUT RTN LNG PARM FOR 'STC'
    14AA D0 87 C9              6633        B     BXU360(,@BR)              LINK TO GENERATE 'STC' PMC
                               6634 *
                               6635 * TEST FOR THE EXISTENCE OF ANOTHER SEGMENT
                               6636 *
    14AD 1F 00 0CA8 E8         6637 BXU290 SLC   B$CPCT,BXUBN1(1,@BR)      IF NO OTHER SEGMENTS EXIST
    14B2 D0 04 79              6638        BNH   BXU220(,@BR)              * GO TEST FOR OTHER ELEMENTS
                               6639 *
                               6640 * GENERATE THE 'PRU' INSTRUCTION IN VIRTUAL MEMORY
                               6641 *
    14B5 D0 87 C2              6642 BXU300 B     BXU350(,@BR)              BRANCH TO GENERATE 'PRU' PMC
                               6643 *
                               6644 * SET 'PRU' OPERAND CODE FOR ANY CONSTANT ESTABLISHED FOR A CHAR STRING
                               6645 * EXCEPT FOR THE FIRST CONSTANT IN THAT STRING SERIES
                               6646 *
    14B8 7C 07 E3              6647 BXU310 MVI   BXUPRO(,@BR),B@PUD2       SET 'PRU' OPND CODE
                               6648 *
                               6649 * SUBTRACT LENGTH OF STRING SEGMENT FROM 'STC' INSTRUCTION OPERAND
                               6650 *
    14BB 5F 01 E6 EA           6651 BXU320 SLC   BXUSCO(,@BR),BXUSUB(@VADDR,@BR)  SUB SEGMENT LENGTH
                               6652 *
                               6653 * BRANCH TO CONTINUE GENERATING THE 'STC'/'PRU' SEQ FOR THE CHAR STRING
                               6654 *
    14BF D0 87 A3              6655 BXU340 B     BXU280(,@BR)              BRANCH TO GENERATE 'STC' INSTR
                               6656 *
                               6657 ***********************************************************************
                               6658 * SUBROUTINE FOR THE GENERATION OF PSEUDOCODE IN VIRTUAL MEMORY -     *
                               6659 * * THE ENTIRE ROUTINE IS USED TO GENERATE THE 'PRU' INSTRUCTION      *
                               6660 * * AND SECONDARY ENTRY POINT ALLOWS THE ROUTINE TO COMPLETE THE      *
                               6661 * * GENERATION FOR THE 'STA', 'BNX' AND 'STC' INSTRUCTIONS.           *
                               6662 ***********************************************************************

                               6664 *
                               6665 * ENTER THE GENERATE SUBROUTINE - FOR 'PRU' INSTRUCTION
                               6666 *
    14C2 D2 02 E2              6667 BXU350 LA    BXUPRC(,@BR),@XR          LOAD CADDR OF 'PRU' INSTR
    14C5 3C 01 0A41            6668        MVI   B$PNBY,B@LPRU-1           SET PUT RTN FOR LENGTH PARM
                               6669 *
                               6670 * SECONDARY ENTRY POINT TO GENERATE SUBROUTINE FOR 'STA', 'BNX', 'STC'
                               6671 *
    14C9 74 08 DB              6672 BXU360 ST    BXU370+@OP1(,@BR),@ARR    STORE RETURN ADDRESS
    14CC 34 02 0A40            6673        ST    B$PCAD,@XR                SET PUT RTN VADDR PARM
    14D0 C0 87 093A            6674        B     B$PUTC                    LINK TO GENERATE PMC
    14D4 35 02 0878            6675        L     B$GPTR,@XR                RESTORE TEXT POINTER
    14D8 C0 87 0000            6676 BXU370 B     *-*                       BRANCH TO RETURN ADDRESS
```

```
                               6678 ***********************************************************************
                               6679 * PRINT USING STATEMENT RTN PARAMETER AND STORAGE AREAS
                               6680 ***********************************************************************
                               6681 *
    14DC 34                14DC 6682 BXUSTC DC    AL(B@LCOP)(B@CSTA)        'STA' INSTR OPCODE
    14DD 0000              14DE 6683 BXUSTO DC    XL(B@LCVA)'00'            'STA' INSTR OPERAND IMAGE
                               6684 *
    14DF 4A                14DF 6685 BXUBNC DC    AL(B@LCOP)(B@CBNX)        'INX' INSTR OPCODE
    14E0 0000              14E1 6686 BXUBNO DC    XL(B@LCVA)'00'            'INX' INSTR OPERAND IMAGE
                               6687 *
    14E2 62                14E2 6688 BXUPRC DC    AL(B@LCOP)(B@CPRU)        'PRU' INSTR OPCODE
    14E3                   14E3 6689 BXUPRO DS    CL(B@LCXX)                'PRU' INSTR OPERAND
                               6690 *
    14E4 28                14E4 6691 BXUSCC DC    AL(B@LCOP)(B@CSTC)        'STC' INSTR OPCODE
    14E5                   14E6 6692 BXUSCO DS    CL(B@LCVA)                'STC' INSTR OPERAND

                               6694 ***********************************************************************
                               6695 * PRINT USING STATEMENT ROUTINE CONSTANTS
                               6696 ***********************************************************************
                               6697 *
    14E7 0001              14E8 6698 BXUBN1 DC    IL(@VADDR)'1'             BINARY 1
                               6699 *
    14E9 0013              14EA 6700 BXUSUB DC    AL(@VADDR)(B@LCRV)        LENGTH OF STRING SEGMENT
                               6701 *
                               6702 ***********************************************************************
                               6703 *
                               6704 * END OF 'PRINT USING' STATEMENT ROUTINE CODING
                               6705 *
```

```
6707 ***********************************************************************
6708 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
6709 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
6710 *                                                                     *
6711 ***********************************************************************
6712 *STATUS                                                               *
6713 *   VERSION 1 MODIFICATION 0                                          *
6714 *                                                                     *
6715 *FUNCTION                                                             *
6716 *   BNFDEF IS EXECUTED TO TRANSLATE DEF STATEMENTS AS THEY OCCUR IN A *
6717 *   BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE    *
6718 *   PSEUDOCODE IN VIRTUAL MEMORY.                                     *
6719 *                                                                     *
6720 *ENTRY POINTS                                                         *
6721 *   BNFDEF HAS ONLY ONE ENTRY POINT:                                  *
6722 *        BNFDEF - TRANSLATE DEF STATEMENT                             *
6723 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
6724 *        B     BNFDEF                                                 *
6725 *                                                                     *
6726 *INPUT                                                                *
6727 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
6728 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
6729 *     LEADING KEYWORD, DEF.                                           *
6730 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
6731 *     CHARACTER IN THE LEADING KEYWORD, DEF.                          *
6732 *   * FUNCTION ATTRIBUTE FIELDS - THE CORE-RESIDENT VIRTUAL ADDRESS   *
6733 *     STORAGE LOCATIONS FOR EACH OF THE 29 POSSIBLE USER FUNCTIONS.   *
6734 *     ATTRIBUTE FIELDS FOR PREVIOUSLY DEFINED USER FUNCTIONS CONTAIN  *
6735 *     THE ENTRY POINT VIRTUAL ADDRESS ASSOCIATED WITH EACH FUNCTION.  *
6736 *     UNDEFINED ATTRIBUTE FIELDS ARE CLEARED TO ZERO.                 *
6737 *                                                                     *
6738 *OUTPUT                                                               *
6739 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
6740 *     GENERATED BY BNFDEF IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
6741 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
6742 *     SEQUENCES.                                                      *
6743 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
6744 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
6745 *   * FUNCTION ATTRIBUTE FIELDS - UPDATED WITH ENTRY POINT VIRTUAL    *
6746 *     ADDRESS ASSOCIATED WITH THE USER FUNCTION DEFINED BY THE        *
6747 *     CURRENT STATEMENT.                                              *
6748 *     B$BRVA - CONTAINS THE VIRTUAL ADDRESS OF THE RIGHT BYTE OF THE  *
6749 *     ADDRESS OPERAND FIELD IN THE BYPASS BRANCH INSTRUCTION          *
6750 *     GENERATED FOR THE CURRENT STATEMENT.                            *
6751 *   * B$NXSW - SET TO ON STATUS TO CAUSE RESOLUTION OF THE BYPASS     *
6752 *     BRANCH INSTRUCTION OPERAND ADDRESS.                             *
6753 *                                                                     *
6754 *EXTERNAL REFERENCES                                                  *
6755 *      B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIE,AL ROUTINE.  *
6756 *      B$PUTC - (B$PFNC, B$PCAD, B$PNBY, B$PCOL, B@PERC, B$PVAD) -    *
6757 *               ENTRY TO COMPILER VIRTUAL MEMORY OUTPUT ROUTINE.      *
6758 *      B$SYMB - (B$BCKT, B$FSVA, B$FSSW, B$FACA, B$FSC1, B$FSC2) -    *
6759 *               ENTRY TO BASIC SYMBOL TRANSLATION ROUTINE.            *
6760 *      B$SCAN - ENTRY TO BASIC ARITHMETIC EXPRESSION SCAN ROUTINE.    *
6761 *      B$BTAB - (B$BRVA) - ENTRY TO BASIC COMPILER BRANCH TABLE       *
6762 *               ROUTINE.                                              *
```

```
                               6763 *      $XIND1 - INDICATOR FOR LONG CO SHOW PRECISION.                 *
                               6764 *      B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                  *
                               6765 *                                                                     *
                               6766 *EXITS, NORMAL                                                        *
                               6767 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               6768 *                                                                     *
                               6769 *EXITS, ERROR                                                         *
                               6770 *   N/A                                                               *
                               6771 *                                                                     *
                               6772 *TABLES/WORK AREAS                                                    *
                               6773 *   * FUNCTION ATTRIBUTE FIELDS - EXTERNAL TO 1NFDEF, THESE FIELDS    *
                               6774 *     CONTAIN VIRTUAL ADDRESSES FOR THE 29 POSSIBLE USER FUNCTION     *
                               6775 *     ENTRY POINTS AS THEY ARE DEFINED IN A PROGRAM.                  *
                               6776 *                                                                     *
                               6777 *ATTRIBUTES                                                           *
                               6778 *   BNFDEF IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               6779 *                                                                     *
                               6780 *CHARACTER CODE DEPENDENCY                                            *
                               6781 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               6782 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               6783 *                                                                     *
                               6784 *NOTES                                                                *
                               6785 *   ERROR PROCEDURES                                                  *
                               6786 *      WHEN A DEF STATEMENT ATTEMPTS TO DEFINE A USER FUNCTION WHICH  *
                               6787 *      HAS BEEN PREVIOUSLY DEFINED IN THE SAME PROGRAM, THE ERROR     *
                               6788 *      CONDITION CODE FOR 'DUPLICATE DEFINITION OF USER FUNCTION' IS  *
                               6789 *      LOGGED IN VIRTUAL MEMORY USING OUTPUT ROUTINE B@PUTC.          *
                               6790 *      COMPILATION IS OTHERWISE UNAFFECTED.                           *
                               6791 *                                                                     *
                               6792 *   REGISTER USAGE                                                    *
                               6793 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               6794 *                                                                     *
                               6795 *   SAVED/RESTORED AREAS                                              *
                               6796 *      N/A                                                            *
                               6797 *                                                                     *
                               6798 *   MODIFICATION CONSIDERATIONS                                       *
                               6799 *      BNFDEF RESIDES ON ONE SECTOR. ANY MODIFICATION SHOULD CONSIDER *
                               6800 *      THE SIZE LIMITATION.                                           *
                               6801 *                                                                     *
                               6802 *      REQUIRED MODULES                                               *
                               6803 *        @SYSEQ - COMMON SYSTEM EQUATES.                              *
                               6804 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.   *
                               6805 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               6806 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               6807 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               6808 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               6809 *        $B$EQU - COMPILER FIXED EQUATES.                             *
                               6810 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               6811 *                                                                     *
                               6812 *   OTHER                                                             *
                               6813 *      BNFDEF IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS       *
                               6814 ***********************************************************************

    1500                       6816        ORG   *,256,0                   PLACE MODULE AT PAGE BOUNDARY
                          1500 6817        USING *,@BR                     ESTABLISH BASE ADDRESSING
                               6818 *
                               6819 * ENTER BNFDEF - 'DEF' STATEMENT ROUTINE
                               6820 *
                          1500 6821 BNFDEF EQU   *
                               6822 *
                               6823 * SET INPUT PARAMETER TO SKIP KEYWORD 'DEF'
                               6824 *
    1500 3C 03 0873            6825 BNF010 MVI   B$NUMC,B@LDEF             SET GET RTN TO SKIP 'DEF'
    1504 C0 87 0867            6826        B     B$GETC                    LINK TO ADVANCE POINTER
                               6827 *
                               6828 * GENERATE A BYPASS BRANCH INSTRUCTION IMAGE
                               6829 *
    1508 D2 02 BC              6830 BNF020 LA    BNFBRC(,@BR),@XR          LOAD CADDR OF 'BRA' INSTR
    150B 34 02 0A40            6831        ST    B$PCAD,@XR                SET PUT RTN VADDR FOR 'BRA'
    150F C0 87 093A            6832        B     B$PUTC                    LINK TO GENERATE 'BRA' PMC
                               6833 *
                               6834 * SAVE NEXT AVAILABLE PMC VADDR FOR BRANCH RESOLUTIONS AND
                               6835 * FUNCTION TABLE ENTRY
                               6836 *
    1513 0C 01 19EF 0A43       6837 BNF030 MVC   B$BRVA,B$PVAD(@VADDR)     SAVE 'BRA' VADDR FOR RESOLUTION
                               6838 *
                               6839 * CALL SYMBOL ROUTINE TO DETERMINE THE VIRTUAL ADDRESS OF THE FUNCTION
                               6840 * TABLE LOCATION ASSOCIATED WITH THE CURRENT USER FUNCTION
                               6841 *
    1519 35 02 0878            6842 BNF040 L     B$GPTR,@XR                RESTORE TEXT POINTER
    151D C0 87 0DBC            6843        B     B$SYMB                    LINK TO GET CADDR OF USER FUNC
                               6844 *
                               6845 * CHECK CADDR OF USER FUNC FOR INDICATION OF PREVIOUS DEFINITION
                               6846 *
    1521 35 02 0E53            6847 BNF050 L     B$FACA,@XR                LOAD CADDR OF USER FUNCTION
    1525 BD 56 00              6848        CLI   B@FVAD-1(,@XR),B@DVC1     IF FUNCTION NOT DEFINED
    1528 F2 82 0C              6849        JL    BNF070                    * JUMP TO PROCESS USER FUNCTION
                               6850 *
                               6851 * GENERATE ERROR MESSAGE IF FUNCTION HAS BEEN PREVIOUSLY DEFINED
                               6852 *
    152B 3C 33 094E            6853 BNF060 MVI   B$PFNC,B$PFAE             SET PUT RTN FOR ERROR OUTPUT
    152F 3C AA 0A39            6854        MVI   B$PERC,@@E604             SET PUT RTN FOR 'INVALID FUNC'
    1533 C0 87 093A            6855        B     B$PUTC                    LINK TO GENERATE ERROR PMC
                               6856 *
                               6857 * TEST FOR PRECISION BEFORE GENERATING FUNCTION LINKAGE SEQUENCE
                               6858 *
    1537 38 40 03D0            6859 BNF070 TBN   $XIND1,$XPREC             IF PRECISION IS STANDARD
    153B F2 90 06              6860        JF    BNF080                    * SKIP TO GENERATE LINKAGE SE0
    153E 7C 0D CA              6861        MVI   BNFSPA(,@BR),BNFLIP       SET LENGTH FOR LONG PREC
    1541 7C 09 C0              6862        MVI   BNFDAN(,@BR),B@LILP       SET 'DWA' OPERAND FON LONG PREC
                               6863 *
                               6864 * GENERATE RETURN LINKAGE 'BRA' INSTR AND PARAMETER AREA
                               6865 *
    1544 1C 00 0A41 CA         6866 BNF080 MVC   B$PNBY,BNFSPA(1,@BR)      SET PUT RTN LNG FOR 'BRA' RET
    1549 C0 87 093A            6867        B     B$PUTC                    LINK TO GENERATE RET LINK SEQ
    154D 4C 01 CD 0A43         6868        MVC   BNFBDO(,@BR),B$PVAD(@VADDR)   MOVE VIRTUAL ADDR OF LINKAGE
    1552 4F 00 CD 09D3         6869        SLC   BNFBDO(,@BR),B$PCDL(@VADDR-1) * BRA INST TO 'BRD' OPERAND
                               6870 *
                               6871 * ESTABLISH THE VADDR OF THE 'BRA' RETURN LINKAGE PMC AS THE FUNCTION
                               6872 * TABLE ENTRY FOR THE USER FUNCTION CURRENTLY REFERENCED
                               6873 *
    1557 35 02 0E53            6874 BNF090 L     B$FACA,@XR                MOVE CADDR OF FUNC TBL ENTRY
    155B 9C 01 01 CD           6875        MVC   B@FVAD(,@XR),BNFBDO(@VADDR,@BR)  MOVE VADDR OF 'BRA' INSTR
                               6876 *
                               6877 * ADVANCE TEXT POINTER TO REFERENCE 1ST CHAR OF THE FUNC DUMMY ARG
                               6878 *
    155F C0 87 0867            6879 BNF100 B     B$GETC                    LINK TO GET NEXT CHARACTER
                               6880 *
                               6881 * MOVE THE FIRST CHARACTER OF THE DUMMY ARG NAME INTO THE DUMMY SYMBOL
                               6882 * WORD OF THE SYMBOL ROUTINE
                               6883 *
    1563 2C 00 0E4C 00         6884 BNF110 MVC   B$FSC1,B@CHAR(1,@XR)      MOVE 1ST USER FUNC ARC CHAR
    1568 C0 87 0867            6885        B     B$GETC                    LINK TO GET NEXT CHAR
                               6886 *
                               6887 * TEST FOR A SECOND USER FUNCTION CHARACTER
                               6888 *
    156C BD 5D 00              6889 BNF120 CLI   B@CHAR(,@XR),B@RPAR       IF NO 2ND USER FUNC ARG CHAR
    156F F2 81 10              6890        JE    BNF140                    * JUMP TO 'BLANK' SYMBOL WORD
                               6891 *
                               6892 * MOVE 2ND CHAR OF DUMMY IN NAME INTO DUMMY SYMBOL WORD OF SYMBOL RTN
                               6893 *
    1572 2C 00 0E4D 00         6894 BNF130 MVC   B$FSC2,B@CHAR(1,@XR)      MOVE 2ND USER FUNC ARG CHAR
    1577 3C 02 0873            6895        MVI   B$NUMC,BNFSKP             SET GET RTN TO SKIP 10
    157B C0 87 0867            6896        B     B$GETC                    LINK TO ADVANCE TEXT POINTER
    157F F2 87 08              6897        J     BNF150                    JUMP TO SET OTHER TM RTN PARM
                               6898 *
                               6899 * MOVE A BLANK AS 2ND CHAR OF USER FUNC DUMMY ARC NAME INTO THE DUMMY
                               6900 * SYMBOL WORD OF THE SYMBOL ROUTINE
                               6901 *
    1582 3C 40 0E4D            6902 BNF140 MVI   B$FSC2,B@BLNK             MOVE A BLNK INTO DUMMY SYM ND
    1586 C0 87 0867            6903        B     B$GETC                    LINK TO GET NEXT CHARACTER
                               6904 *
                               6905 * MOVE THE VADDR OF THE 'BRA' RETURN LINKAGE PARAMATER AREA
                               6906 * INTO THE SYMBOL ROUTINE INPUT PARAMETER
                               6907 *
    158A 1C 01 0E4F CD         6908 BNF150 MVC   B$FSVA,BNFBDO(@VADDR,@BR)   MOVE THE VADDR OF LAST 'BRA'
    158F 1E 00 0E4F CE         6909        ALC   B$FSVA,BNFLTH(@VADDR-1,@BR) ADJUST TO VADDR OF WORK AREA
                               6910 *
                               6911 * SET THE FUNCTION SCAN SWITCH ON TO INDICATE THE VARIABLE IS A USER
                               6912 * FUNCTION DUMMY ARGUMENT NAME
                               6913 *
    1594 3A 07 0E5C            6914 BNF160 SBN   B$FSSW,B$FSMK             SET FUNCTION SCAN SWITCH
                               6915 *
                               6916 * CALL THE ARITH SCAN RTN TO GENERATE THE PMC'S FOR THE ARITH EXPR
                               6917 *
    1598 C0 87 1514            6918 BNF170 B     B$SCAN                    LINK TO PROCESS ARUM EXPR
    159C 3B 07 0E5C            6919        SBF   B$FSSW,B$FSMK             SET FUNC SCAN SI41V-4 OFF
                               6920 *
                               6921 * GENERATE A 'BRD' INSTRUCTION TO COMPLETE 1HE TRANSFER OF CONTROL TO
                               6922 * THE CALLING EXPRESSION
                               6923 *
    15A0 D2 02 CB              6924 BNF180 LA    BNFBDC(,@BR),@XR          LOAD CADDR OF 'BRD' INSIR
    15A3 34 02 0A40            6925        ST    B$PCAD,@XR                SET PUT RTN VADDR FOR 'BRD'
    15A7 3C 02 0A41            6926        MVI   B$PNBY,B@LBRD-1           SET LENGTH OF 'BRD'.
    15AB C0 87 093A            6927        B     B$PUTC                    LINK TO GENERATE 'BRD' PMC
                               6928 *
                               6929 * STORE THE VADDR OF THE FIRST 'BRA' INSTR OPERAND FOE ADDRESS
                               6930 * RESOLUTION IN THE BRANCH ADDRESS TABLE
                               6931 *
    15AF 1F 00 19EF CF          6932 BNF190 SLC   B$BRVA,BNFBN1(@VADDR-1,@BR)  ADJUST VADDR TO 'BRA' OPRND
                               6933 *
                               6934 * SET 'NEXT' SWITCH TO CAUSE BRANCH TABLE LINE NUMBER RESOLUTION
                               6935 *
    15B4 3A 07 071D            6936 BNF200 SBN   B$NXSW,B$NXMK             SET 'NEXT' SWITCH ON
                               6937 *
                               6938 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               6939 *
    15B8 C0 87 0700            6940 BNF210 B     B$DIST                    RETURN TO DISTRIBUTOR
```

```
                               6942 ***********************************************************************
                               6943 * 'DEF' STATEMENT ROUTINE PMC AND STORAGE PARAMETERS
                               6944 ***********************************************************************
                               6945 *
    15BC 46                 15BC 6946 BNFBRC DC    AL(B@LCOP)(B@CBRA)        'BRA' IMAGE OPCODE
    15BD 0000               15BE 6947 BNFBRO DC    XL(B@LCVA)'00'            'BRA' IMAGE OPERAND
                               6948 *
    15BF 6E                 15BF 6949 BNFDAC DC    AL(B@LCOP)(B@CDWA)        'DNA' INSTRUCTION OPCODE
    15C0                    15C0 6950 BNFDAN DS    CL(B@LCNN)                'DNA' INSTRUCTION OPERAND
    15C0                         6951        ORG   BNFDAN                    INITIALIZE 'DMA' OPERAND FOR
    15C0 05                 15C0 6952        DC    AL(B@LCNN)(B@LISP)        * STANDARD PREC PACKED FLT PT
                               6953 *
    15C1 0000000000000000 15C9 6954 BNFWKA DC    XL(B@LILP)'00'            USER FUNCTION ARGUMENT AREA
                               6955 *
    15CA                    15CA 6956 BNFSPA DS    CL1                       'BRA' & ARG FIELD LENGTH - 1
    15CA                         6957        ORG   BNFSPA                    LENGTH SET FOR SHORT PRECISION
    15CA 09                 15CA 6958        DC    AL1(B@LBRA+B@LDWA+B@LISP-1)  * CHANGE FOR LONG PRECISION
                               6959 *
    15CB 48                 15CB 6960 BNFBDC DC    AL(B@LCOP)(B@CBRD)        'BRD' INSTR OPCODE
    15CC                    15CD 6961 BNFBDO DS    CL(B@LCVA)                'BRD' INSTR OPERAND

                               6963 ***********************************************************************
                               6964 * 'DEF' STATEMENT ROUTINE CONSTANTS AND EQUATES
                               6965 ***********************************************************************
                               6966 *
                               6967 * CONSTANTS
                               6968 *
    15CE 05                 15CE 6969 BNFLTH DC    AL1(B@LBRA+B@LDWA)        LENGTH OF 'BRA' A 'DWA' PMC'S
    15CF 01                 15CF 6970 BNFBN1 DC    IL(@VADDR-1)'1'           BINARY INTEGER +1
                               6971 *
                               6972 * EQUATES
                               6973 *
                          0002 6974 BNFSKP EQU   2                         LENGTH OF TWO CHARACTERS
                               6975 *
                          000D 6976 BNFLIP EQU   B@LBRA+B@LDWA+B@LILP-1    LENGTH FOR LONG INTERNAL PREC
                               6977 *
                               6978 ***********************************************************************
                               6979 *
                               6980 * END OF 'DEF' STATEMENT ROUTINE CODING
                               6981 *
                               6983 ************************************************************************
                               6984 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                   *
                               6985 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083       *
                               6986 *                                                                      *
                               6987 ************************************************************************
                               6988 *STATUS                                                                *
                               6989 *   VERSION 1 MODIFICATION 0                                           *
                               6990 *                                                                      *
                               6991 *FUNCTION                                                              *
                               6992 *   BPMLET IS EXECUTED TO TRANSLATE MULTIPLE ARITHMETIC ASSIGNMENT     *
                               6993 *   AND LET STATEMENTS AS THEY OCCUR IN A BASIC PROGRAM INTO THE       *
                               6994 *   APPROPRIATE PSEUDOCODE AND TO PLACE THE PSEUDOCODE INTO VIRTUAL    *
                               6995 *   MEMORY.                                                            *
                               6996 *                                                                      *
                               6997 *ENTRY POINTS                                                          *
                               6998 *   BPMLET HAS TWO ENTRY POINTS:                                       *
                               6999 *        BPMASN - TRANSLATE MULTIPLE ARITHMETIC ASSIGNMENT STATEMENT   *
                               7000 *        BPMLET - TRANSLATE MULTIPLE ARITHMETIC LET STATEMENT          *
                               7001 *   THE FORMAT OF THE CALLING SEQUENCES IS AS FOLLOWS:                 *
                               7002 *        B     BPMASN                                                  *
                               7003 *        B     BPMLET                                                  *
                               7004 *                                                                      *
                               7005 *INPUT                                                                 *
                               7006 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING   *
                               7007 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE        *
                               7008 *     LEADING KEYWORD, LET, OR IN THE ASSIGNMENT LIST IF THE           *
                               7009 *     OPTIONAL KEYWORD IS OMITTED.                                     *
                               7010 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE 1ST    *
                               7011 *     CHARACTER IN THE LEADING KEYWORD, LET, OR IN THE ASSIGNMENT      *
                               7012 *     LIST IF THE OPTIONAL KEYWORD IS OMITTED.                         *
                               7013 *                                                                      *
                               7014 *OUTPUT                                                                *
                               7015 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE         *
                               7016 *     GENERATED BY BPMLET IS STORED IN THE NEXT AVAILABLE VIRTUAL      *
                               7017 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION          *
                               7018 *     SEQUENCES. GENERATED PROGRAM CONSTANTS WILL BE STORED UNDER      *
                               7019 *     CONTROL OF THE COMPILER CONSTANT ROUTINE BCFCON.                 *
                               7020 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE        *
                               7021 *     CHARACTER WHICH TERMINATES THE STATEMENT.                        *
                               7022 *                                                                      *
                               7023 *EXTERNAL REFERENCES                                                   *
                               7024 *        B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC TEXT RETRIEVAL     *
                               7025 *                 ROUTINE.                                             *
                               7026 *        B$PUTC - (B$PCAD, B$PNBY, B$PVAD) - ENTRY TO COMPILER         *
                               7027 *                 VIRTUAL MEMORY OUTPUT ROUTINE.                       *
                               7028 *        B$SCAN - ENTRY TO BASIC ARITHMETIC EXPRESSION SCAN ROUTINE.   *
                               7029 *        B$LIST - ENTRY TO BASIC COMPILER LIST ADDRESS ROUTINE.        *
                               7030 *        B$BTAB - (B$BRVA, B$BRLN) - ENTRY TO BASIC COMPILER BRANCH    *
                               7031 *                 TABLE ROUTINE.                                       *
                               7032 *        B$DIST - (B$NXSW) - ENTRY TO BASIC COMPILER DISTRIBUTOR       *
                               7033 *        B$WORK - ENTRY TO WORK AREA IN COMMON AREA OF CORE.           *
                               7034 *                                                                      *
                               7035 *EXITS, NORMAL                                                         *
                               7036 *   B$DIST - (B$NXSW) - ENTRY TO BASIC COMPILER DISTRIBUTOR            *
                               7037 *                                                                      *
                               7038 *EXITS, ERROR                                                          *
                               7039 *   N/A                                                               *
                               7040 *                                                                     *
                               7041 *TABLES/WORK AREAS                                                    *
                               7042 *   * WORK AREA &WRK, WHOSE ADDRESS IS REFERENCED BY B$WORK, IS       *
                               7043 *     USED FOR THE RUN-TIME STACKING AND UNSTACXING OF THE VALUE OF   *
                               7044 *     THE ARITHMETIC EXPRESSION ON THE RIGHT SIDE OF THE EQUAL SIGN.  *
                               7045 *                                                                     *
                               7046 *ATTRIBUTES                                                           *
                               7047 *   BPMLET IS NATURALLY RELOCATABLE AND REUSABLE                      *
                               7048 *                                                                     *
                               7049 *CHARACTER CODE DEPENDENCY                                            *
                               7050 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               7051 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               7052 *                                                                     *
                               7053 *NOTES                                                                *
                               7054 *   ERROR PROCEDURES                                                  *
                               7055 *      N/A                                                            *
                               7056 *                                                                     *
                               7057 *   REGISTER USAGE                                                    *
                               7058 *      BOTH THE INDEX AND BASE REGISTERS ARE USED IN THE EXECUTION    *
                               7059 *      OF BPMLET.                                                     *
                               7060 *                                                                     *
                               7061 *   SAVED/RESTORED AREAS                                              *
                               7062 *      N/A                                                            *
                               7063 *                                                                     *
                               7064 *   MODIFICATION CONSIDERATIONS                                       *
                               7065 *      BPMLET IS CO-RESIDENT ON A SECTOR WITH BMINPT.                 *
                               7066 *      ANY MODIFICATION TO BPMLET WILL CHANGE THE ENTRY ADDRESS       *
                               7067 *      OF BMINPT AND MUST CONSIDER THE LIMITATION OF THE SECTOR       *
                               7068 *      BOUNDARY ON SIZE.                                              *
                               7069 *                                                                     *
                               7070 *   REQUIRED MODULES                                                  *
                               7071 *        @SYSEQ - COMMON SYSTEM EQUATES                               *
                               7072 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES    *
                               7073 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES       *
                               7074 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES                    *
                               7075 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES                         *
                               7076 *        @ERMEQ - ERROR MESSAGE EQUATES                               *
                               7077 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES                       *
                               7078 *        $B$EQU - COMPILER FIXED EQUATES                              *
                               7079 *        $B@EQU - COMPILER SYSTEM EQUATES                             *
                               7080 *                                                                     *
                               7081 *   OTHER                                                             *
                               7082 *      BPMLET IS ASSEMBLED WITH ALL OF OTHER STATEMENT PROCESSORS.    *
                               7083 ***********************************************************************

    1600                       7085        ORG   *,256,0                   BEGIN AT CORE PAGE BOUNDARY
                          1600 7086        USING *,@BR                     DEFINE BASE ADDR FOR CORE PAGE
                               7087 *
                               7088 * ENTER BPMLET - MULTIPLE ARITHMETIC 'LET' STATEMENT PROCESSOR
                               7089 *
                          1600 7090 BPMLET EQU   *                   BPMLET ENTRY POINT
                               7091 *
                               7092 * SKIP PAST 'LET' TO 1ST LIST ELEMENT SYMBOL CHARACTER
                               7093 *
    1600 3C 03 0873            7094 BPM010 MVI   B$NUMC,B@LLET       SET GET ROUTINE TO SKIP 'LET'
    1604 C0 87 0867            7095        B     B$GETC              LINK TO GET 1ST SYMBOL CHAR
                               7096 *
                               7097 * ENTER BPMASN - MULTIPLE ARITHMETIC ASSIGNMENT STATEMENT PROCESSOR
                               7098 *
                          1608 7099 BPMASN EQU *                           BPMASN ENTRY POINT
                               7100 *
                               7101 * GENERATE A BRANCH INSTRUCTION IMAGE - THIS INSTRUCTION IS REQUIRED
                               7102 * TO TRANSFER CONTROL PAST THE ASSIGNMENT ADDRESS STACKING SEQUENCE
                               7103 * TO THE SEQUENCE WHICH ESTABLISHES THE SOURCE FLOATING POINT VALUE
                               7104 *
    1608 D2 02 C5              7105 BPM020 LA    BPMBIC(,@BR),@XR          LOAD CADDR OF 'BRA' INSTR
    160B 34 02 0A40            7106        ST    B$PCAD,@XR                SET VADDR PARM FOR PUT RTN
    160F 3C 02 0A41            7107        MVI   B$PNBY,B@LBRA-1           SET LENGTH PARM FOR PUT RTN
    1613 C0 87 093A            7108        B     B$PUTC                    LINK TO OUTPUT THE  IMAGE
                               7109 *
                               7110 * STORE NEXT AVAILABLE PMC VIRTUAL ADDRESS (ADDRESS OF 1ST INSTRUCTION
                               7111 * IN THE ADDRESS STACKING SEQUENCE) AS OPERAND IN A 'RETURN BRANCH'
                               7112 * PSEUDO INSTRUCTION
                               7113 *
    1617 4C 01 CA 0A43         7114 BPM030 MVC   BPMBRO(,@BR),B$PVAD(@VADDR)  SET 'RETURN BRANCH' OPERAND
                               7115 *
                               7116 * ESTABLISH &WRK AS OPERAND OF A 'STACK FLOATING VALUE' INSTRUCTION
                               7117 *
    161C 4C 01 D0 15A0         7118 BPM040 MVC   BPMSFO(,@BR),B$WORK(@VADDR)  SET 'STF' OPERAND &WRK
                               7119 *
                               7120 * GENERATE ADDRESS STACKING INSTRUCTIONS FOR AN ASSIGNMENT LIST ELEMENT
                               7121 *
    1621 35 02 0878            7122 BPM045 L     B$GPTR,@XR                RESTORE TEXT POINTER
    1625 C0 87 1853            7123 BPM050 B     B$LIST                    LINK TO PROCESS LIST ELEMENT
    1629 6C 00 4C 00           7124        MVC   BPM070+@Q(,@BR),B@CHAR(1,@XR)  SAVE CADDR OF NEXT CHAR
                               7125 *
                               7126 * GENERATE PSEUDO INSTRUCTIONS TO STACK THE SOURCE VALUE AND UNSTACK
                               7127 * IT TO THE ASSIGNMENT LIST ELEMENT ADDRESS
                               7128 *
    162D D2 02 CE              7129 BPM060 LA    BPMSFC(,@BR),@XR          LOAD CADDR OF 'STF' INSTR
    1630 34 02 0A40            7130        ST    B$PCAD,@XR                SET VADDR PARM FOR PUT RTN
    1634 3C 02 0A41            7131        MVI   B$PNBY,B@LSTF-1           SET LENGTH PARM FOR PUT RTN
    1638 C0 87 093A            7132        B     B$PUTC                    LINK TO OUTPUT 'STF URIC'
    163C D2 02 D1              7133        LA    BPMUFC(,@BR),@XR          LOAD CADDR OF 'UV' INSTR
    163F 34 02 0A40            7134        ST    B$PCAD,@XR                SET VADDR PARM FOR PUT RTN
    1643 3C 00 0A41            7135        MVI   B$PNBY,B@LUSF-1           SET LENGTH PARM FOR PUT RTN
    1647 C0 87 093A            7136        B     B$PUTC                    LINK TO OUTPUT 'USF' INST
                               7137 *
                               7138 * TEST FOR END OF THE MULTIPLE ASSIGNMENT LIST
                               7139 *
    164B 7D 00 D2              7140 BPM070 CLI   BPMIND(,@BR),*-*          IF LIST DELIMITER IS
    164E F2 81 07              7141        JE    BPM090                    * EXIT LIST PROCESSING LOOP
                               7142 *
                               7143 * ADVANCE TEXT POINTER PAST LIST DELIMITER AND BRANCH TO PROCESS
                               7144 * NEXT ELEMENT IN THE ASSIGNMENT LIST
                               7145 *
    1651 C0 87 0867            7146 BPM080 B     B$GETC                    LINK TO GET NEXT CHARACTER
    1655 D0 87 25              7147        B     BPM050(,@BR)              GO PROCESS NEXT LIST ELEMENT
                               7148 *
                               7149 * GENERATE A BRANCH INSTRUCTION IMAGE - THIS INSTRUCTION IS REQUIRED
                               7150 * TO TRANSFER CONTROL PAST THE SEQUENCE WHICH ESTABLISHES THE SOURCE
                               7151 * VALUE TO THE STATEMENT FOLLOWING THAT WHICH IS BEING PROCESSED
                               7152 *
    1658 D2 02 C5              7153 BPM090 LA    BPMBIC(,@BR),@XR          LOAD CADDR OF 'BRA' INSTR
    165B 34 02 0A40            7154        ST    B$PCAD,@XR                SET VADDR PARM FOR PUT RTN
    165F 3C 02 0A41            7155        MVI   B$PNBY,B@LBRA-1           SET LENGTH PARM FOR PUT RTN
    1663 C0 87 093A            7156        B     B$PUTC                    LINK TO OUTPUT THE 'BRA' IMAGE
                               7157 *
                               7158 * ESTABLISH CONDITIONS TO RESOLVE THE ADDRESS OPERAND IN THE FIRST
                               7159 * BRANCH INSTRUCTION IMAGE GENERATED ABOVE (BPM020)
                               7160 *
    1667 1C 01 19EF CA         7161 BPM100 MVC   B$BRVA,BPMBRO(@VADDR,@BR)  SET BRANCH TABLE VADDR PARM
    166C 1F 01 19EF C4         7162        SLC   B$BRVA,BPMBN1(@VADDR,@BR)  * FOR THE 'BRA' IMAGE OPERAND
    1671 0C 01 19F1 0A43       7163        MVC   B$BRLN,B$PVAD(@VADDR)     SET BRANCH TABLE LINE NO. PARM
                               7164 *                                      * FOR BRANCH POINT VADDR
    1677 C0 87 1996            7165        B     B$BTAB                    LINK TO SET UP RESOLUTION
                               7166 *
                               7167 * GENERATE INSTRUCTION TO STACK ADDRESS OF &WRK - THE FIRST BRANCH
                               7168 * INSTRUCTION (BPM020) PASSES RUN-TIME CONTROL TO THIS INSTRUCTION
                               7169 *
    167B 5C 01 CD D0           7170 BPM110 MVC   BPMSAO(,@BR),BPMSFO(@VADDR,@BR)  SET 'STA' OPERAND &WRK
    167F D2 02 CB              7171        LA    BPMSAC(,@BR),@XR          LOAD CADDR OF 'STA' INSTR
    1682 34 02 0A40            7172        ST    B$PCAD,@XR                SET VADDR PARM FOR PUT RIP
    1686 3C 02 0A41            7173        MVI   B$PNBY,B@LSTA-1           SET LENGTH PARM FOR PUT RTN
    168A C0 87 093A            7174        B     B$PUTC                    LINK TO OUTPUT 'STA MARK'
                               7175 *
                               7176 * GENERATE PSEUDO INSTRUCTIONS TO PROCESS THE STATEMENT EXPRESSION
                               7177 * AND UNSTACK THE RESULTING VALUE INTO &WRK
                               7178 *
    168E C0 87 1514            7179 BPM120 B     B$SCAN                    LINK TO GENERATE EXPRESSION PMC
    1692 D2 02 D1              7180        LA    BPMUFC(,@BR),@XR          LOAD CADDR OF 'USF' INSTR
    1695 34 02 0A40            7181        ST    B$PCAD,@XR                SET VADDR PARM FOR PUT RTN
    1699 3C 00 0A41            7182        MVI   B$PNBY,B@LUSF-1           SET LENGTH PARM FUR PUT RTN
    169D C0 87 093A            7183        B     B$PUTC                    LINK TO OUTPUT 'USF' INST
                               7184 *
                               7185 * GENERATE THE RETURN BRANCH INSTRUCTION - THIS TRANSFERS CONTROL
                               7186 * TO THE LIST ASSIGNMENT SEQUENCE AFIER THE SOURCE VALUE HAS BEEN
                               7187 * STORED IN INTERNAL VARIABLE MIRK
                               7188 *
    16A1 D2 02 C8              7189 BPM130 LA    BPMBRC(,@BR),@XR          LOAD CADDR OF 'BRA' INSTR
    16A4 34 02 0A40            7190        ST    B$PCAD,@XR                SET VADDR PARM FOR PUT RTN
    16A8 3C 02 0A41            7191        MVI   B$PNBY,B@LBRA-1           SET LENGTH PARM FOR PUT RTN
    16AC C0 87 093A            7192        B     B$PUTC                    LINK TO OUTPUT RETURN 'BRA'
                               7193 *
                               7194 * ESTABLISH CONDITIONS TO RESOLVE THE ADDRESS OPERAND IN THE SECOND
                               7195 * BRANCH INSTRUCTION IMAGE GENERATED ABOVE (BPM090)
                               7196 *
    16B0 0C 01 19EF 19F1       7197 BPM140 MVC   B$BRVA,B$BRLN(@VADDR)     SET BRANCH TABLE VADDR PARM
    16B6 1F 01 19EF C4         7198        SLC   B$BRVA,BPMBN1(@VADDR,@BR) * FOR THE 'BRA' IMAGE OPERAND
    16BB 3A 07 071D            7199        SBN   B$NXSW,B$NXMK             SET 'NEXT STMNT' SNITCH ON TO
                               7200 *                                      * ESTABLISH LINE NO. PARM
                               7201 *
                               7202 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               7203 *
    16BF C0 87 0700            7204 BPM150 B     B$DIST                    BRANCH TO DISTRIBUTOR
```

```
                               7206 ***********************************************************************
                               7207 * MULTIPLE ARITHMETIC 'LET' ROUTINE CONSTANTS
                               7208 ***********************************************************************
                               7209 *
    16C3 0001             16C4 7210 BPMBN1 DC    IL(@VADDR)'1'             BINARY INTEGER -1
                               7212 ***********************************************************************
                               7213 * MULTIPLE ARITHMETIC 'LET' ROUTINE PMC AND STORAGE PARAMETERS
                               7214 ***********************************************************************
                               7215 *
    16C5 46               16C5 7216 BPMBIC DC    AL(B@LCOP)(B@CBRA)        BRANCH IMAGE 'BRA' OPCODE
    16C6 0000             16C7 7217 BPMBIO DC    XL(B@LCVA)'00'            BRANCH IMAGE NULL OPERAND
                               7218 *
    16C8 46               16C8 7219 BPMBRC DC    AL(B@LCOP)(B@CBRA)        RETURN BRANCH 'BRA' OPCODE
    16C9                  16CA 7220 BPMBRO DS    CL(B@LCVA)                RETURN BRANCH OPERAND AREA
                               7221 *
    16CB 34               16CB 7222 BPMSAC DC    AL(B@LCOP)(B@CSTA)        STACK ADDRESS 'STA' OPCODE
    16CC                  16CD 7223 BPMSAO DS    CL(B@LCVA)                STACK ADDRESS OPERAND AREA
                               7224 *
    16CE 20               16CE 7225 BPMSFC DC    AL(B@LCOP)(B@CSTF)        STACK FLOATING 'STF' OPCODE
    16CF                  16D0 7226 BPMSFO DS    CL(B@LCVA)                STACK FLOATING OPERAND AREA
                               7227 *
    16D1 26               16D1 7228 BPMUFC DC    AL(B@LCOP)(B@CUSF)        UNSTACK FLOATING 'USF' OPCODE
                               7229 *
    16D2 7E               16D2 7230 BPMIND DC    AL1(B@EQUL)               DELIMITER COMPARE - '='
                               7231 ***********************************************************************
                               7232 *
                               7233 * END OF MULTIPLE ARITHMETIC 'LET' ROUTINE CODING
                               7234 *
                               7236 ***********************************************************************
                               7237 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               7238 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               7239 *                                                                     *
                               7240 ***********************************************************************
                               7241 *STATUS                                                               *
                               7242 *   VERSION 1 MODIFICATION 0                                          *
                               7243 *                                                                     *
                               7244 *FUNCTION                                                             *
                               7245 *   BMINPT IS EXECUTED TO TRANSLATE MAT INPUT STATEMENTS AS THEY OCCUR*
                               7246 *   IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE   *
                               7247 *   THE PSEUDOCODE IN VIRTUAL MEMORY.                                 *
                               7248 *                                                                     *
                               7249 *ENTRY POINTS                                                         *
                               7250 *   BMINPT HAS ONLY ONE ENTRY POINT:                                  *
                               7251 *        BMINPT - TRANSLATE MAT INPUT STATEMENT                       *
                               7252 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               7253 *        B     BMINPT                                                 *
                               7254 *                                                                     *
                               7255 *INPUT                                                                *
                               7256 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               7257 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               7258 *     LEADING KEYWORD, MAT INPUT.                                     *
                               7259 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                               7260 *     CHARACTER IN THE LEADING KEYWORD, MAT INPUT.                    *
                               7261 *                                                                     *
                               7262 *OUTPUT                                                               *
                               7263 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUE4CE        *
                               7264 *     GENERATED BY BMINPT IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               7265 *     SEQUENCES.                                                      *
                               7266 *   * TEXT CHARACTER POINTER - CONTAINS THE ARE ADDRESS OF THE        *
                               7267 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               7268 *                                                                     *
                               7269 *EXTERNAL REFERENCES                                                  *
                               7270 *        B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.*
                               7271 *        B$PUTC - (B$PCAI, B$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY *
                               7272 *                 ROUTINE.                                            *
                               7273 *        B$MATR - ENTRY TO BASIC MATRIX REFERENCE ROUTINE.            *
                               7274 *        B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                *
                               7275 *                                                                     *
                               7276 *EXITS, NORMAL                                                        *
                               7277 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               7278 *                                                                     *
                               7279 *EXITS, ERROR                                                         *
                               7280 *   N/A                                                               *
                               7281 *                                                                     *
                               7282 *TABLES/WORK AREAS                                                    *
                               7283 *   N/A                                                               *
                               7284 *                                                                     *
                               7285 *ATTRIBUTES                                                           *
                               7286 *   BMINPT IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               7287 *                                                                     *
                               7288 *CHARACTER CODE DEPENDENCY                                            *
                               7289 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON ANY PARTICULAR  *
                               7290 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               7291 *                                                                     *
                               7292 *NOTES                                                                *
                               7293 *   ERROR PROCEDURES                                                  *
                               7294 *      N/A                                                            *
                               7295 *                                                                     *
                               7296 *   REGISTER USAGE                                                    *
                               7297 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DUHING EXECUTION.   *
                               7298 *                                                                     *
                               7299 *   SAVED/RESTORED AREAS                                              *
                               7300 *      N/A                                                            *
                               7301 *                                                                     *
                               7302 *   MODIFICATION CONSIDERATIONS                                       *
                               7303 *      BMINPT RESIDES ON A SECTOR WITH OPITET.  ANY MODIFICATION   1-4*
                               7304 *      SHOULD CONSIDER THE SECTOR BOUNDARY LIMITATION ON SIZE.     1-4*
                               7305 *                                                                     *
                               7306 *   REQUIRED MODULES                                                  *
                               7307 *        @SYSEQ - COMMON SYSTEM EQUATES.                              *
                               7308 *        @FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUE EQUATES. *
                               7309 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.      *
                               7310 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               7311 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               7312 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               7313 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               7314 *        $B$EQU - CCRPILER FIXED EQUATES.                             *
                               7315 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               7316 *                                                                     *
                               7317 *   OTHER                                                             *
                               7318 *      BMINPT IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               7319 ***********************************************************************

                               7321 *
                               7322 * ENTER BMINPT - MAT INPUT STATEMENT ROUTINE
                               7323 *
                          16D3 7324 BMINPT EQU   *                         BMINPT ENTRY POINT
                               7325 *
                               7326 * SET GET ROUTINE TO SKIP TO 'T' IN KEYWORDS 'MAT INPUT'
                               7327 *
    16D3 3C 07 0873            7328 BMI010 MVI   B$NUMC,B@LMIN-1           SET GET TO SKIP TO 'T' IN INPUT
    16D7 C0 87 0867            7329        B     B$GETC                    LINK TO ADVANCE POINTER
                               7330 *
                               7331 * CALL MATRIX REFERENCE PROCESSOR TO GENERATE DOPE VECTOR STACKING
                               7332 * INSTRUCTIONS IN VIRTUAL MEMORY
                               7333 *
    16DB C0 87 18F3            7334 BMI020 B     B$MATR                    LINK TO PROCESS MAT-REFERENCE
                               7335 *
                               7336 * GENERATE 'MF1' INSTRUCTION TO INDICATE INPUT IN VIRTUAL MEMORY
                               7337 *
    16DF D2 02 FC              7338 BMI030 LA    BMIMFC(,@BR),@XR          LOAD CADDR OF 'MF1' INSTR
    16E2 34 02 0A40            7339        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR MF1
    16E6 3C 02 0A41            7340        MVI   B$PNBY,B@LMF1-1           SET LNG PARM OF PUT FOR MFT
    16EA C0 87 093A            7341        B     B$PUTC                    LINK TO GENERATE PMC
                               7342 *
                               7343 * TEST DELIMITER FOR BEING A STATEMENT TERMINATOR
                               7344 *
    16EE 35 02 0878            7345 BMI040 L     B$GPTR,@XR                RESTORE TEXT POINTER
    16F2 BD 1E 00              7346        CLI   B@CHAR(,@XR),B@EOST       IF DELIMITER IS NOT AN EOS
    16F5 D0 01 DB              7347        BNE   BMI020(,@BR)              * GO PROCESS NEXT MAT-REFERENCE
```

```
                               7348 *
                               7349 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               7350 *
    16F8 C0 87 0700            7351 BMI050 B     B$DIST                    RETURN TO DISTRIBUTOR

                               7353 ***********************************************************************
                               7354 * MAT INPUT STATEMENT ROUTINE STORAGE AND PARAMETER AREAS
                               7355 ***********************************************************************
                               7356 *
    16FC 18               16FC 7357 BMIMFC DC  AL(B@LCOP)(B@CMF1)          'MF1' INSTR OPCODE
    16FD 3D00             16FE 7358 BMIMFO DC  AL(B@LCVA)(V$XMIN)          'MF1' INSTR OPND - INPUT
                               7359 *
                               7360 ***********************************************************************
                               7361 *
                               7362 * END OF 'MAT INPUT' STATEMENT ROUTINE CODING
                               7363 *
                               7365 ***********************************************************************
                               7366 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               7367 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               7368 *                                                                     *
                               7369 ***********************************************************************
                               7370 *STATUS                                                               *
                               7371 *   VERSION 1 MODIFICATION 0                                          *
                               7372 *                                                                     *
                               7373 *FUNCTION                                                             *
                               7374 *   BNIMAG IS EXECUTED TO TRANSLATE IMAGE STATEMENTS AS THEY OCCUR    *
                               7375 *   IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE   *
                               7376 *   THE PSEUDOCODE IN VIRTUAL MEMORY.                                 *
                               7377 *                                                                     *
                               7378 *ENTRY POINTS                                                         *
                               7379 *   BNIMAG HAS ONLY ONE ENTRY POINT:                                  *
                               7380 *        BNIMAG - TRANSLATE IMAGE STATEMENT                           *
                               7381 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               7382 *        B     BNIMAG                                                 *
                               7383 *                                                                     *
                               7384 *INPUT                                                                *
                               7385 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               7386 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               7387 *     LEADING KEYWORD, ':'.                                           *
                               7388 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                               7389 *     CHARACTER IN THE LEADING KEYWORD, ':'.                          *
                               7390 *   * B$ERSW - THE COMPILER MODE SWITCH.  THIS SWITCH, TESTED USING   *
                               7391 *     MASK B$ERMK, INDICATES COMPILER ERROR MODE WHEN ON.             *
                               7392 *                                                                     *
                               7393 *OUTPUT                                                               *
                               7394 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               7395 *     GENERATED BY BNIMAG IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               7396 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               7397 *     SEQUENCES.                                                      *
                               7398 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               7399 *     CHARACTER WHICH FOLLOWS THE END-OF-STATEMENT CHARACTER IN THE   *
                               7400 *     IMAGE STATEMENT.                                                *
                               7401 *   * B$BRVA - CONTAINS THE VIRTUAL ADDRESS OF THE RIGHT BYTE OF THE  *
                               7402 *     ADDRESS OPERAND FIELD IN THE STATEMENT BYPASS BRANCH            *
                               7403 *     INSTRUCTION.                                                    *
                               7404 *   * B$NXSN - SET TO ON STATUS TO CAUSE RESOLUTION OF THE STATEMENT  *
                               7405 *     BYPASS BRANCH INSTRUCTION OPERAND BY THE COMPILER DISTRIBUTOR.  *
                               7406 *                                                                     *
                               7407 *EXTERNAL REFERENCES                                                  *
                               7408 *     B$GETC - (B$NUMC) - ENTRY TO BASIC RETRIEVAL ROUTINE.           *
                               7409 *     B$PUTC - (B$PCAD, B$PNBY, B$PVAD, B$PBNL, B$ERSW) - ENTRY TO    *
                               7410 *              COMPILER VIRTUAL MEMORY OUTPUT ROUTINE.                *
                               7411 *     B$FCON - (B$CTYP, B$BCKT, B$CPCT) - ENTRY TO BASIC COMPILER     *
                               7412 *              CONSTANT ROUTINE.                                      *
                               7413 *     B UTAB - (B$BRVA) - ENTRY TO COMPILER BRANCH TABLE ROUTINE.     *
                               7414 *     B$DIST - (B$NISW, B$LINE) - ENTRY TO BASIC COMPILER DISTRIBUTOR *
                               7415 *                                                                     *
                               7416 *EXITS, NORMAL                                                        *
                               7417 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               7418 *                                                                     *
                               7419 *EXITS, ERROR                                                         *
                               7420 *   N/A                                                               *
                               7421 *                                                                     *
                               7422 *TABLES/WORK AREAS                                                    *
                               7423 *   N/A                                                               *
                               7424 *                                                                     *
                               7425 *ATTRIBUTES                                                           *
                               7426 *   BNIMAG IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               7427 *                                                                     *
                               7428 *CHARACTER CODE DEPENDENCY                                            *
                               7429 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR   *
                               7430 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.           *
                               7431 *                                                                     *
                               7432 *NOTES                                                                *
                               7433 *   ERROR PROCEDURES                                                  *
                               7434 *      N/A                                                            *
                               7435 *                                                                     *
                               7436 *   REGISTER USAGE                                                    *
                               7437 *      BOTH THE INDEX AND BASE REGISTERS ARC USED DURING EXECUTION.   *
                               7438 *                                                                     *
                               7439 *   SAVED/RESTORED AREAS                                              *
                               7440 *      N/A                                                            *
                               7441 *                                                                     *
                               7442 *   MODIFICATION CONSIDERATIONS                                       *
                               7443 *      BNIMAG IS CO-RESIDENT ON A SECTOR WITH BMREAD. ANY          1-4*
                               7444 *      MODIFICATION TO BNIMAG WILL CHANGE THE ENTRY ADDRESS OF     1-4*
                               7445 *      BMREAD AND MUST CONSIDER THE LIMITATION OF THE SECTOR       1-4*
                               7446 *      BOUNDARY ON SIZE.                                           1-4*
                               7447 *                                                                     *
                               7448 *   REQUIRED MODULES                                                  *
                               7449 *         @SYSEQ - COMMON SYSTEM EQUATES.                             *
                               7450 *         @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.  *
                               7451 *         @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS.             *
                               7452 *         @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                  *
                               7453 *         @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                       *
                               7454 *         @ERMEQ - ERROR MESSAGE EQUATES.                             *
                               7455 *         $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                     *
                               7456 *         $B$EQU - COMPILER FIXED EQUATES.                            *
                               7457 *         $B@EQU - COMPILER SYSTEM EQUATES.                           *
                               7458 *                                                                     *
                               7459 *   OTHER                                                             *
                               7460 *      BNIMAG IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               7461 ***********************************************************************

    1700                       7463        ORG   *,256,0                   BEGIN AT CORE PAGE BOUNDARY
                          1700 7464        USING *,@BR                     DEFINE BASE ADDR FOR CORE PAGE
                               7465 *
                               7466 * ENTER BNIMAG - 'IMAGE' STATEMENT ROUTINE
                               7467 *
                          1700 7468 BNIMAG EQU   *                         BNIMAG ENTRY POINT
                               7469 *
                               7470 * REPLACE IMAGE STATEMENT 'STH' PSEUDO INSTRUCTION WITH SPECIAL
                               7471 * IMAGE STATEMENT HEADER ('IMH') INSTRUCTION - INSTRUCTION REPLACEMENT
                               7472 * IS NOT PERFORMED WHEN THE COMPILER IS OPERATING IN ERROR MODE
                               7473 *
    1700 38 07 0993            7474        TBN   B$ERSW,B$ERMK             TEST ERROR SWITCH - BYPASS SIN
    1704 F2 10 1E              7475        JT    BNI005                    * REPLACEMENT IF COMPILER ERRS
    1707 1E 00 0A01 CC         7476        ALC   B$PBNL,BNISHL(1,@BR)      ADJUST INIC DUFFER POINTERS TO
    170C 1F 00 0A43 CC         7477        SLC   B$PVAD,BNISHL(1,@BR)      * DELETE LAST 'STH' PSEUDO INST
    1711 4C 01 C0 07D0         7478        MVC   BNIIHO(,@BR),B$LINE(B@LCLN)  SET 'IMH' OPERAND = LINE NO.
    1716 D2 02 BE              7479        LA    BNIIMH(,@BR),@XR          LOAD 'IMH' INSTRUCTION CADDR
    1719 34 02 0A40            7480        ST    B$PCAD,@XR                SET 'PUT' RTNPARM FOR 'INH'
    171D 3C 02 0A41            7481        MVI   B$PNBY,B@LIMH-1           SET 'PUT' RTN LENGTHPARM
    1721 C0 87 093A            7482        B     B$PUTC                    LINK TO PUT THE 'IMH' INST
                               7483 *
                               7484 * 'ADVANCE' CHARACTER POINTER TO LAST CHARACTER OF IMAGE 'KEYWORD'
                               7485 *
    1725 3C 00 0873            7486 BNI005 MVI   B$NUMC,B@LIMG-1           SET GETPARM TO SKIP KEYWORD
    1729 C0 87 0867            7487        B     B$GETC                    LINK TO GET LAST KEYWORD CHAR
                               7488 *
                               7489 * GENERATE A 'BRA' IMAGE INSTRUCTION IN VIRTUAL MEMORY
                               7490 *
    172D D2 02 C1              7491 BNI010 LA    BNIBRC(,@BR),@XR          LOAD CADDR OF 'BRA' INSTR
    1730 34 02 0A40            7492        ST    B$PCAD,@XR                SET PUT RTN VADDRPARM FOR BRA
    1734 3C 02 0A41            7493        MVI   B$PNBY,B@LBRA-1           SET PUT RTN LENGTHPARM FOR BRA
    1738 C0 87 093A            7494        B     B$PUTC                    LINK TO GENERATE 'BRA' INSTR
                               7495 *
                               7496 *  ESTABLISH 'BRA' OPERAND FOR ADDRESS RESOLUTION
                               7497 *
    173C 0C 01 19EF 0A43       7498 BNI020 MVC   B$BRVA,B$PVAD(@VADDR)      SET BRA TABLE FOR 'BRA' VADOR
    1742 1F 01 19EF CB         7499        SLC   B$BRVA,BNIBN1(@VADDR,@BR)  ADJUST VADDR TO 'BRA' OPERAND
                               7500 *
                               7501 *  SET THE TEXT POINTER TO REFERENCE A DUMMY TERMINATOR
                               7502 *
    1747 D2 02 CD              7503 BNI030 LA    BNIEOS(,@BR),@XR          SET PTR TO DUMMY TERMINATOR
                               7504 *
                               7505 * CALL THE CONSTANT ROUTINE TO GENERATE THE CHARACTER STRING
                               7506 *
    174A 3C 1B 0A5F            7507 BNI040 MVI   B$CTYP,B$SCON             SET CON RTN FOR CHAR STRING
    174E C0 87 0A46            7508        B     B$FCON                    LINK TO GENERATE CHAR STRING
    1752 3C 00 0873            7509        MVI   B$NUMC,B@GETS             DISABLE THE GET ROUTINE
                               7510 *
                               7511 * TEST FOR THIS BEING A NULL STRING
                               7512 *
    1756 3D 00 0CA8            7513 BNI050 CLI   B$CPCT,@ZERO              IF THIS WAS NOT A NULL STRING
    175A F2 01 29              7514        JNE   BNI110                    * GO GENERATE STC/PRU SEQUENCE
                               7515 *
                               7516 *  MOVE A CODE OF '01' TO THE 'PRU' INSTR OPERAND TO INDICATE THAT THE
                               7517 *  STATEMENT CONTAINS NO IMAGE SPECIFICATIONS
                               7518 *
    175D 7C 01 C5              7519 BNI060 MVI   BNIPRO(,@BR),B@PUI0       SET 'PRU' CODE TO ONE
                               7520 *
                               7521 * GENERATE THE 'PRU' INSTRUCTION IN VIRTUAL MEMORY
                               7522 *
    1760 D2 02 C4              7523 BNI070 LA    BNIPRC(,@BR),@XR          LOAD CADDR OF 'PRU' INSTR
    1763 34 02 0A40            7524        ST    B$PCAD,@XR                SET PUT RTN VADDRPARM FOR PRU
    1767 3C 01 0A41            7525        MVI   B$PNBY,B@LPRU-1           SET PUT RTN LNG PARM, FOR PRU
    176B C0 87 093A            7526        B     B$PUTC                    LINK TO GENERATE 'PRU' INSTR
                               7527 *
                               7528 * GENERATE A 'BRS' INSTRUCTION IN VIRTUAL MEMORY
                               7529 *
    176F D2 02 C9              7530 BNI080 LA    BNIBSC(,@BR),@XR          LOAD CADDR OF 'BRS' INSTR
    1772 34 02 0A40            7531        ST    B$PCAD,@XR                SET PUT RTN VADDR PARM FOR 'BRS'
    1776 3C 00 0A41            7532        MVI   B$PNBY,B@LBRS-1           SET PUT RTN LNGPARM FOR 'BRS'
    177A C0 87 093A            7533        B     B$PUTC                    LINK TO GENERATE 'BRS' INSTR
                               7534 *
                               7535 * SET DISTRIBUTOR TO SET UP RESOLUTION CONDITIONS FOR 'BRA' OPERAND
                               7536 *
    177E 3A 07 071D            7537 BNI090 SBN   B$NXSW,B$NXMK             SET 'NEXT' SNITCH ON
                               7538 *
                               7539 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               7540 *
    1782 C0 87 0700            7541 BNI100 B     B$DIST                    RETURN TO DISTRIBUTOR
                               7542 *
                               7543 * IF THIS IS A CHARACTER STRING MOVE THE VADDR OF THE 1ST SEGMENT TO AN
                               7544 * 'STC' INSTRUCTION OPERAND
                               7545 *
    1786 4C 01 C8 1590         7546 BNI110 MVC   BNISTO(,@BR),B$BCKT(@VADDR)  SET 'STC' OPERAND FOR VADDR
                               7547 *
                               7548 * MOVE A CODE OF '04' TO THE 'PRU' INSTR OPERAND TO INDICATE THAT THE
                               7549 * FIRST CHARACTER CONSTANT IS ESTABLISHED FOR THE IMAGE SPECIFICATION
                               7550 *
    178B 7C 04 C5              7551 BNI120 MVI   BNIPRO(,@BR),B@PUI1       SET 'PRU' CODE TO FOUR
                               7552 *
                               7553 * GENERATE AN 'STC' INSTRUCTION IN VIRTUAL MEMORY
                               7554 *
    178E D2 02 C6              7555 BNI130 LA    BNISTC(,@BR),@XR          LOAD CADDR OF 'STC' INSTR
    1791 34 02 0A40            7556        ST    B$PCAD,@XR                SET PUT RTN VADDRPARM FOR SIC
    1795 3C 02 0A41            7557        MVI   B$PNBY,B@LSTC-1           SET PUT RTN LNGPARM FOR STC
    1799 C0 87 093A            7558        B     B$PUTC                    LINK TO GENERATE 'STC' INSTR
                               7559 *
                               7560 * GENERATE THE 'PRU' INSTRUCTION IN VIRTUAL MEMORY
                               7561 *
    179D D2 02 C4              7562 BNI140 LA    BNIPRC(,@BR),@XR          LOAD CADDR OF 'PRU' INSTR
    17A0 34 02 0A40            7563        ST    B$PCAD,@XR                SET PUT RTN VADDRPARM FOR PRU
    17A4 3C 01 0A41            7564        MVI   B$PNBY,B@LPRU-1           SET PUT RTN LNGPARM FOR PRU
    17A8 C0 87 093A            7565        B     B$PUTC                    LINK TO GENERATE 'PRU' INSTR
                               7566 *
                               7567 * MOVE A CODE OF '05' TO THE 'PRU' INSTR OPERAND TO INDICATE THAT THE
                               7568 * CHARACTER CONSTANT IS ANY ESTABLISHED FOR THE IMAGE SPECIFICATION
                               7569 * EXCEPT THE FIRST
                               7570 *
    17AC 7C 05 C5              7571 BNI150 MVI   BNIPRO(,@BR),B@PUI2       SET THE PRU CODE TO FIVE
                               7572 *
                               7573 * SUBTRACT THE LENGTH OF A STRING SEGMENT FROM 'STC' INSIR OPERAND
                               7574 *
    17AF 5F 01 C8 CF           7575 BNI160 SLC   BNISTO(,@BR),BNISUB(@VADDR,@BR)  SUB LNG OF STRING SEGMENT
                               7576 *
                               7577 * TEST FOR THE PRESENCE OF OTHER STRING SEGMENTS
                               7578 *
    17B3 1F 00 0CA8 CB         7579 BNI170 SLC   B$CPCT,BNIBN1(1,@BR)      IF OTHER SEGMENTS ARE PRESENT
    17B8 D0 84 8E              7580        BH    BNI130(,@BR)              * BRANCH TO GENERATE 'STC'
                               7581 *
                               7582 * IF OTHER SEGMENTS ARE NOT PRESENT BRANCH TO GENERATE THE 'BRS' INSTR
                               7583 *
    17BB D0 87 6F              7584 BNI180 B     BNI080(,@BR)              BRANCH TO GENERATE 'BRS' INSTR
```

```
                               7586 ***********************************************************************
                               7587 * 'IMAGE' STATEMENT ROUTINE STORAGE AND PARAMETER AREA
                               7588 ***********************************************************************
                               7589 *
    17BE 66                17BE 7590 BNIIMH DC    AL(B@LCOP)(B@CIMH)        'IMH' INSTRUCTION OPCODE
    17BF                   17C0 7591 BNIIHO DS    CL(B@LCLN)                'IMH' INSTRUCTION OPERAND
                               7592 *
    17C1 46                17C1 7593 BNIBRC DC    AL(B@LCOP)(B@CBRA)        'BRA' INSTR OPCODE
    17C2 0000              17C3 7594 BNIBRO DC    XL(B@LCVA)'00'            'BRA' INSTR OPERAND
                               7595 *
    17C4 62                17C4 7596 BNIPRC DC    AL(B@LCOP)(B@CPRU)        'PRU' INSTR OPCODE
    17C5                   17C5 7597 BNIPRO DS    CL(B@LCXX)                'PRU' INSTR OPERAND
                               7598 *
    17C6 28                17C6 7599 BNISTC DC    AL(B@LCOP)(B@CSTC)        'STC' INSTR OPCODF
    17C7                   17C8 7600 BNISTO DS    CL(@VADDR)                'STC' INSTR OPERAHD
                               7601 *
    17C9 4C                17C9 7602 BNIBSC DC    AL(B@LCOP)(B@CBRS)        'BRS' INSTR OPCODE *

                               7604 ***********************************************************************
                               7605 * 'IMAGE' STATEMENT ROUTINE CONSTANTS
                               7606 *
    17CA 0001              17CB 7607 BNIBN1 DC    IL(@VADDR)'1'             BINARY 1
    17CC 03                17CC 7608 BNISHL DC    AL1(B@LSTH)               LENGTH OF 'STH' INSTRUCTION
    17CD 1E                17CD 7609 BNIEOS DC    AL1(B@EOST)               DUMMY TERMINATOR
    17CE 0013              17CF 7610 BNISUB DC    AL(@VADDR)(B@LCRV)        LENGTH OF STRING SEGMENT
                               7611 *
                               7612 ***********************************************************************
                               7613 *
                               7614 * END OF 'IMAGE' STATEMENT ROUTINE CODING
                               7615 *
                               7617 ************************************************************************
                               7618 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                   *
                               7619 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               7620 *                                                                      *
                               7621 ************************************************************************
                               7622 *STATUS                                                                *
                               7623 *   VERSION 1 MODIFICATION 0                                           *
                               7624 *                                                                      *
                               7625 *FUNCTION                                                              *
                               7626 *   BNREAD IS EXECUTED TO TRANSLATE MAT READ STATEMENTS AS THEY OCCUR *
                               7627 *   IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE   *
                               7628 *   THE PSEUDOCODE IN VIRTUAL MEMORY.                                  *
                               7629 *                                                                      *
                               7630 *ENTRY POINTS                                                          *
                               7631 *   BMREAD HAS ONLY ONE ENTRY POINT:                                   *
                               7632 *        BMREAD - TRANSLATE MAT READ STATEMENT                         *
                               7633 *   THE FORMAT OF THE CALLING SEQUENCE IS:                             *
                               7634 *        B     BMREAD                                                  *
                               7635 *                                                                      *
                               7636 *INPUT                                                                 *
                               7637 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               7638 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE        *
                               7639 *     LEADING KEYWORD, MAT READ.                                       *
                               7640 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                               7641 *     CHARACTER IN TIE LEADING KEYWORD, MAT READ.                      *
                               7642 *                                                                      *
                               7643 *OUTPUT                                                                *
                               7644 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE         *
                               7645 *     GENERATED BY BMREAD IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               7646 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               7647 *     SEQUENCES.                                                       *
                               7648 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               7649 *     CHARACTER WHICH TERMINATES THE STATEMENT.                        *
                               7650 *                                                                      *
                               7651 *EXTERNAL REFERENCES                                                   *
                               7652 *     B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.   *
                               7653 *     B$PUTC - (B$PCAD, B$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY    *
                               7654 *              OUTPUT ROUTINE.                                         *
                               7655 *     B$MATR - ENTRY TO BASIC COMPILER MATRIX REFERENCE ROUTINE.      *
                               7656 *     B$DIST - ENTRY TO BASIC COMPILER DISTRUBUTOR.                    *
                               7657 *                                                                      *
                               7658 *EXITS, NORMAL                                                         *
                               7659 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTGE.                      *
                               7660 *                                                                      *
                               7661 *EXITS, ERROR                                                          *
                               7662 *   N/A                                                                *
                               7663 *                                                                      *
                               7664 *TABLES/WORK AREAS                                                     *
                               7665 *   N/A                                                                *
                               7666 *                                                                      *
                               7667 *ATTRIBUTES                                                            *
                               7668 *   BMREAD IS NATURALLY RELOCATABLE AND REUSABLE.                      *
                               7669 *                                                                      *
                               7670 *CHARACTER CODE DEPENDENCY                                             *
                               7671 *   THE OPERATION OF THIS MODUE DOES NOT DEPEND UPON A PARTICULAR     *
                               7672 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.             *
                               7673 *                                                                     *
                               7674 *NOTES                                                                *
                               7675 *   ERROR PROCEDURES                                                  *
                               7676 *      N/A                                                            *
                               7677 *                                                                     *
                               7678 *   REGISTER USAGE                                                    *
                               7679 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               7680 *                                                                     *
                               7681 *   SAVED/RESTORED AREAS                                              *
                               7682 *      N/A                                                            *
                               7683 *                                                                     *
                               7684 *   MODIFICATION CONSIDERATIONS                                       *
                               7685 *      BMREAD IS CO-RESIDENT ON A SECTOR WITH BNIMAG.  ANY        1-4*
                               7686 *      MODIFICATION SHOULD CONSIDER THE CO-RESIDENCY AND THE      1-4*
                               7687 *      LIMITATION OF THE SECTOR BOUNDARY ON SIZE.                 1-4*
                               7688 *                                                                     *
                               7689 *   REQUIRED MODULES                                                  *
                               7690 *        @SYSEQ - COMMON SYSTEM EQUATES.                              *
                               7691 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.   *
                               7692 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.      *
                               7693 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               7694 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               7695 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               7696 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               7697 *        $B$EQU - COMPILER FIXED EQUATES.                             *
                               7698 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               7699 *                                                                     *
                               7700 *   OTHER                                                             *
                               7701 *      BMREAD IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               7702 ***********************************************************************

                               7704 *
                               7705 * ENTER BMREAD - MAT READ STATEMENT ROUTINE
                               7706 *
                          17D0 7707 BMREAD EQU   *                         BMREAD ENTRY POINT
                               7708 *
                               7709 * SET GET RTN TO SKIP TO 'D' IN KEYWORD 'MAT READ'
                               7710 *
    17D0 3C 06 0873            7711 BMR010 MVI   B$NUMC,B@LMRD-1           SET GETC TO SKIP TO 'D'
    17D4 C0 87 0867            7712        B     B$GETC                    LINK IT ADVANCE POINTER
                               7713 *
                               7714 * CALL MATRIX REFERENCE ROUTINE TO GENERATE DOPE VECTOR STACKING INSTR
                               7715 *
    17D8 C0 87 18F3            7716 BMR020 B     B$MATR                    LINK TO PROCESS MAT-REFERENCE
                               7717 *
                               7718 * GENERATE A MATRIX FUNCTION CALL INSTR WHICH REFERENCES THE VADDR OF
                               7719 * THE RUN-TIME MATRIX DATA READ ROUTINE
                               7720 *
    17DC D2 02 F9              7721 BMR030 LA    BMRMFC(,@BR),@XR          LOAD CADDR OF 'MF1' INSTR
    17DF 34 02 0A40            7722        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'MF1'
    17E3 3C 02 0A41            7723        MVI   B$PNBY,B@LMF1-1           SET LNG PARM OF PUT FOR 'MF1'
    17E7 C0 87 093A            7724        B     B$PUTC                    LINK TO GENERATE 'MF1' INSTR
                               7725 *
                               7726 * TEST DELIMITER FOR BEING A STATEMENT TERMINATOR
                               7727 *
    17EB 35 02 0878            7728 BMR040 L     B$GPTR,@XR                RESTORE TEXT POINTER
    17EF BD 1E 00              7729        CLI   B@CHAR(,@XR),B@EOST       IF DELIMITER IS NOT EOS
    17F2 D0 01 D8              7730        BNE   BMR020(,@BR)              * GO PROCESS NEXT MAT-REFERENCE
                               7731 *
                               7732 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               7733 *
    17F5 C0 87 0700            7734 BMR050 B     B$DIST                    RETURN TO DISTRIBUTOR
                               7735 ***********************************************************************
                               7736 * MAT READ STATEMENT ROUTINE STORAGE AND PARAMETER AREA
                               7737 ***********************************************************************
                               7738 *
    17F9 18               17F9 7739 BMRMFC DC    AL(B@LCOP)(B@CMF1)        'MF1' INSTR OPCODE
    17FA 3E00             17FB 7740 BMRMFO DC    AL(B@LCVA)(V$XMRD)        'MF1' INSTR OPERAND
                               7741 *
                               7742 ***********************************************************************
                               7743 *
                               7744 * END OF 'MAT READ' STATEMENT ROUTINE CODING
                               7745 *
                               7747 ***********************************************************************
                               7748 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               7749 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               7750 *                                                                     *
                               7751 ***********************************************************************
                               7752 *STATUS                                                               *
                               7753 *   VERSION 1 MODIFICATION 0                                          *
                               7754 *                                                                     *
                               7755 *FUNCTION                                                             *
                               7756 *   BXPUTX IS EXECUTED TO TRANSLATE PUT STATEMENTS AS THEY OCCUR IN A *
                               7757 *   BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE    *
                               7758 *   PSEUDOCODE IN VIRTUAL MEMORY.                                     *
                               7759 *                                                                     *
                               7760 *ENTRY POINTS                                                         *
                               7761 *   BXPUTX HAS ONLY ONE ENTRY POINT:                                  *
                               7762 *        BXPUTX - TRANSLATE PUT STATEMENT                             *
                               7763 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               7764 *        B     BXPUTX                                                 *
                               7765 *                                                                     *
                               7766 *LINK                                                                 *
                               7767 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               7768 *     THAT RECORD SEGMENT CONTAINS THE FIRST CHARACTER IN THE         *
                               7769 *     LEADING KEYWORD, PUT.                                           *
                               7770 *     TEST CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                               7771 *     CHARACTER IN THE LEADING KEYWORD, PUT.                          *
                               7772 *                                                                     *
                               7773 *OUTPUT                                                               *
                               7774 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               7775 *     GENERATED BY BXPUTX IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               7776 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               7777 *     SEQUENCES.                                                      *
                               7778 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               7779 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               7780 *                                                                     *
                               7781 *EXTERNAL REFERENCES                                                  *
                               7782 *        B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.*
                               7783 *        B$PUTC - (B$PCAD, ISPABY, B$ARSA, - ENTRY TO COMPILER VIRT   *
                               7784 *                 MEMORY ROUTINE.                                     *
                               7785 *        B$CSCN - (B$CSSW) - ENTRY TO BASIC COMPILER CHARACTER SCAN   *
                               7786 *                 ROUTINE.                                            *
                               7787 *        B$SCAN - ENTRY TO BASIC COMPILER ARITMETIC EXPRESSION SCAN   *
                               7788 *                 ROUTINE.                                            *
                               7789 *        B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                *
                               7790 *                                                                     *
                               7791 *EXITS, NORMAL                                                        *
                               7792 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               7793 *                                                                     *
                               7794 *EXITS, ERROR                                                         *
                               7795 *   N/A                                                               *
                               7796 *                                                                     *
                               7797 *TABLES/WORK AREAS                                                    *
                               7798 *   N/A                                                               *
                               7799 *                                                                     *
                               7800 *ATTRIBUTES                                                           *
                               7801 *   BXPUTX IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               7802 *                                                                     *
                               7803 *CHARACTER CODE DEPENDENCY                                            *
                               7804 *                                                                     *
                               7805 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               7806 *                                                                     *
                               7807 *NTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.                *
                               7808 *                                                                     *
                               7809 *NOTES                                                                *
                               7810 *   ERROR PROCEDURES                                                  *
                               7811 *      N/A                                                            *
                               7812 *                                                                     *
                               7813 *   REGISTER USAGE                                                    *
                               7814 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               7815 *                                                                     *
                               7816 *   SAVED/RESTORED AREAS                                              *
                               7817 *      N/A                                                            *
                               7818 *                                                                     *
                               7819 *   MODIFICATION CONSIDERATIONS                                       *
                               7820 *   BXPUTX RESIDES ON THE SAME SECTOR WITH BPCLET AND BXGETX.  1-4    *
                               7821 *   ANY MODIFICATION TO BXPUTX WILL CHANGE THE ENTRY ADDRESSES 1-40   *
                               7822 *   OF BPCLET AND BXGETX AND MUST CONSIDER THE LIMITATION 1-4.        *
                               7823 *   OF THE SECTOR BOUNDARY ON SIZE. 1-40                              *
                               7824 *                                                                     *
                               7825 *   REQUIRED MODULES                                                  *
                               7826 *        @STSEQ - COMMON SYSTEM EQUATES.                              *
                               7827 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.   *
                               7828 *        @CANEQ - COMION CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.      *
                               7829 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               7830 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               7831 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               7832 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               7833 *        $B$EQU - COMPILER FIXED EQUATES.                             *
                               7834 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               7835 *                                                                     *
                               7836 *   OTHER                                                             *
                               7837 *      BXPUTX IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               7838 ***********************************************************************

    1800                       7840        ORG   *,256,0                   BEGIN AT CORE PAGE BOUNDARY 1-4
                          1800 7841        USING *,@BR                     DEFINE BASE AMA FOR CORE PG 1-4
                               7842 *
                               7843 * ENTER BXPUTX - 'PUT' STATEMENT ROUTINE
                               7844 *
                          1800 7845 BXPUTX EQU *                           BXPUTX ENTRY POINT
                               7846 *
                               7847 * SET POINTER TO SKIP TO CHARACTER FOLLOWING 'PUT'
                               7848 *
    1800 3C 02 0873            7849 BXP010 MVI   B$NUMC,B@LKPT-1           SET GET RTN TO SKIP KEYWORD
    1804 C0 87 0867            7850        B     B$GETC                    LINK TO ADVANCE POINTER
    1808 C0 87 14B0            7851        B     B$CSCN                    LINK TO PROCESS FILE REFERENCE
                               7852 *
                               7853 *   GENERATE THE 'ADF' PMC IN VIRTUAL MEMORY (IF THE FILENAME IN THE
                               7854 *   STMT DID NOT MATCH ONE OF THE TABLE ENTRIES, THE 'ADF' OPERAND WILL
                               7855 *   BE ZERO)
                               7856 *
    180C D2 02 63              7857 BXP100 LA    BXPAFC(,@BR),@XR          LOAD CADDR OF 'ADF' INSTR
    180F 34 02 0A40            7858        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'ADF'
    1813 3C 01 0A41            7859        MVI   B$PNBY,B@LADF-1           SET LNG PARM OF PUT FOR 'ADF'
    1817 C0 87 093A            7860        B     B$PUTC                    LINK TO GENERATE 'ADF' PMC
                               7861 *
                               7862 * CALL GET ROUTINE TO GET NEXT CHARACTER
                               7863 *
    181B 3C 00 0873            7864 BXP120 MVI   B$NUMC,B@GETS             DISABLE GET ROUTINE
    181F C0 87 0867            7865        B     B$GETC                    LINK TO GET CHARACTER POINTER
                               7866 *
                               7867 * ATTEMPT TO PROCESS THE VARIABLE AS ARITHMETIC VARIABLE
                               7868 *
    1823 C0 87 1514            7869 BXP140 B     B$SCAN                    LINK TO ATTEMPT 4RITH PROCESS
                               7870 *
                               7871 * TEST FOR ANY PMC GENERATION
                               7872 *
    1827 38 01 0A45            7873 BXP150 TBN   B$ARSW,B$ARMK             IF NO PMC GENERATED
    182B F2 90 06              7874        JF    BXP170                    * GO TEST FOR CHAR VARIABLE
                               7875 *
                               7876 * SET 'PUT' OPERAND FOR ARITH VARIABLE AND BRANCH TO GENERATE 'PUT' PMC
                               7877 *
    182E 7C 02 66              7878 BXP160 MVI   BXPPTO(,@BR),BXPC02 SET CODE FOR ARITH VARIABLE
    1831 D0 87 46              7879        B     BXP210(,@BR)              GO GENERATE 'PUT' PMC
                               7880 *
                               7881 * TEST FOR THIS BEING A CHARACTER VARIABLE
                               7882 *
    1834 38 07 14BC            7883 BXP170 TBN   B$CSSW,B$CSMK             IF VAR IS CHAR VARIABLE
    1838 F2 10 04              7884        JT    BXP190                    * JUMP TO PROCESS CHAR VAR
                               7885 *
                               7886 * IF LIST ELEMENT IS A CHAR CONSTANT DISABLE GET ROUTINE SKIP PARAMETER
                               7887 *
    183B 3C 00 0873            7888 BXP180 MVI   B$NUMC,B@GETS             DISABLE GET RTN SKIPPARM
                               7889 *
                               7890 * BRANCH TO CHARACTER SCAN ROUTINE TO PROCESS CHARACTER ELEMENT
                               7891 *
    183F C0 87 14B0            7892 BXP190 B     B$CSCN                    LINK TO PROCESS CHAR ELEMENT
                               7893 *
                               7894 * SET 'PUT' OPERAND FOR A CHARACTER ELEMENT
                               7895 *
    1843 7C 04 66              7896 BXP200 MVI   BXPPTO(,@BR),BXPC04       SET CODE FOR CHAR ELEMENT
                               7897 *
                               7898 * GENERATE THE 'PUT' PMC IN VIRTUAL MEMORY
                               7899 *
    1846 D2 02 65              7900 BXP210 LA    BXPPTC(,@BR),@XR          LOAD CADOR OF 'PUT' INSTR
    1849 34 02 0A40            7901        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'PUT'
    184D 3C 01 0A41            7902        MVI   B$PNBY,B@LPUT-1           SET LNG PARM CF PUT FOR 'PUT'
    1851 C0 87 093A            7903        B     B$PUTC                    LINK TO GENERATE 'PUT' PMC
                               7904 *
                               7905 * TEST NEXT TEXT CHAR FOR BEING THE END-OF-STATEMENT
                               7906 *
    1855 35 02 0878            7907 BXP220 L     B$GPTR,@XR                RESTORE TEXT POINTER
    1859 BD 1E 00              7908        CLI   B@CHAR(,@XR),B@EOST       IF OTHER ELEMENTS EXIST
    185C D0 01 23              7909        BNE   BXP140(,@BR)              GO PROCESS NEXT LIST ELEMENT
                               7910 *
                               7911 * TEST NEXT TEXT CHAR BEING THE EOND-OF-STATEMENT
                               7912 *
    185F C0 87 0700            7913 BXP230 B     B$DIST                    RETURN TO DISTRIBUTOR
```

```
                               7915 ***********************************************************************
                               7916 * 'PUT' STATEMENT STORAGE AND PARAMETER AREAS
                               7917 ***********************************************************************
                               7918 *
    1863 58               1863 7919 BXPAFC DC    AL(B@LCOP)(B@CADF)        'ADF' INSTR OPCODE
    1864 01               1864 7920 BXPAFO DC    XL1'01'                   PUT INDICATOR FOR 'ADF' INSTR
                               7921 *
    1865 54               1865 7922 BXPPTC DC    AL(B@LCOP)(B@CPUT)        'PUT' INSTR OPCODE
    1866                  1866 7923 BXPPTO DS    CL(B@LCXX)                'PUT' INSTR OPERAND

                               7925 ***********************************************************************
                               7926 * 'PUT' STATEMENT CONSTANTS AND EQUATES
                               7927 ***********************************************************************
                               7928 *
                               7929 * CONSTANTS
                               7930 *
                          1867 7931 BXPSFA EQU   *
    1867 0001             1868 7932 BXPBN1 DC    IL(@CADDR)'1'             BINARY 1
                               7933 *
                               7934 * EQUATES
                               7935 *
                          0002 7936 BXPC02 EQU   X'02'                     ARITH VARIABLE CODE
                          0004 7937 BXPC04 EQU   X'04'                     CHARACTER VAR OR CONSTANT CODE
                               7938 *
                               7939 ***********************************************************************
                               7940 *
                               7941 * END OF 'PUT' STATEMENT ROUTINE CODING
                               7942 *
                               7944 ***********************************************************************
                               7945 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               7946 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               7947 *                                                                     *
                               7948 ***********************************************************************
                               7949 *STATUS                                                               *
                               7950 *   VERSION 1 MODIFICATION 0                                          *
                               7951 *                                                                     *
                               7952 *FUNCTION                                                             *
                               7953 *   BPCLET IS EXECUTED TO TRANSLATE CHARACTER ASSIGNMENT AND LET      *
                               7954 *   STATEMENTS AS THEY OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE  *
                               7955 *   PSEUDOCODE AND TO PLACE THE PSEUDOCODE IN VIRTUAL MEMORY.         *
                               7956 *                                                                     *
                               7957 *ENTRY POINTS                                                         *
                               7958 *   BPCLET HAS TWO ENTRY POINTS:                                      *
                               7959 *        BPCASN - TRANSLATE CHARACTER ASSIGNMENT STATEMENT            *
                               7960 *        BPCLET - TRANSLATE CHARACTER LET STATEMENT                   *
                               7961 *   THE FORMAT OF THE CALLING SEQUENCES IS:                           *
                               7962 *        B     BPCASN                                                 *
                               7963 *        B     BPCLET                                                 *
                               7964 *                                                                     *
                               7965 *INPUT                                                                *
                               7966 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               7967 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               7968 *     LEADING KEYWORD, LET, OR THE FIRST CHARACTER IN THE ASSIGNMENT  *
                               7969 *     LIST IF THE OPTIONAL KEYWORD IS OMITTED.                        *
                               7970 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE 1ST   *
                               7971 *     CHARACTER IN THE LEADING KEYWORD, LET, OR IN THE ASSIGNMENT     *
                               7972 *     LIST IF THE KEYWORD IS OMITTED.                                 *
                               7973 *                                                                     *
                               7974 *OUTPUT                                                               *
                               7975 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               7976 *     GENERATED BY BPCLET IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               7977 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               7978 *     SEQUENCES.                                                      *
                               7979 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               7980 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               7981 *                                                                     *
                               7982 *EXTERNAL RE,RENCES                                                   *
                               7983 *        B$GETC - (B$NUMC) - ENTRY TO BASIC RETRIEVAL ROUTINE.        *
                               7984 *        B$PUTC - (B$PCAD, B$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY *
                               7985 *                 OUTPUT ROUTINE.                                     *
                               7986 *        B$LIST - ENTRY TO BASIC COMPILER LIST ADDRESS ROUTINE.       *
                               7987 *        B$CSCN - ENTRY TO BASIC COMPILER CHARACTER SCAN ROUTINE.     *
                               7988 *        B$LIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                *
                               7989 *                                                                     *
                               7990 *EXITS, NORMAL                                                        *
                               7991 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR                      *
                               7992 *                                                                     *
                               7993 *EXITS, ERROR                                                         *
                               7994 *   N/A                                                               *
                               7995 *                                                                     *
                               7996 *TABLES/WORK AREAS                                                    *
                               7997 *   N/A                                                               *
                               7998 *                                                                     *
                               7999 *ATTRIBUTES                                                           *
                               8000 *   BPCLET IS NATURALLY RELOCATABLE AND REUSABLE.                    *
                               8001 *                                                                     *
                               8002 *CHARACTER CODE DEPENDENCY                                            *
                               8003 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON ANY PARTICULAR  *
                               8004 *   INTERNAL REPRESENTATION OF THE INTERNAL CHARACTER SET.            *
                               8005 *                                                                     *
                               8006 *NOTES                                                                *
                               8007 *   ERROR PROCEDURES                                                  *
                               8008 *      N/A                                                            *
                               8009 *                                                                     *
                               8010 *   REGISTER USAGE                                                    *
                               8011 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               8012 *                                                                     *
                               8013 *   SAVED/RESTORED AREAS                                              *
                               8014 *      N/A                                                            *
                               8015 *                                                                     *
                               8016 *   MODIFICATION CONSIDERATIONS                                       *
                               8017 *      BPCLET RESIDES ON THE SAME SECTOR WITH BXPUTX AND BXGETX.   1-4*
                               8018 *      ANY MODIFICATION TO BPCLET WILL CHANGE THE ENTRY ADDRESS    1-4*
                               8019 *      OF BXGETX AND MUST CONSIDER THE LIMITATION OF THE SECTOR    1-4*
                               8020 *      BOUNDARY ON SIZE.                                           1-4*
                               8021 *                                                                     *
                               8022 *   REQUIRED MODULES                                                  *
                               8023 *        @SYSEQ - COMMON SYSTEM EQUATES.                              *
                               8024 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.   *
                               8025 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.      *
                               8026 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               8027 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               8028 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               8029 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES                       *
                               8030 *        $B$EQU - COMPILER FIXED EQUATES.                             *
                               8031 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               8032 *                                                                     *
                               8033 *   OTHER                                                             *
                               8034 *      BPCLET IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS       *
                               8035 ***********************************************************************

                               8037 *
                               8038 * ENTER BPCLET - CHARACTER 'LET' STATEMENT PROCESSOR
                               8039 *
                          1869 8040 BPCLET EQU   *                         BPCLET ENTRY POINT
                               8041 *
                               8042 * SKIP PAST 'LET' TO 1ST ASSIGNMENT LIST SYMBOL CHARACTER
                               8043 *
    1869 3C 03 0873            8044 BPC010 MVI   B$NUMC,B@LLET             SET GET ROUTINE TO SLIP 'LET'
    186D C0 87 0867            8045        B     B$GETC                    LINK TO GET 1ST SYMBOL CHAR
                               8046 *
                               8047 * ENTER BPCASN - CHARACTER ASSIGNMENT STATEMENT PROCESSOR
                               8048 *
                          1871 8049 BPCASN EQU   *                         BPCASN ENTRY POINT
                               8050 *
                               8051 * ESTABLISH A COUNT OPERAND FIELD WHICH INDICATES TINE NUMBER OF
                               8052 * VARIABLES IN THE ASSIGNMENT LIST AND INITIALIZE THE COUNT TO ZERO
                               8053 *
    1871 7C 00 A2              8054 BPC020 MVI   BPCUCO(,@BR),@ZERO        SET SYMNC. COUNT TO ZERO
                               8055 *
                               8056 * EVALUATE EACH OF THE CHARACTER SYMBOLS IN THE ASSIGN
                               8057 *
    1874 C0 87 1853            8058 BPC030 B     B$LIST                    LINK TO PROCESS CHAR SYMBOL
    1878 5E 00 A2 A0           8059        ALC   BPCUCO(,@BR),BPCBN1(B@LCNN,@BR)  ADD 1 TO LIST COUNT
                               8060 *
                               8061 * IF DELIMITER IS NOT AN EQUAL SIGN (IE) A COMMA) CONTINUE TO PROCESS
                               8062 * THE ASSIGNMENT LIST
                               8063 *
    187C BD 7E 00              8064 BPC040 CLI   B@CHAR(,@XR),B@EQUL       IF DELIMITER IS AN EQUAL SIGN
    187F F2 81 07              8065        JE    BPC050                     * DETERMINE THE ASSGNMNT VALUE
    1882 C0 87 0867            8066        B     B$GETC                    LINK TO GET NEXT SYMBOL CHAR
    1886 D0 87 74              8067        B     BPC030(,@BR)              GO PROCESS NEXT SYMBOL CHAR
                               8068 *
                               8069 * EVALUATE VALUE TO BE ASSIGNED THE CHARACTER SYMBOLS IN THE LIST AND
                               8070 * SET UP PMC FOR 'USC' BEFORE BRANCHING TO THE KIT ROUTINE
                               8071 *
    1889 C0 87 14B0            8072 BPC050 B     B$CSCN                    LINK TO CHAR SCAN ROUTINE
    188D D2 02 A1              8073        LA    BPCUCC(,@BR),@XR          LOAD CADDR OF 'USC' INSTR
    1890 34 02 0A40            8074        ST    B$PCAD,@XR                SET VADDR PARM FOR PUT RTN
    1894 3C 01 0A41            8075        MVI   B$PNBY,B@LUSC-1           SET LENGTH PARM FOR PUT RTN
    1898 C0 87 093A            8076        B     B$PUTC                    LINK TO OUTPUT 'USC' INSTR
    189C C0 87 0700            8077        B     B$DIST                    RETURN TO DISTRIBUTOR

                               8079 ***********************************************************************
                               8080 * CHARACTER 'LET' ROUTINE CONSTANTS
                               8081 ***********************************************************************
                               8082 *
    18A0 01               18A0 8083 BPCBN1 DC    IL(B@LCNN)'1'             BINARY INTEGER +1

                               8085 ***********************************************************************
                               8086 * CHARACTER 'LET' ROUTINE PMC AND STORAGE PARAMETERS
                               8087 ***********************************************************************
                               8088 *
    18A1 2C               18A1 8089 BPCUCC DC    AL(B@LCOP)(B@CUSC)        UNSTACK CHAR OPCODE
    18A2                  18A2 8090 BPCUCO DS    CL(B@LCNN)                UNSTACK CHAR OPERAND
                               8091 *
                               8092 ***********************************************************************
                               8093 *
                               8094 * END OF CHARACTER 'LET' ROUTINE CODING
                               8095 *
```

```
8097 ***********************************************************************
8098 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
8099 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
8100 *                                                                     *
8101 ***********************************************************************
8102 *STATUS                                                               *
8103 *   VERSION 1 MODIFICATION 0                                          *
8104 *                                                                     *
8105 *FUNCTION                                                             *
8106 *   BXGETX IS EXECUTED TO TRANSLATE GET STATEMENTS AS THEY OCCUR IN   *
8107 *   A BASIC PROGRAM INTO THE APPROPRIATE PSEODOCODE AND TO PLACE THE  *
8108 *   PSEODOCODE IN VIRTUAL MEMORY.                                     *
8109 *                                                                     *
8110 *ENTRY POINTS                                                         *
8111 *   BXGETX HAS ONLY ONE ENTRY POINT:                                  *
8112 *        BXGETX - TRANSLATE GET STATEMENT                             *
8113 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
8114 *        B     BXGETX                                                 *
8115 *                                                                     *
8116 *INPUT                                                                *
8117 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
8118 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
8119 *     LEADING KEYWORD, GET.                                           *
8120 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE 1ST   *
8121 *     CHARACTER IN LEADING KEYWORD, GET.                              *
8122 *                                                                     *
8123 *OUTPUT                                                               *
8124 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
8125 *     GENERATED BY BXGETX IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
8126 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
8127 *     SEQUENCES.                                                      *
8128 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
8129 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
8130 *                                                                     *
8131 *EXTERNAL REFERENCES                                                  *
8132 *        B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.*
8133 *        B$PUTC - (B$PCAD, B$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY *
8134 *                 OUTPUT ROUTINE.                                     *
8135 *        B$LIST - ENTRY TO BASIC COMPILER LIST ADDRESS ROUTINE.       *
8136 *        B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                *
8137 *                                                                     *
8138 *EXITS, NORMAL                                                        *
8139 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
8140 *                                                                     *
8141 *EXITS, ERROR                                                         *
8142 *   N/A                                                               *
8143 *                                                                     *
8144 *TABLES/WORE AREAS                                                    *
8145 *   N/A                                                               *
8146 *                                                                     *
8147 *ATTRIBUTES                                                           *
8148 *   BXGETX IS NATURALLY RELOCATABLE AND REUSABLE.                     *
8149 *                                                                     *
8150 *CHARACTER CODE DEPENDENCY                                            *
8151 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
8152 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
```

```
                               8153 *                                                                     *
                               8154 *NOTES                                                                *
                               8155 *   ERROR PROCEDURES                                                  *
                               8156 *      N/A                                                            *
                               8157 *                                                                     *
                               8158 *   REGISTER USAGE                                                    *
                               8159 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               8160 *                                                                     *
                               8161 *   SAVED/RESTORED AREAS                                              *
                               8162 *      N/A                                                            *
                               8163 *                                                                     *
                               8164 *   MODIFICATION CONSIDERATIONS                                       *
                               8165 *      BXGETX RESIDES ON THE SAME SECTOR WITH BXPUTX AND BPCLET.   1-4*
                               8166 *      ANY MODIFICATION TO BXGETX MUST CONSIDER THIS CO-RESIDENCY  1-4*
                               8167 *      AND THE LIMITATION OF THE SECTOR BOUNDARY ON SIZE.          1-4*
                               8168 *                                                                     *
                               8169 *   REQUIRED MODULES                                                  *
                               8170 *        @SYSEQ - COMMON SYSTEM EQUATES                               *
                               8171 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES    *
                               8172 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS               *
                               8173 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES                    *
                               8174 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES                         *
                               8175 *        @ERMEQ - ERROR MESSAGE EQUATES                               *
                               8176 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES                       *
                               8177 *        SB$EQU - COMPILER FIXED EQUATES                              *
                               8178 *        SB@EQU - COMPILER SYSTEM EQUATES                             *
                               8179 *                                                                     *
                               8180 *   OTHER                                                             *
                               8181 *      BXGETX IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS       *
                               8182 ***********************************************************************

                               8184 *
                               8185 * ENTER BXGETX - 'GET' STATEMENT ROUTINE
                               8186 *
                          18A3 8187 BXGETX EQU   *                         BXGETX ENTRY POINT
                               8188 *
                               8189 * SET POINTER TO SKIP TO CHARACTER FOLLOWING KEYWORD 'GET'
                               8190 *
    18A3 3C 02 0873            8191 BXG010 MVI   B$NUMC,B@LKGT-1           SET GET RTN TO SKIP KEYWORD
    18A7 C0 87 0867            8192        B     B$GETC                    LINK TO ADVANCE POINTER
    18AB C0 87 14B0            8193        B     B$CSCN                    LINK TO PROCESS FILE REFERENCE
                               8194 *
                               8195 * GENERATE THE 'ADF' PMC IN VIRTUAL MEMORY (IF FILE NAME IN THE STMT
                               8196 * DID NOT MATCH ONE OF THE TABLE ENTRIES, THE 'ADF' OPERAND WILL BE
                               8197 * ZERO.
                               8198 *
    18AF D2 02 EB              8199 BXG100 LA    BXGAFC(,@BR),@XR          LOAD CADDR OF 'ADF' INSTR
    18B2 34 02 0A40            8200        ST    B$PCAD,@XR                SET PUT RTN VADDR PARM FOR 'ADF'
    18B6 3C 01 0A41            8201        MVI   B$PNBY,B@LADF-1           SET LNG PARM OF PUT FOR 'ADF'
    18BA C0 87 093A            8202        B     B$PUTC                    LINK TO GENERATE 'ADF' PMC
                               8203 *
                               8204 * CALL GET RTN TO GET NEXT CHARACTER
                               8205 *
    18BE C0 87 0867            8206 BXG110 B     B$GETC                    LINK TO GET NEXT CHARACTER
                               8207 *
                               8208 * GET NEXT CHARACTER
                               8209 *
    18C2 3C 00 0873            8210        MVI   B$NUMC,B@GETS             DISABLE GET ROUTINE
    18C6 C0 87 0867            8211 BXG120 B     B$GETC                    LINK TO GET CHARACTER POINTER
                               8212 *
                               8213 * CALL LIST ROUTINE TO PROCESS CHARACTER
                               8214 *
    18CA C0 87 1853            8215 BXG130 B     B$LIST                    LINK TO PROCESS CHARACTER
                               8216 *
                               8217 * GENERATE 'GET' PMC IN VIRTUAL MEMORY
                               8218 *
    18CE D2 02 ED              8219 BXG140 LA    BXGGTC(,@BR),@XR          LOAD CADDR OF 'GET' PMC
    18D1 34 02 0A40            8220        ST    B$PCAD,@XR                SET PUT RTN VADDR PARM FOR GET
    18D5 3C 02 0A41            8221        MVI   B$PNBY,B@LGET-1           SET PUT RTN LNG PARM FOR GET
    18D9 C0 87 093A            8222        B     B$PUTC                    LINK TO GENERATE PMC
                               8223 *
                               8224 * TEST FOR END OF STATEMENT
                               8225 *
    18DD 35 02 0878            8226 BXG150 L     B$GPTR,@XR                RESTORE TEXT POINTER
    18E1 BD 1E 00              8227        CLI   B@CHAR(,@XR),B@EOST       IF THIS IS NOT TERMINATOR
    18E4 D0 01 C6              8228        BNE   BXG120(,@BR)              * BRANCH TO GET NEXT CHAR
                               8229 *
                               8230 * RETURN CONTROL TO THE COMPLIER DISTRIBUTOR
                               8231 *
    18E7 C0 87 0700            8232 BXGI60 B     B$DIST                    RETURN TO DISTRIBUTOR

                               8234 ***********************************************************************
                               8235 * 'GET' STATEMENT ROUTINE STORAGE AND PARAMETER AREAS
                               8236 ***********************************************************************
                               8237 *
    18EB 58               18EB 8238 BXGAFC DC    AL(B@LCOP)(B@CADF)        'ADF' INSTR OPCODE
    18EC 00               18EC 8239 BXGAFO DC    XL1'00'                   GET INDICATOR FOR 'ADF' INSTR
                               8240 *
    18ED 52               18ED 8241 BXGGTC DC    AL(B@LCOP)(B@CGET)        'GET' INSTR OPCODE
    18EE 2100             18EF 8242 BXGGTO DC    AL(B@LCVA)(V$XSGT)        'GET' INSTR OPERAND

                               8244 ***********************************************************************
                               8245 * 'GET' STATEMENT ROUTINE CONSTANTS AND EQUATES
                               8246 ***********************************************************************
                               8247 *
                               8248 * CONSTANTS
                               8249 *
                          18F0 8250 BXGSFA EQU   *
    18F0 0001             18F1 8251 BXGBN1 DC    IL(@CADDR)'1'             BINARY 1
                               8252 *
                               8253 **********************************************************************,
                               8254 *
                               8255 * END OF 'GET' STATEMENT ROUTINE CODING
                               8256 *
                               8258 ***********************************************************************
                               8259 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               8260 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               8261 *                                                                     *
                               8262 ***********************************************************************
                               8263 *STATUS                                                               *
                               8264 *   VERSION 1 MODIFICATION 0                                          *
                               8265 *                                                                     *
                               8266 *FUNCTION                                                             *
                               8267 *   BKNEXT IS EXECUTED TO TRANSLATE NEXT STATEMENTS AS THEY OCCUR IN  *
                               8268 *   A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE  *
                               8269 *   PSEUDOCODE IN VIRTUAL MEMORY.                                     *
                               8270 *                                                                     *
                               8271 *ENTRY POINTS                                                         *
                               8272 *   BKNEXT HAS ONLY ONE ENTRY POINT:                                  *
                               8273 *        BKNEXT - TRANSLATE NEXT STATEMENT                            *
                               8274 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               8275 *        B     BKNEXT                                                 *
                               8276 *                                                                     *
                               8277 *INPUT                                                                *
                               8278 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               8279 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               8280 *     LEADING KEYWORD, NEXT.                                          *
                               8281 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                               8282 *     CHARACTER IN THE LEADING KEYWORD, NEXT.                         *
                               8283 *   * FOR TABLE - CONTAINS 4-BYTE ENTRIES.  EACH CONTAINING THE       *
                               8284 *     VIRTUAL ADDRESSES OF A FOR-LOOP CONTROL VARIABLE AND OF THE     *
                               8285 *     NXT INSTRUCTION IN THE ASSOCIATED FOR OBJECT CODE SEQUENCE.     *
                               8286 *   * B$FTPT - CONTAINS THE CORE ADDRESS OF THE 1ST BYTE OF THE ENTRY *
                               8287 *     LAST PLACED IN THE FOR TABLE, OR OF THE BOTTOM GUARD ENTRY      *
                               8288 *     WHEN THE TABLE IS EMPTY.                                        *
                               8289 *                                                                     *
                               8290 *OUTPUT                                                               *
                               8291 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               8292 *     GENERATED BY BKNEXT IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               8293 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               8294 *     SEQUENCES.                                                      *
                               8295 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               8296 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               8297 *   * FOR TABLE - WHEN THE CURRENT TABLE ENTRY CONTROL VARIABLE       *
                               8298 *     MATCHES THAT SPECIFIED IN THE NEXT STATEMENT, THAT ENTRY IS     *
                               8299 *     DELETED FROM THE TABLE. THE TABLE IS NOT AFFECTED WHEN A        *
                               8300 *     COMPILER ERROR OCCURS.                                          *
                               8301 *   * B$FTPT - CONTAINS THE CORE ADDRESS OF THE 1ST BYTE OF THE FOR   *
                               8302 *     TABLE ENTRY PRECEDING THAT DELETED FROM THE TABLE.  B$FTPT IS   *
                               8303 *     NOT MODIFIED WHEN A COMPILER ERROR OCCURS.                      *
                               8304 *   * B$BRVA - CONTAINS THE VIRTUAL ADDRESS OF THE RIGHT BYTE         *
                               8305 *     OF THE ADDRESS OPERAND FIELD IN THE NXT INSTRUCTION REFERENCED  *
                               8306 *     BY THE CURRENT (BEFORE DELETION) FOR TABLE ENTRY.               *
                               8307 *   * B$NXSU - SET TO ON STATUS TO CAUSE RESOLUTION OF THE NXT        *
                               8308 *     INSTRUCTION OPERAND BY THE COMPILER DISTRIBUTOR.                *
                               8309 *                                                                     *
                               8310 *EXTERNAL REFERENCES                                                  *
                               8311 *        B$GETC - (B$NUMC) - ENTRY TO BASIC RETRIEVAL ROUTINE.        *
                               8312 *        B$PUTC - (B$PFNC, B$PCAD, B$PNBY, B$PERC) - ENTRY TO COMPILER*
                               8313 *                 VIRTUAL MEMORY OUTPUT ROUTINE.                      *
                               8314 *        B$SYMB - (B$BCKT) - ENTRY TO BASIC SYMBOL TRANSLATION RTN.   *
                               8315 *        B$BTAB - (B$BRVA) - ENTRY TO BASIC COMPILER BRANCH TABLE RTN.*
                               8316 *        B$FTPT - ENTRY TO FOR TABLE.                                 *
                               8317 *        B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                *
                               8318 *                                                                     *
                               8319 *EXITS, NORMAL                                                        *
                               8320 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               8321 *                                                                     *
                               8322 *EXITS, ERROR                                                         *
                               8323 *   N/A                                                               *
                               8324 *                                                                     *
                               8325 *TABLES/WORK AREAS                                                    *
                               8326 *   FOR TABLE - EXTERNAL TO BKNEXT, THIS PUSH-DONN TABLE CONTAINS     *
                               8327 *   TEN 4-BYTE ENTRY LOCATIONS.  THE FIRST ENTRY LOCATION IS          *
                               8328 *   ALWAYS CLEARED TO ZEROS, AND IS USED TO GUARD AGAINST A TABLE     *
                               8329 *   REFERENCE WHEN THE TABLE IS EMPTY.  THE FOLLOWING NINE ENTRY      *
                               8330 *   LOCATIONS MAY EACH CONTAIN VIRTUAL ADDRESSES REFERENCING AN       *
                               8331 *   UNFINISHED FOR-LOOP CONTROL VARIABLE AND ITS ASSOCIATED NXT       *
                               8332 *   INSTRUCTION, DEPENDING ON THE CURRENT LOOP NESTING DEPTH IN THE   *
                               8333 *   PROGRAM.                                                          *
                               8334 *                                                                     *
                               8335 *ATTRIBUTES                                                           *
                               8336 *   BKNEXT IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               8337 *                                                                     *
                               8338 *CHARACTER CODE DEPENDENCY                                            *
                               8339 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               8340 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               8341 *                                                                     *
                               8342 *NOTES                                                                *
                               8343 *   ERROR PROCEDURES                                                  *
                               8344 *      WHEN THE CURRENT NEXT CONTROL VARIABLE DOES NOT MATCH THE      *
                               8345 *      LAST FOR TABLE ENTRY THE ERROR CONDITION CODE FOR              *
                               8346 *      FOR/NEXT NESTED INCORRECTLY IS LOGGED IN VIRTUAL MEMORY.       *
                               8347 *      WHEN NO ACTIVE ENTRY EXISTS IN THE FOR TABLE THE ERROR         *
                               8348 *      CONDITION CODE FOR NEXT STATEMENT OUT OF SEQUENCE IS LOGGED    *
                               8349 *      IN VIRTUAL MEMORY.                                             *
                               8350 *                                                                     *
                               8351 *   REGISTER USAGE                                                    *
                               8352 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               8353 *                                                                     *
                               8354 *   SAVED/RESTORED AREAS                                              *
                               8355 *      N/A                                                            *
                               8356 *                                                                     *
                               8357 *   MODIFICATION CONSIDERATIONS                                       *
                               8358 *      BKNEXT RESIDES ON THE SAME SECTOR WITH BMGETX AND BKGOTO.   1-4*
                               8359 *      ANY MODIFICATION TO BKNEXT WILL CHANGE THE ENTRY ADDRESSES  1-4*
                               8360 *      OF  BMGETX AND BKGOTO AND MUST CONSIDER THE LIMITATION      1-4*
                               8361 *      OF THE SECTOR BOUNDARY ON SIZE.                             1-4*
                               8362 *                                                                     *
                               8363 *   REQUIRED MODULES                                                  *
                               8364 *        @SYSEQ - COMMON SYSTEM EQUATES.                              *
                               8365 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.   *
                               8366 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.      *
                               8367 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               8368 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               8369 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               8370 *        $V$EQU - FIXED VIRTUAL ADDRESSES EQUATES.                    *
                               8371 *        $B$EQU - COMPILER FIXED EQUATES.                             *
                               8372 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               8373 *                                                                     *
                               8374 *   OTHER                                                             *
                               8375 *      BKNEXT IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               8376 ***********************************************************************

    1900                       8378        ORG   *,256,0                    BEGIN AT CORE PAGE BOUNDARY
                          1900 8379        USING *,@BR                      DEFINE BASE ADDR FOR CORE PAGE
                               8380 *
                               8381 * ENTER BKNEXT - 'NEXT' STATEMENT ROUTINE
                               8382 *
                          1900 8383 BKNEXT EQU   *                          BKNEXT ENTRY POINT
                               8384 *
                               8385 * SET INPUT PARAMETER TO SKIP KEYWORD 'NEXT'
                               8386 *
    1900 3C 04 0873            8387 BKN010 MVI   B$NUMC,B@LNEX             SET GET RTN TO SKIP 'NEXT'
    1904 C0 87 0867            8388        B     B$GETC                     LINK TO ADVANCE POINTER
                               8389 *
                               8390 * FIND THE VIRTUAL ADDRESS OF THE 'NEXT' CONTROL VARIABLE
                               8391 *
    1908 C0 87 0DBC            8392 BKN020 B     B$SYMB                     LINK TO FIND CTRL VAR VADDR
                               8393 *
                               8394 * COMPARE 'NEXT' CTRL VAR VADDR WITH 'FOR' TABLE CTRL VAR VADDR
                               8395 *
    190C 35 02 1B0D            8396 BKN030 L     B$FTPT,@XR                 LOAD THE 'FOR' TABLE POINTER
    1910 8D 01 01 1590         8397        CLC   BKNFTD(,@XR),B$BCKT(@VADDR)  IF CTRL VARIABLES MATCH
    1915 F2 81 1C              8398        JE    BKN090                     * JUMP TO PROCESS 'BRA' PMC
                               8399 *
                               8400 * SET PUT ROUTINE FOR ERROR OUTPUT
                               8401 *
    1918 3C 33 094E            8402 BKN040 MVI   B$PFNC,B$PFAE             SET PUT RTN FOR ADD ERROR COND
                               8403 *
                               8404 * CHECK 'FOR TABLE' CTRL VAR FOR DUMMY ENTRY
                               8405 *
    191C BD 00 01              8406 BKN050 CLI   BKNFTD(,@XR),BKNDUM       IF 'FOR TABLE' VADDR IS DUMMY
    191F F2 81 07              8407        JE    BKN070                     * JUMP TO SET PROPER ERROR CODE
                               8408 *
                               8409 * GENERATE ERROR CODE FOR UNBALANCED 'FOR'/'NEXT' CONTROL VARIABLES
                               8410 *
    1922 3C AC 0A39            8411 BKN060 MVI   B$PERC,@@E607             GENERATE ERROR CODE
    1926 F2 87 04              8412        J     BKN080                     JUMP TO LINK TO PUT RTN
                               8413 *
                               8414 * GENERATE ERROR CODE FOR 'NEXT' WITH NON-EXISTENT 'FOR'
                               8415 *
    1929 3C AB 0A39            8416 BKN070 MVI   B$PERC,@@E606             GENERATE ERROR CODE
    192D C0 87 093A            8417 BKN080 B     B$PUTC                     LINK TO WRITE ERROR CODE
    1931 F2 87 26              8418        J     BKN120                     JUMP TO BKNEXT EXIT
                               8419 *
                               8420 * ESTABLISH THE VIRTUAL ADDRESS OF THE 'FOR TABLE' NXT INSTRUCTION
                               8421 * AS THE OPERAND OF A 'BRA' INSTRUCTION
                               8422 *
    1934 6C 01 64 03           8423 BKN090 MVC   BKNBRO(,@BR),BKNNXT(@VADDR,@XR)  SET 'BRA' OPERAND
    1938 D2 02 62              8424        LA    BKNBRC(,@BR),@XR           LOAD CADDR OF 'BRA' INSTR
    193B 34 02 0A40            8425        ST    B$PCAD,@XR                 SET PUT RTN FOR VADDR OF 'BRA'
    193F 3C 02 0A41            8426        MVI   B$PNBY,B@LBRA-1           SET PUT RTN FOR LENGTH OF 'BRA'
    1943 C0 87 093A            8427        B     B$PUTC                    LINK TO GENERATE PMC
                               8428 *
                               8429 * DECREMENT FOR TABLE' POINTER TO NEXT OUTER DEPTH LEVEL
                               8430 *
    1947 1F 01 1B0D 5F         8431 BKN100 SLC   B$FTPT,BKNFEL(@CADDR,@BR)  DECREMENT FOR TABLE' POINTER
                               8432 *
                               8433 * SET PARAMETERS FOR DISTRIBUTOR BRANCH TABLE UPDATE
                               8434 *
    194C 3A 07 071D            8435 BKN110 SBN   B$NXSW,B$NXMK              SET NEXT SWITCH ON
    1950 1C 01 19EF 64         8436        MVC   B$BRVA,BKNBRO(@VADDR,@BR)  MOVE VADDR OF NXT INSTR
    1955 1E 01 19EF 61         8437        ALC   B$BRVA,BKNEX2(@VADDR,@BR)  SET PARAMETER FOR 'NXT' OPND
                               8438 *
                               8439 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               8440 *
    195A C0 87 0700            8441 BKN120 B     B$DIST                    RETURN TO DISTRIBUTOR

                               8443 ***********************************************************************
                               8444 * 'NEXT' STATEMENT ROUTINE CONSTANTS AND EQUATES
                               8445 ***********************************************************************
                               8446 *
                               8447 * EQUATES
                               8448 *
                          0001 8449 BKNFTD EQU   1                         DISP FOR 'FOR TABLE' CTRL VAR
                          0000 8450 BKNDUM EQU   0                         DUMMY ENTRY COMPARISON
                          0003 8451 BKNNXT EQU   3                         DISP FOR 'FOR TABLE' NXT VADDR
                               8452 *
                               8453 * CONSTANTS
                               8454 *
    195E 0004             195F 8455 BKNFEL DC    AL(@CADDR)(B@LFRT)        LENGTH OF 'FOR TABLE' ENTRY
    1960 0002             1961 8456 BKNEX2 DC    IL(@CADDR)'2'             BINARY 2

                               8458 ***********************************************************************
                               8459 * 'NEXT' STATEMENT ROUTINE PMC AND STORAGE PARAMETERS
                               8460 ***********************************************************************
                               8461 *
    1962 46               1962 8462 BKNBRC DC    AL(B@LCOP)(B@CBRA)        'BRA' INSTR OPCODE
    1963                  1964 8463 BKNBRO DS    CL(@VADDR)                'BRA' INSTR OPERAND
                               8464 *
                               8465 ***********************************************************************
                               8466 *
                               8467 * END OF 'NEXT' STATEMENT ROUTINE CODING
                               8468 *
```

```
***********************************************************************
*  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
*           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
*                                                                     *
***********************************************************************
*STATUS                                                               *
*   VERSION 1 MODIFICATION 0                                          *
*                                                                     *
*FUNCTION                                                             *
*   BMGETX IS EXECUTED TO TRANSLATE MAT GET STATEMENTS IF THEY OCCUR  *
*   IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE   *
*   THE PSEUDOCODE IN VIRTUAL MEMORY.                                 *
*                                                                     *
*ENTRY POINTS                                                         *
*   BMGETX HAS ONLY ONE ENTRY POINT:                                  *
*        BMGETX - TRANSLATE MAT GET STATEMENT                         *
*   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
*        B     BMGETX                                                 *
*                                                                     *
*INPUT                                                                *
*   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
*     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER OF THE       *
*     LEADING KEYWORD, MAT GET.                                       *
*   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
*     CHARACTER IN THE LEADING KEYWORD, MAT GET.                      *
*                                                                     *
*OUTPUT                                                               *
*   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
*     GENERATED BY BMGETX IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
*     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
*     SEQUENCES.                                                      *
*   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
*     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
*                                                                     *
*EXTERNAL REFERENCES                                                  *
*        B$GETC - (B$NUMC) - ENTRY TO BASIC RETRIEVAL ROUTINE.        *
*        B$PUTC - (B$PCAD)(B$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY *
*                 OUTPUT ROUTINE.                                     *
*        B$MATR - ENTRY TO BASIC COMPILER MATRIX REFERENCE ROUTINE.   *
*        B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                *
*                                                                     *
*EXITS, NORMAL                                                        *
*   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
*                                                                     *
*EXITS, ERROR                                                         *
*   N/A                                                               *
*                                                                     *
*TAILES/WORK AREAS                                                    *
*   N/A                                                               *
*                                                                     *
*ATTRIBUTES                                                           *
*   BNGETX IS RELOCATABLE AND REUSABLE.                               *
*                                                                     *
*CHARACTER CODE DEPENDENCY                                            *
*   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON ANY PARTICULAR  *
*   INTERNAL REPRESENTATION UP THE EXTERNAL CHARACTER SET.            *
```

```
                               8526 *                                                                     *
                               8527 *NOTES                                                                *
                               8528 *   ERROR PROCEDURES                                                  *
                               8529 *      N/A                                                            *
                               8530 *                                                                     *
                               8531 *   REGISTER USAGE                                                    *
                               8532 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               8533 *                                                                     *
                               8534 *   SAVED/RESTORED AREAS                                              *
                               8535 *      N/A                                                            *
                               8536 *                                                                     *
                               8537 *   MODIFICATION CONSIPERATIPAS                                       *
                               8538 *      BMGETX RESIDES ON A SECTOR WITH BKNEXT AND BKGOTO.  ANY     1-4*
                               8539 *      MODIFICATION TO BMGETX WILL CHANGE THE ENTRY ADDRESS OF     1-4*
                               8540 *      BKCOTO AND MUST CONSIDER THE LIMITATION OF THE SECTOR       1-4*
                               8541 *      BOUNDARY ON SIZE.                                           1-4*
                               8542 *                                                                     *
                               8543 *   REQUIRED MODULES                                                  *
                               8544 *        @SYSEQ - COMMON SYSTEM EQUATES.                              *
                               8545 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.   *
                               8546 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS.              *
                               8547 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               8548 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               8549 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               8550 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               8551 *        $B$EQU - COMPILER FIXED EQUATES.                             *
                               8552 *        SB@EQU - COMPILER SYSTEM EQUATES.                            *
                               8553 *                                                                     *
                               8554 *   OTHER                                                             *
                               8555 *      BMGETX IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               8556 ***********************************************************************

                               8558 *
                               8559 * ENTER BMGETX - 'MAT GET' STATEMENT
                               8560 *
                          1965 8561 BMGETX EQU  *                         BMGETX ENTRY POINT
                               8562 *
                               8563 * SET GET ROUTINE TO SKIP TO THE CHARACTER FOLLOWING KEYWORD 'MAT GET'
                               8564 *
    1965 3C 05 0873            8565 BMG010 MVI   B$NUMC,B@LMGT-1           SET GET TO SKIP KEYWORD
    1969 C0 87 0867            8566        B     B$GETC                    LINK TO ADVANCE POINTER
    196D C0 87 14B0            8567        B     B$CSCN                    LINK TO PROCESS FILE REFERENCE
                               8568 *
                               8569 * GENERATE THE 'ADF' PMC IN V.M. (IF OPND IS ZERO, THE FILENAME IS
                               8570 * NOT IN THE ENTRY TABLE)
                               8571 *
    1971 D2 02 AC              8572 BMG100 LA    BMGAFC(,@BR),@XR          LOAD CADDR OF 'ADF' INSTR
    1974 34 02 0A40            8573        ST    B$PCAD,@XR                SET VADIIR PARM OF PUT FOR 'ADF'
    1978 3C 01 0A41            8574        MVI   B$PNBY,B@LADF-1           SET LNG PARM, OF PUT FOR 'ADF'
    197C C0 87 093A            8575        B     B$PUTC                    LINK TO GENERATE 'ADF' PMC
                               8576 *
                               8577 * CALL GET ROUTINE TO REFERENCE THE NEXT VARIABLE
                               8578 *
    1980 3C 00 0873            8579 BMG110 MVI   B$NUMC,B@GETS             DISABLE GET ROUTINE
    1984 C0 87 0867            8580        B     B$GETC                    LINK TO GET CHARACTER POINTER
                               8581 *
                               8582 * CALL ROUTINE TO GENERATE DOPE VECTOR STACKING INSTRUCTIONS
                               8583 *
    1988 C0 87 18F3            8584 BMG120 B     B$MATR                    LINK TO GENERATE PMC
    198C 74 02 A1              8585        ST    BMG150+@OP1(,@BR),@XR     SAVE TEXT POINTER
                               8586 *
                               8587 * GENERATE THE 'MF1' INSTRUCTION IN VIRTUAL MEMORY
                               8588 *
    198F D2 02 AE              8589 BMG140 LA    BMGMFC(,@BR),@XR          LOAD CADDR OF 'MF1' INSTR
    1992 34 02 0A40            8590        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'MF1'
    1996 3C 02 0A41            8591        MVI   B$PNBY,B@LMF1-1           SET LNG PARM OF PUT FOR 'MF1'
    199A C0 87 093A            8592        B     B$PUTC                    LINK TO GENERATE 'MF1' INSTR
                               8593 *
                               8594 * TEST THE DELIMITER FOR BEING AN END-OF-STATEMENT
                               8595 *
    199E C2 02 0000            8596 BMG150 LA    *-*,@XR                   RESTORE TEXT POINTER
    19A2 BD 1E 00              8597        CLI   B@CHAR(,@XR),B@EOST       IF DELIMITER IS AN EOS
    19A5 D0 01 88              8598        BNE   BMG120(,@BR)              * BRANCH TO GET NEXT CHAR
                               8599 *
                               8600 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               8601 *
    19A8 C0 87 0700            8602 BMG160 B     B$DIST                    RETURN TO DISTRIBUTOR

                               8604 ***********************************************************************
                               8605 * 'MAT GET' STATEMENT STORAGE AND PARAMETER AREA
                               8606 ***********************************************************************
                               8607 *
    19AC 58               19AC 8608 BMGAFC DC    AL(B@LCOP)(B@CADF)        'ADF' INSTR OPCODE
    19AD 00               19AD 8609 BMGAFO DC    XL1'00'                   'ADF' INSTR OPERAND
                               8610 *
    19AE 18               19AE 8611 BMGMFC DC    AL(B@LCOP)(B@CMF1)        'MF1' INSTR OPCODE
    19AF 3E06             19B0 8612 BMGMFO DC    AL(B@LCVA)(V$XMGT)        'MF1' INSTR OPERAND

                               8614 ***********************************************************************
                               8615 * 'MAT GET' STATEMENT CONSTANTS AND EQUATES
                               8616 ***********************************************************************
                               8617 *
                               8618 * CONSTANTS
                               8619 *
                          19B1 8620 BMGSFA EQU   *
    19B1 0001             19B2 8621 BMGBN1 DC    IL(@CADDR)'1'             BINARY 1
                               8622 *
                               8623 ***********************************************************************
                               8624 *
                               8625 * END OF 'MAT GET' STATEMENT ROUTINE CODING
                               8626 *
                               8628 ***********************************************************************
                               8629 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               8630 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               8631 *                                                                     *
                               8632 ***********************************************************************
                               8633 *STATUS                                                               *
                               8634 *   VERSION 1 MODIFICATION 0                                          *
                               8635 *                                                                     *
                               8636 *FUNCTION                                                             *
                               8637 *   BKGOTO IS EXECUTED TO TRANSLATE SIMPLE GOTO STATEMENTS AS THEY    *
                               8638 *   OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO   *
                               8639 *   PLACE THE PSEUDOCODE IN VIRTUAL MEMORY.                           *
                               8640 *                                                                     *
                               8641 *ENTRY POINTS                                                         *
                               8642 *   BKGOTO HAS ONLY ONE ENTRY POINT:                                  *
                               8643 *        BKGOTO - TRANSLATE GOTO STATEMENT                            *
                               8644 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               8645 *        B     BKGOTO                                                 *
                               8646 *                                                                     *
                               8647 *INPUT                                                                *
                               8648 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               8649 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               8650 *     LEADING KEYWORD, GOTO.                                          *
                               8651 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                               8652 *     CHARACTER IN THE LEADING KEYWORD, GOTO.                         *
                               8653 *                                                                     *
                               8654 *OUTPUT                                                               *
                               8655 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               8656 *     GENERATE BY BKGOTO IS STORED IN THE NEXT AVAILABLE VIRTUAL      *
                               8657 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               8658 *     SEQUENCES.                                                      *
                               8659 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               8660 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               8661 *                                                                     *
                               8662 *EXTERNAL REFERENCES                                                  *
                               8663 *        B$GETC - (B$NUMC) - ENTRY TO BASIC RETRIEVAL ROUTINE.        *
                               8664 *        B$PUTC - (B$PCAD, B$PNBY, B$PVAD) - ENTRY TO COMPILER        *
                               8665 *                 VIRTUAL MEMORY OUTPUT ROUTINE.                      *
                               8666 *        B$BTAB - (B$BRVA, B$BRLN) - ENTRY TO BASIC COMPILER BRANCH   *
                               8667 *                 TABLE ROUTINE.                                      *
                               8668 *        B$ZDBN - (B$BINO) - ENTRY TO BASIC COMPILER ZONED DECIMAL    *
                               8669 *                 TO BINARY CONVERSION ROUTINE.                       *
                               8670 *        B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                *
                               8671 *                                                                     *
                               8672 *EXITS, NORMAL                                                        *
                               8673 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               8674 *                                                                     *
                               8675 *EXITS, ERROR                                                         *
                               8676 *   N/A                                                               *
                               8677 *                                                                     *
                               8678 *TABLES/WORK AREAS                                                    *
                               8679 *   N/A                                                               *
                               8680 *                                                                     *
                               8681 *ATTRIBUTES                                                           *
                               8682 *   BKGOTO IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               8683 *                                                                     *
                               8684 *CHARACTER CODE DEPENDENCY                                            *
                               8685 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               8686 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               8687 *                                                                     *
                               8688 *NOTES                                                                *
                               8689 *   ERROR PROCEDURES                                                  *
                               8690 *      N/A                                                            *
                               8691 *                                                                     *
                               8692 *   REGISTER USAGE                                                    *
                               8693 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               8694 *                                                                     *
                               8695 *   SAVED/RESTORED AREAS                                              *
                               8696 *      N/A                                                            *
                               8697 *                                                                     *
                               8698 *   MODIFICATION CONSIDERATIONS                                       *
                               8699 *      BKGOTO RESIDES ON A SECTOR WITH BKNEXT AND BMGETX.          1-4*
                               8700 *      ANY MODIFICATION TO BKGOTO MUST CONSIDER THIS CO-RESIDENCY  1-4*
                               8701 *      AND THE LIMITATION OF THE SECTOR BOUNDARY ON SIZE.          1-4*
                               8702 *                                                                     *
                               8703 *   REQUIRED MODULES                                                  *
                               8704 *        @SYSEQ - COMMON SYSTEM EQUATES                               *
                               8705 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES    *
                               8706 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS               *
                               8707 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES                    *
                               8708 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES                         *
                               8709 *        @ERMEQ - ERROR MESSAGE EQUATES                               *
                               8710 *        $VSEQU - FIXED VIRTUAL ADDRESS                               *
                               8711 *        $B$EQU - COMPILER FIXED EQUATES                              *
                               8712 *        $B@EQU - COMPILER SYSTEM EQUATES                             *
                               8713 *                                                                     *
                               8714 *   OTHER                                                             *
                               8715 *      BKGOTO IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS       *
                               8716 ***********************************************************************

                               8718 *
                               8719 * ENTER BKGOTO - 'GOTO' STATEMENT ROUTINE
                               8720 *
                          19B3 8721 BKGOTO EQU   *                         BKGOTO ENTRY POINT
                               8722 *
                               8723 * SET INPUT PARAMETER TO SKIP KEYWORD 'GOTO'
                               8724 *
    19B3 3C 04 0873            8725 BKG010 MVI   B$NUMC,B@LGTO             SET GET RTN TO SKIP 'GOTO'
    19B7 C0 87 0867            8726        B     B$GETC                    LINK TO ADVANCE POINTER
                               8727 *
                               8728 * CONVERT THE 'GOTO' LINE NUMBER TO BINARY FROM ITS DECIMAL FORM
                               8729 *
    19BB C0 87 19F2            8730 BKG020 B     B$ZDBN                    LINK TO CONVERT LINE NO. TO BIN
                               8731 *
                               8732 * GENERATE A 'BRA' PMC IMAGE IN VIRTUAL MEMORY
                               8733 *
    19BF D2 02 E7              8734 BKG030 LA    BKGBRC(,@BR),@XR          LOAD CADDR OF 'BRA' INSTR
    19C2 34 02 0A40            8735        ST    B$PCAD,@XR                SET VADDR PARM FOR PUT RTN
    19C6 3C 02 0A41            8736        MVI   B$PNBY,B@LBRA-1           SET LENGTH PARM FOR PUT RTN
    19CA C0 87 093A            8737        B     B$PUTC                    LINK TO GENERATE PMC
                               8738 *
                               8739 * UPDATE UNRESOLVED BRANCH TABLE
                               8740 *
    19CE 0C 01 19F1 1A6A       8741 BKG040 MVC   B$BRLN,B$BINO(@VADDR)      SET BRANCH TABLE LINE NUMBER
    19D4 0C 01 19EF 0A43       8742        MVC   B$BRVA,B$PVAD(@VADDR)      SET BRANCH TABLE VADDR
    19DA 1F 01 19EF EB         8743        SLC   B$BRVA,BKGBN1(@VADDR,@BR)  ADJUST VADDR FOR 'BRA' OPERAND
                               8744 *
                               8745 * ESTABLISH RESOLUTION OF LINE NUMBER AND VIRTUAL ADDR IN BRANCH TABLE
                               8746 *
    19DF C0 87 1996            8747 BKG050 B     B$BTAB                    LINK TO WRITE BRANCH TBL ENTRY
                               8748 *
                               8749 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               8750 *
    19E3 C0 87 0700            8751 BKG060 B     B$DIST                    RETURN TO DISTRIBUTOR

                               8753 ***********************************************************************
                               8754 * 'GOTO' STATEMENT ROUTINE PMC AND STORAGE PARAMETERS
                               8755 ***********************************************************************
                               8756 *
    19E7 46               19E7 8757 BKGBRC DC    AL(B@LCOP)(B@CBRA)        'BRA' INSTR OPCODE
    19E8 0000             19E9 8758 BKGBRO DC    XL(B@LCVA)'00'            'BRA' INSTR OPERAND IMAGE

                               8760 ***********************************************************************
                               8761 * 'GOTO' STATEMENT CONSTANTS
                               8762 ***********************************************************************
                               8763 *
    19EA 0001             19EB 8764 BKGBN1 DC    IL(@VADDR)'1'             BINARY '1'
                               8765 *
                               8766 ***********************************************************************
                               8767 *
                               8768 * END OF 'GOTO' STATEMENT ROUTINE CODING
                               8769 *
                               8771 ***********************************************************************
                               8772 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               8773 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               8774 *                                                                     *
                               8775 ***********************************************************************
                               8776 *STATUS                                                               *
                               8777 *   VERSION 1 MODIFICATION 0                                          *
                               8778 *                                                                     *
                               8779 *FUNCTION                                                             *
                               8780 *   BKARIF IS EXECUTED TO TRANSLATE ARITHMETIC IF STATEMENTS AS THEY  *
                               8781 *   OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO   *
                               8782 *   PLACE THE PSEUDOCODE IN VIRTUAL MEMORY.                           *
                               8783 *                                                                     *
                               8784 *ENTRY POINTS                                                         *
                               8785 *   BKARIF HAS ONLY ONE ENTRY POINT:                                  *
                               8786 *        BKARIF - TRANSLATE ARITHMETIC IF STATEMENT                   *
                               8787 *   THE FORMAT FOR THE CALLING SEQUENCE IS:                           *
                               8788 *        B     BKARIF                                                 *
                               8789 *                                                                     *
                               8790 *INPUT                                                                *
                               8791 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               8792 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               8793 *     LEADING KEYWORD, IF.                                            *
                               8794 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST.*
                               8795 *     CHARACTER IN THE LEADING KEYWORD, IF.                           *
                               8796 *                                                                     *
                               8797 *OUTPUT                                                               *
                               8798 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               8799 *     GENERATED BY BKARIF IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               8800 *     MEMORY LOCATION. FOLLOWING PREVIOUSLY STORED INSTRUCTION        *
                               8801 *     SEQUENCES.                                                      *
                               8802 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               8803 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               8804 *                                                                     *
                               8805 *EXTERNAL REFERENCES                                                  *
                               8806 *        B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIEVAL RTN.    *
                               8807 *        B$PUTC - (B$PCAD, B$PNBY, B$PVAD) - ENTRY TO COMPILER VIRT   *
                               8808 *                 MEMORY OUTPUT ROUTINE.                              *
                               8809 *        B$BTAB - (B$BRVA, B$BRLN) - ENTRY TO BASIC COMPILER BRANCH   *
                               8810 *                 TABLE ROUTINE.                                      *
                               8811 *        B$ZOBN - (B$BINO) - ENTRY TO BASIC COMPILER ZONED DECIMAL    *
                               8812 *                 TO BINARY CONVERSION ROUTINE.                       *
                               8813 *        B$SCAN - ENTRY TO BASIC COMPILER ARITHMETIC EXPRESSION SCAN  *
                               8814 *                 ROUTINE.                                            *
                               8815 *        B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                *
                               8816 *                                                                     *
                               8817 *EXITS, NORMAL                                                        *
                               8818 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               8819 *                                                                     *
                               8820 *EXITS, ERROR                                                         *
                               8821 *   N/A                                                               *
                               8822 *                                                                     *
                               8823 *TABLES/WORK AREAS                                                    *
                               8824 *   * RELATIONAL OPERATOR TABLE - INTERNAL TO OKARIF, THIS TABLE      *
                               8825 *     CONTAINS BRC INSTRUCTION CONDITION CODES ASSOCIATED WITH EVERY  *
                               8826 *     SIMPLE OR COMPOUND RELATIONAL OPERATOR. OPERATOR ENTRIES IN     *
                               8827 *     THE TABLE CONSIST OF THE EBCDIC CHARACTER CODE FOR SIMPLE       *
                               8828 *     OPERATORS AND THE SUM OF EBCDIC CHARACTER CODES FOR COMPOUND    *
                               8829 *     OPERATORS.                                                      *
                               8830 *   * RELATIONAL OPERATOR BUCKET - INTERNAL TO BKARIF, THIS 1-BYTE    *
                               8831 *     FIELD IS USED TO STORE SIMPLE AND COMPOUND RELATIONAL OPERATOR  *
                               8832 *     CHARACTERS FOR ASSOCIATION WITH A RELATIONAL OPERATOR TABLE     *
                               8833 *     ENTRY.                                                          *
                               8834 *                                                                     *
                               8835 *ATTRIBUTES                                                           *
                               8836 *   BKARIF IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               8837 *                                                                     *
                               8838 *CHARACTER CODE DEPENDENCY                                            *
                               8839 *   THE OPERATION OF THIS MODULE DEPENDS UPON AN INTERNAL REPRRESEN-  *
                               8840 *   TATION OF THE EXTERNAL CHARACTER SET WHICH IS EQUIVALENT TO THE   *
                               8841 *   ONE USED AT ASSEMBLY TIME. THE CODING HAS BEEN ARRANGED SO THAT   *
                               8842 *   REDEFINITION OF CHARACTER CONSTANTS. BY REASSEMBLY, WILL RESULT   *
                               8843 *   IN A CORRECT MODULE FOR THE NEW DEFINITIONS.                      *
                               8844 *                                                                     *
                               8845 *NOTES                                                                *
                               8846 *   ERROR PROCEDURES                                                  *
                               8847 *      N/A                                                            *
                               8848 *                                                                     *
                               8849 *   REGISTER USAGE                                                    *
                               8850 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               8851 *                                                                     *
                               8852 *   SAVED/RESTORED AREAS                                              *
                               8853 *      N/A                                                            *
                               8854 *                                                                     *
                               8855 *   MODIFICATION CONSIDERATIONS                                       *
                               8856 *      BKARIF RESIDES ON A SECTOR WITH BMDPRT.  ANY MODIFICATION   1-4*
                               8857 *      TO BKARIF WILL CHANGE THE ENTRY ADDRESS OF BMDPRT AND       1-4*
                               8858 *      MUST TAKE INTO CONSIDERATION THE LIMITATION OF THE SECTOR   1-4*
                               8859 *      BOUNDARY ON SIZE.                                           1-4*
                               8860 *                                                                     *
                               8861 *   REQUIRED MODULES                                                  *
                               8862 *        @SYSEQ - COMMON SYSTEM EQUATES                               *
                               8863 *        @FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUES EQUATES *
                               8864 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES       *
                               8865 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               8866 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES                         *
                               8867 *        @ERNEQ - ERROR MESSAGE EQUATES                               *
                               8868 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES                       *
                               8869 *        $B$EQU - COMPILER FIXED EQUATES                              *
                               8870 *                                                                     *
                               8871 *   OTHER                                                             *
                               8872 *      BKARIF IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               8873 ***********************************************************************
                               8874 *
    1A00                       8875        ORG   *,256,0                   BEGIN AT CORE PAGE BOUNDARY
                          1A00 8876        USING *,@BR                     DEFINE BASE ADDR FOR CORE PAGE
                               8877 *
                               8878 * ENTER BKARIF - ARITHMETIC IF STATEMENT ROUTINE
                               8879 *
                          1A00 8880 BKARIF EQU   *                         BKARIF ENTRY POINT
                               8881 *
                               8882 * SET INPUT PARAMETER TO SKIP 'I' IN KEYWORD 'IF' TO REFERENCE THE
                               8883 * CHARACTER PRECEDING THE FIRST ARITHMETIC EXPRESSION
                               8884 *
    1A00 3C 01 0873            8885 BKA010 MVI   B$NUMC,B@LKIF-1           SET GET RTN TO SKIP 'I' IN IF.
    1A04 C0 87 0867            8886        B     B$GETC                    LINK TO ADVANCE POINTER
                               8887 *
                               8888 * BRANCH TO SCAN ROUTINE TO GENERATE 'STF' INSTR
                               8889 *
    1A08 C0 87 1514            8890 BKA020 B     B$SCAN                    LINK TO GENERATE 'STF' PMC
                               8891 *
                               8892 * STORE THE FIRST RELATIONAL OPERATOR IN THE OPERAND OF A CLI INSTR.
                               8893 *
    1A0C 6C 00 32 00           8894 BKA030 MVC   BKA090+@Q(,@BR),B@CHAR(1,@XR)  STORE 1ST RELATIONAL OPTR
                               8895 *
                               8896 * GET NEXT CHARACTER TO CHECK IF COMPOUND OPERATOR IS INDICATED
                               8897 *
    1A10 C0 87 0867            8898 BKA040 B     B$GETC                    LINK TO GET NEXT CHARACTER
    1A14 BD 7E 00              8899        CLI   B@CHAR(,@XR),B@EQUL       IF CHAR IS '='
    1A17 F2 81 0D              8900        JE    BKA060                    * GO COMPUTE OPERATOR
    1A1A BD 6E 00              8901        CLI   B@CHAR(,@XR),B@GRTR       IF CHAR IS '>'
    1A1D F2 81 07              8902        JE    BKA060                    * GO COMPUTE OPERATOR
                               8903 *
                               8904 * IF NO SECOND RELATIONAL OPERATOR DISABLE BAGETC TO KEEP THE TEXT
                               8905 * POINTER IN PLACE
                               8906 *
    1A20 3C 00 0873            8907 BKA050 MVI   B$NUMC,B@GETS             DISABLE GET ROUTINE
    1A24 F2 87 04              8908        J     BKA070                    GO SEARCH OPERATOR TABLE
                               8909 *
                               8910 * IF RELATIONAL OPERATOR IS COMPOUND ADD CURRENTLY REFERENCED CHARACTER
                               8911 * TO THE CONTENTS OF THE OPERATOR OPERAND TO DEKIVE A CHARACTER CODE
                               8912 *
    1A27 6E 00 32 00           8913 BKA060 ALC   BKA090+@Q(,@BR),B@CHAR(1,@XR)  ADD TO GET CHAR CODE
                               8914 *
                               8915 * SEARCH RELATIONAL OPERATOR TABLE FOR THE CONDITION CODE THAT MATCHES
                               8916 * THE CHARACTER CODE IN THE OPERATOR BUUKET-EITHER SIMPLE OR COMPOUND
                               8917 *
    1A2B D2 02 8B              8918 BKA070 LA    BKAOT1(,@BR),@XR          LOAD TABLE BASE ADDR IN XR
    1A2E E2 02 02              8919 BKA080 LA    BKALTH(,@XR),@XR          ADD LENGTH TO ADDR IN XR
    1A31 BD 00 00              8920 BKA090 CLI   BKAOD1(,@XR),*-*          IF TEXT OPERATOR - TABLE ENTRY
    1A34 D0 01 2E              8921        BNE   BKA080(,@BR)              * FALL THROUGH
                               8922 *
                               8923 * STORE CONDITION CODE IN OPERAND FIELD OF 'BRC' INSTRUCTION IMAGE
                               8924 *
    1A37 6C 00 8A 01           8925 BKA100 MVC   BKAB02(,@BR),BKAOD2(,@XR)  SET 'BRC' COND CODE OPERAND
                               8926 *
                               8927 * GO TO ARITHMETIC SCAN ROUTINE TO GENERATE PMC FOR THE SECOND
                               8928 * ARITHMETIC EXPRESSION
                               8929 *
    1A3B 35 02 0878            8930 BKA110 L     B$GPTR,@XR                RESTORE TEXT POINTER
    1A3F C0 87 1514            8931        B     B$SCAN                    LINK TO GENERATE PMC
                               8932 *
                               8933 * SET PARAMETER TO SKIP EMBEDDED KEYWORD 'GOTO' OR 'THEN' TO ADVANCE
                               8934 * THE TEXT POINTER TO THE LINE NUMBER
                               8935 *
    1A43 3C 03 0873            8936 BKA120 MVI   B$NUMC,B@LTHN-1           SET GET RTN TO SKIP KEYWORD
    1A47 C0 87 0867            8937        B     B$GETC                    LINK TO ADVANCE POINTER
                               8938 *
                               8939 * CONVERT THE 'GOTO' LINE NUMBER TO BINARY FROM DECIMAL
                               8940 *
    1A4B C0 87 19F2            8941 BKA130 B     B$ZDBN                    LINK TO CONVERT LINE NUMBER
                               8942 *
                               8943 * GENERATE A COMPARE FLOATING POINT VALUE PMC IN VIRTUAL MEMORY
                               8944 *
    1A4F D2 02 86              8945 BKA140 LA    BKACMC(,@BR),@XR          LOAD CADDR OF 'CMF' INSTR
    1A52 34 02 0A40            8946        ST    B$PCAD,@XR                SET PUT RTN FOR VADDR OF 'CMF'
    1A56 3C 00 0A41            8947        MVI   B$PNBY,B@LCMF-1           SET PUT RTN FOR LENGTH OF 'CMF'
    1A5A C0 87 093A            8948        B     B$PUTC                    LINK TO GENERATE 'CMF' INSTK
                               8949 *
                               8950 * GENERATE BRANCH ON CONDITION INSTR IN VIRTUAL MEMORY
                               8951 *
    1A5E D2 02 87              8952 BKA150 LA    BKABRC(,@BR),@XR          LOAD CADDR OF 'BRC' INSTR
    1A61 34 02 0A40            8953        ST    B$PCAD,@XR                SET PUT RTN FOR VADDR OF 'BRC'
    1A65 3C 03 0A41            8954        MVI   B$PNBY,B@LBRC-1           SET PUT RTN FOR LENGTH OF 'BRC'
    1A69 C0 87 093A            8955        B     B$PUTC                    UNK TO GENERATE 'BRC' INSTR
                               8956 *
                               8957 * ESTABLISH ADDRESS AND LINE NUMBER PARAMETERS FOR BRANCH TABLE
                               8958 * RESOLUTION ROUTINE
                               8959 *
    1A6D 0C 01 19EF 0A43       8960 BKA160 MVC   B$BRVA,B$PVAD(@VADDR)     SET ADDR PARAMETER
    1A73 1F 01 19EF 8C         8961        SLC   B$BRVA,BKALNG(@VADDR,@BR) * TO ADDRESS BRANCH VADDR
    1A78 0C 01 19F1 1A6A       8962        MVC   B$BRLN,B$BINO(B@LCLN)     SET LINE NO PARAMETER
    1A7E C0 87 1996            8963        B     B$BTAB                    LINK TO WRITE BRANCH TAT ENTRY
                               8964 *
                               8965 * RETURN CONTROL TO THE DISTRIBUTOR
                               8966 *
    1A82 C0 87 0700            8967 BKA170 B     B$DIST                    RETURN TO DISTRIBUTOR
```

```
                               8969 ***********************************************************************
                               8970 * ARITHMETIC 'IF' ROUTINE PMC AND STORAGE PARAMETERS
                               8971 ***********************************************************************
                               8972 *
    1A86 40               1A86 8973 BKACMC DC    AL(B@LCOP)(B@CCMF)        COMPARE FLOATING VALUES OPCODE
    1A87 44               1A87 8974 BKABRC DC    AL(B@LCOP)(B@CBRC)        BRANCH ON CONDITION OPCODE
    1A88 0000             1A89 8975 BKAB01 DC    XL(B@LCVA)'00'            BRANCH ON CONDITION VABOR OPND
    1A8A                  1A8A 8976 BKAB02 DS    CL(B@LCCC)                BRANCH ON COND COMO CODE OPND

                               8978 ***********************************************************************
                               8979 * ARITHMETIC 'IF' ROUTINE CONSTANTS
                               8980 ***********************************************************************
                               8981 *
    1A8B 0002             1A8C 8982 BKALNG DC    AL(@VADDR)(B@LCCC+1)      LENGTH OF CONDITION CODE + 1

                               8984 ***********************************************************************
                               8985 * RELATIONAL OPERATOR - CONDITION CODE TABLE
                               8986 ***********************************************************************
                               8987 *
                          1A8D 8988 BKATAB EQU   *                         START OF CODE TABLE
                          0000 8989 BKAOD1 EQU   0                         DISP FOR TABLE OPERATOR
                          0001 8990 BKAOD2 EQU   1                         DISP FOR TABLE COND CODE
                          0002 8991 BKALTH EQU   2                         LENGTH OF TABLE ENTRY
                          1A8B 8992 BKAOT1 EQU   BKATAB-BKALTH             CODE TABLE BASE ADDRESS
                               8993 *
    1A8D 7E               1A8D 8994        DC    AL1(B@EQUL)               RELATIONAL OPERATOR - '='
    1A8E 84               1A8E 8995        DC    AL1(B@BREQ)               BRANCH CONDITION - EQUAL
                               8996 *
    1A8F 6E               1A8F 8997        DC    AL1(B@GRTR)               RELATIONAL OPERATOR - '>'
    1A90 88               1A90 8998        DC    AL1(B@BRHI)               BRANCH CONDITION - HIGH
                               8999 *
    1A91 4C               1A91 9000        DC    AL1(B@LESS)               RELATIONAL OPERATOR - '<'
    1A92 82               1A92 9001        DC    AL1(B@BRLO)               BRANCH CONDITION - LOW
                               9002 *
    1A93 BA               1A93 9003        DC    AL1(B@LESS+B@GRTR)        RELATIONAL OPERATOR - '><'
    1A94 94               1A94 9004        DC    AL1(B@BRNE)               BRANCH CONDITION - NOT EQUAL
                               9005 *
    1A95 CA               1A95 9006        DC    AL1(B@LESS+B@EQUL)        RELATIONAL OPERATOR - '<='
    1A96 98               1A96 9007        DC    AL1(B@BRNH)               BRANCH CONDITION - NOT HIGH
                               9008 *
    1A97 EC               1A97 9009        DC    AL1(B@GRTR+B@EQUL)        RELATIONAL OPERATOR - '>='
    1A98 92               1A98 9010        DC    AL1(B@BRNL)               BRANCH CONDITION - NOT LOW
                               9011 *
    1A99 7F               1A99 9012        DC    AL1(B@NEQL)               RELATIONAL OPERATOR - ' '
    1A9A 94               1A9A 9013        DC    AL1(B@BRNE)               BRANCH CONDITION - NOT EQUAL
                               9014 *
                               9015 ***********************************************************************
                               9016 *
                               9017 * END OF ARITHMETIC IF ROUTINE CODING
                               9018 *
                               9020 ***********************************************************************
                               9021 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               9022 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               9023 *                                                                     *
                               9024 ***********************************************************************
                               9025 *STATUS                                                               *
                               9026 *   VERSION 1 MODIFICATION 0                                          *
                               9027 *                                                                     *
                               9028 *FUNCTION                                                             *
                               9029 *   BMDPRT IS EXECUTED TO TRANSLATE MAT PRINT STATEMENTS AS THEY OCCUR*
                               9030 *   IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE   *
                               9031 *   THE PSEUDOCODE IN VIRTUAL MEMORY.                                 *
                               9032 *                                                                     *
                               9033 *ENTRY POINTS                                                         *
                               9034 *   BMDPRT HAS ONLY ONE ENTRY POINT:                                  *
                               9035 *        BMDPRT - TRANSLATE MAT PRINT STATEMENT                       *
                               9036 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               9037 *        B     BMDPRT                                                 *
                               9038 *                                                                     *
                               9039 *INPUT                                                                *
                               9040 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               9041 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               9042 *     LEADING KEYWORD, MAT PRINT.                                     *
                               9043 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                               9044 *     CHARACTER IN THE LEADING KEYWORD, MAT PRINT.                    *
                               9045 *                                                                     *
                               9046 *OUTPUT                                                               *
                               9047 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               9048 *     GENERATED BY BMDPRT IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               9049 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               9050 *     SEQUENCES.                                                      *
                               9051 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               9052 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               9053 *                                                                     *
                               9054 *EXTERNAL REFERENCES                                                  *
                               9055 *        B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIEVAL RTN.    *
                               9056 *        B$PUTC - (B$PCAD, B$PNBY) - ENTRT TO COMPILER VIRTUAL MEMORY *
                               9057 *                 OUTPUT ROUTINE.                                     *
                               9058 *        B$MATR - ENTRY TO BASIC COMPILER MATRIX REFERENCE ROUTINE.   *
                               9059 *        B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                *
                               9060 *                                                                     *
                               9061 *EXITS, NORMAL                                                        *
                               9062 *   B@DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               9063 *                                                                     *
                               9064 *EXITS, ERROR                                                         *
                               9065 *   N/A                                                               *
                               9066 *                                                                     *
                               9067 *TABLES/WORK AREAS                                                    *
                               9068 *   N/A                                                               *
                               9069 *                                                                     *
                               9070 *ATTRIBUTES                                                           *
                               9071 *   BMDPRT IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               9072 *                                                                     *
                               9073 *CHARACTER CODE DEPENDENCY                                            *
                               9074 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               9075 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               9076 *                                                                     *
                               9077 *NOTES                                                                *
                               9078 *   ERROR PROCEDURES                                                  *
                               9079 *      N/A                                                            *
                               9080 *                                                                     *
                               9081 *   REGISTER USAGE                                                    *
                               9082 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               9083 *                                                                     *
                               9084 *   SAVED/RESTORED AREAS                                              *
                               9085 *      N/A                                                            *
                               9086 *                                                                     *
                               9087 *   MODIFICATION CONSIDERATIONS                                       *
                               9088 *      BADPRT RESIDES ON A SECTOR WITH BKARIF. ANY MODIFICATION    1-4*
                               9089 *      TO RMDPRT MUST TAKE INTO CONSIDERATION THIS CO-RESIDENCY    1-4*
                               9090 *      AND THE LIMITATION OF THE SECTOR BOUNDARY ON SIZE.          1-4*
                               9091 *                                                                     *
                               9092 *   REQUIRED MODULES                                                  *
                               9093 *        @SYSEQ - COMMON JESTER EQUATES.                              *
                               9094 *        @FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUES EQUATES.*
                               9095 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS.              *
                               9096 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               9097 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               9098 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               9099 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               9100 *        $B$EQU - COMPILER FIXED EQUATES.                             *
                               9101 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               9102 *                                                                     *
                               9103 *   OTHER                                                             *
                               9104 *      BMDPRT IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS       *
                               9105 ***********************************************************************

                               9107 *
                               9108 * ENTER BMDPRT - MAT PRINT STATEMENT ROUTINE
                               9109 *
                          1A9B 9110 BMDPRT EQU   *                         BMDPRT ENTRY POINT
                               9111 *
                               9112 * SET GET ROUTINE TO SKIP TO CHAR FOLLOWING KEYWORDS 'MAT PRINT'
                               9113 *
    1A9B 3C 08 0873            9114 BMD010 MVI   B$NUMC,B@LMPR             SET GET TO SKIP 'MAT PRINT'
    1A9F C0 87 0867            9115        B     B$GETC                    LINK TO ADVANCE POINTER
                               9116 *
                               9117 * DISABLE GET RTN BEFORE CALLING THE MATRIX REFERENCE PROCESSOR
                               9118 *
    1AA3 3C 00 0873            9119 BMD020 MVI   B$NUMC,B@GETS             DISABLE GET RTN NOT TO GET CHAR
    1AA7 C0 87 18F3            9120        B     B$MATR                    LINK TO PROCESS MAT-REFERENCE
                               9121 *
                               9122 * TEST DELIMITER FOR BEING A SEMI-COLON (INDICATING SHORT FORM)
                               9123 *
    1AAB BD 5E 00              9124 BMD030 CLI   B@CHAR(,@XR),B@SCLN       IF CHAR IS NOT SEMI-COLON
    1AAE F2 01 12              9125        JNE   BMD050                    * GO GENERATE 'MF1' FOR LONG FORM
                               9126 *
                               9127 * GENERATE AN 'MF1' INSTR FOR SHORT FORM
                               9128 *
    1AB1 D2 02 EA              9129 BMD040 LA    BMDM1C(,@BR),@XR          LOAD CADDR OF 'MF1' INSTR
    1AB4 34 02 0A40            9130        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'MF1'
    1AB8 3C 02 0A41            9131        MVI   B$PNBY,B@LMF1-1           SET LNG PARM OF PUT FOR 'MF1'
    1ABC C0 87 093A            9132       B     B$PUTC                    LINK TO GENERATE 'MF1' INSTR
    1AC0 F2 87 19              9133       J     BMD060                    GO GET NEXT CHARACTER
                               9134 *
                               9135 * GENERATE AN 'MF1' INSTR FOR LONG FORM
                               9136 *
    1AC3 D2 02 ED              9137 BMD050 LA    BMDM2C(,@BR),@XR          LOAD CADDR OF 'MF1' INSTR
    1AC6 34 02 0A40            9138       ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'MF1'
    1ACA 3C 02 0A41            9139       MVI   B$PNBY,B@LMF1-1           SET LNG PARM OF PUT FOR 'MF1'
    1ACE C0 87 093A            9140       B     B$PUTC                    LINK TO GENERATE 'MF1' INSTR
                               9141 *
                               9142 * TEST DELIMITER FOR BEING A STATEMENT TERMINATOR
                               9143 *
    1AD2 35 02 0878            9144 BMD055 L     B$GPTR,@XR                RESTORE TEXT POINTER
    1AD6 BD 1E 00              9145       CLI   B@CHAR(,@XR),B@EOST       IF DELIMITER IS AN EOS
    1AD9 D0 81 E6              9146       BE    BMD080(,@BR)              * RETURN CONTROL TO DIST
                               9147 *
                               9148 * CALL GET ROUTINE TO GET NEXT CHARACTER
                               9149 *
    1ADC C0 87 0867            9150 BMD060 B     B$GETC                    LINK TO GET NEXT CHAR
                               9151 *
                               9152 * TEST DELIMITER FOR BEING A STATEMENT TERMINATOR
                               9153 *
    1AE0 BD 1E 00              9154 BMD070 CLI   B@CHAR(,@XR),B@EOST       IF DELIMITER IS NOT AN EOS
    1AE3 D0 01 A3              9155       BNE   BMD020(,@BR)              * GO PROCESS NEXT LIST ELEMENT
                               9156 *
                               9157 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               9158 *
    1AE6 C0 87 0700            9159 BMD080 B     B$DIST                    RETURN TO DISTRIBUTOR

                               9161 ***********************************************************************
                               9162 * MAT PRINT STATEMENT ROUTINE STORAGE AND PARAMETER AREA
                               9163 ***********************************************************************
                               9164 *
    1AEA 18               1AEA 9165 BMDM1C DC    AL(B@LCOP)(B@CMF1)        'MF1' INSTR OPCODE
    1AEB 3F00             1AEC 9166 BMDM10 DC    AL(B@LCVA)(V$XMPS)        'MF1' INSTR OPND - SHORT FORM
                               9167 *
    1AED 18               1AED 9168 BMDM2C DC    AL(B@LCOP)(B@CMF1)        'MF1' INSTR OPCODE
    1AEE 3F06             1AEF 9169 BMDM20 DC    AL(B@LCVA)(V$XMPL)        'MF1' INSTR OPND - LONG FORM
                               9170 *
                               9171 **********************************************************************
                               9172 *
                               9173 * END OF 'MAT PRINT' STATEMENT ROUTINE CODING
                               9174 *
                               9176 ***********************************************************************
                               9177 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               9178 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               9179 *                                                                     *
                               9180 ***********************************************************************
                               9181 *STATUS                                                               *
                               9182 *   VERSION 1 MODIFICATION 0                                          *
                               9183 *                                                                     *
                               9184 *FUNCTION                                                             *
                               9185 *   BKCRIF IS EXECUTED TO TRANSLATE CHARACTER IF STATEMENTS AS THEY   *
                               9186 *   OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO   *
                               9187 *   PLACE THE PSEUDOCODE IN VIRTUAL MEMORY.                           *
                               9188 *                                                                     *
                               9189 *ENTRY POINTS                                                         *
                               9190 *   BKCRIF HAS ONLY ONE ENTRY POINT                                   *
                               9191 *        BKCRIF - TRANSLATE CHARACTER IF STATEMENT                    *
                               9192 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               9193 *        B     BKCRIF                                                 *
                               9194 *                                                                     *
                               9195 *INPUT                                                                *
                               9196 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               9197 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               9198 *     LEADING KEYWORD, IF.                                            *
                               9199 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                               9200 *     CHARACTER IN THE LEADING KEYWORD, IF.                           *
                               9201 *                                                                     *
                               9202 *OUTPUT                                                               *
                               9203 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               9204 *     GENERATED BY BKCRIF IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               9205 *     MEMORY LOCATION, FOLLOWING PREVIOUSLY STORED INSTRUCTION        *
                               9206 *     SEQUENCES.                                                      *
                               9207 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               9208 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               9209 *                                                                     *
                               9210 *EXTERNAL REFERENCES                                                  *
                               9211 *        B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE *
                               9212 *        B$PUTC - (B$PCAD, B$PNBY, B$PVAD) - ENTRY TO COMPILER VIRT   *
                               9213 *                 MEMORY OUTPUT ROUTINE.                              *
                               9214 *        B$BTAB - (B$BRVA, B$BRIN) - ENTRY TO BASIC COMPILER BRANCH   *
                               9215 *                 TABLE ROUTINE.                                      *
                               9216 *        B$ZDBN - (B$BINO) - ENTRY TO COMPILER ZONED DECIMAL TO       *
                               9217 *                 BINARY CONVERSION ROUTINE.                          *
                               9218 *        B$CSCN - ENTRY TO BASIC COMPILER CHARACTER SCAN ROUTINE      *
                               9219 *        B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                *
                               9220 *                                                                     *
                               9221 *EXITS, NORMAL                                                        *
                               9222 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                               9223 *                                                                     *
                               9224 *EXITS, ERROR                                                         *
                               9225 *   N/A                                                               *
                               9226 *                                                                     *
                               9227 *TABLES/WORK AREAS                                                    *
                               9228 *   * RELATIONAL OPERATOR TABLE - INTERNAL TO BKCRIF, THIS TABLE      *
                               9229 *     CONTAINS 'BRC' INSTRUCTION CONDITION CODES ASSOCIATED WITH      *
                               9230 *     EVERY SIMPLE OR COMPOUND RELATIONAL OPERATOR.  OPERATOR ENTRIES *
                               9231 *     IN THE TABLE CONSIST OF THE EBCDIC CHARACTER CODE FOR SIMPLE    *
                               9232 *     OPERATORS AND THE SUM OF EBCDIC CHARACTER CODES FOR COMPOUND    *
                               9233 *     OPERATORS.                                                      *
                               9234 *   * RELATIONAL OPERATOR BUCKET - INTERNAL TO BKCRIF,  THIS 1-BYTE   *
                               9235 *     FIELD IS USED TO STORE SIMPLE AND COMPOUND RELATIONAL OPERATOR  *
                               9236 *     CHARACTERS FOR ASSOCIATION WITH A RELATIONAL OPERATOR TABLE     *
                               9237 *     ENTRY.                                                          *
                               9238 *                                                                     *
                               9239 *ATTRIBUTES                                                           *
                               9240 *   BKCRIF IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               9241 *                                                                     *
                               9242 *CHARACTER CODE DEPENDENCY                                            *
                               9243 *   THE OPERATION OF THIS MODULE DEPENDS UPON AS INTERNAL REPRESENTA- *
                               9244 *   TION OF THE EXTERNAL CHARACTER SET WHICH IS EQUIVALENT TO THE     *
                               9245 *   ONE USED AT ASSEMBLY TIME.  THE CODING HAS BEEN ARRANGED SO THAT  *
                               9246 *   REDEFINITION OF CHARACTER CONSTANTS, BY REASSEMBLY, WILL RESULT   *
                               9247 *   IN A CORRECT MODULE FOR THE NEW DEFINITIONS.                      *
                               9248 *                                                                     *
                               9249 *NOTES                                                                *
                               9250 *   ERROR PROCEDURES                                                  *
                               9251 *      N/A                                                            *
                               9252 *                                                                     *
                               9253 *   REGISTER USAGE                                                    *
                               9254 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               9255 *                                                                     *
                               9256 *   SAVED/RESTORED AREAS                                              *
                               9257 *      N/A                                                            *
                               9258 *                                                                     *
                               9259 *   MODIFICATION CONSIDERATIONS                                       *
                               9260 *      BKCRIF RESIDES ON A SECTOR WITH BMPUTX.  ANY MODIFICATION   1-4*
                               9261 *      TO BKCRIF SHOULD CONSIDER THIS CO-RESIDENCY SINCE IT WILL   1-4*
                               9262 *      CHANGE THE ENTRY ADDRESS OF BMPUTX.  THE SIZE LIMITATION    1-4*
                               9263 *      OF THE SECTOR BOUNDARY MUST ALSO BE CONSIDERED.                *
                               9264 *                                                                     *
                               9265 *   REQUIRED MODULES                                                  *
                               9266 *        @SYSEQ - COMMON SYSTEM EQUATES.                              *
                               9267 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.   *
                               9268 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.      *
                               9269 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               9270 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               9271 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               9272 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               9273 *        $B$EQU - COMPILER FIXED EQUATES.                             *
                               9274 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               9275 *                                                                     *
                               9276 *   OTHER                                                             *
                               9277 *      BKCRIF IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               9278 ***********************************************************************
                               9279 *
    1B00                       9280       ORG   *,256,0                   BEGIN AT CORE PAGE BOUNDARY
                          1B00 9281       USING *,@BR                     DEFINE BASE ADDR FOR CORE PAGE
                               9282 *
                               9283 * ENTER BKCRIF - CHARACTER 'IF' STATEMENT PROCESSOR
                               9284 *
                          1B00 9285 BKCRIF EQU   *                         BKCRIF ENTRY POINT
                               9286 *
                               9287 * SKIP PAST 'I' IN KEYWORD 'IF' TO REFERENCE CHARACTER PRECEDING THE
                               9288 * FIRST EXPRESSION CHARACTER
                               9289 *
    1B00 3C 01 0873            9290 BKC010 MVI   B$NUMC,B@LKIF-1           SET PARAMETER TO SKIP 'I' IN IF
    1B04 C0 87 0867            9291        B     B$GETC                    LINK TO ADVANCE POINTER
                               9292 *
                               9293 * GENERATE PNC FOR FIRST CHARACTER EXPRESSION
                               9294 *
    1B08 C0 87 14B0            9295 BKC020 B     B$CSCN                    LINK TO GENERATE PMC
                               9296 *
                               9297 * STORE FIRST RELATIONAL OPERATOR CHARACTER IN OPERAND OF CLI INSTR.
                               9298 *
    1B0C 6C 00 32 00           9299 BKC030 MVC   BKC090+@Q(,@BR),B@CHAR(1,@XR)  STORE 1ST RELATIONAL OPTR
                               9300 *
                               9301 * GET NEXT CHARACTER TO CHECK IF COMPOLND OPERATOR IS INDICATED
                               9302 *
    1B10 C0 87 0867            9303 BKC040 B     B$GETC                    LINK TO GET NEXT CHARACTER
    1B14 BD 7E 00              9304        CLI   B@CHAR(,@XR),B@EQUL       IF CHAR IS '='
    1B17 F2 81 0D              9305        JE    BKC060                    * GO COMPUTE OPERATOR
    1B1A BD 6E 00              9306        CLI   B@CHAR(,@XR),B@GRTR       IF CHAR IS '>'
    1B1D F2 81 07              9307        JE    BKC060                    * GO COMPUTE OPERATOR
                               9308 *
                               9309 * IF RELATIONAL CPERATOR IS NOT COMPOUND DISABLE BAGETC TO KEEP TEXT
                               9310 * POINTER STATIONARY
                               9311 *
    1B20 3C 00 0873            9312 BKC050 MVI   B$NUMC,B@GETS             DISABLE GET RTN FOR NEXT CHAR
    1B24 F2 87 04              9313        J     BKC070                    GO SEARCH OPERATOR TABLE
                               9314 *
                               9315 * IF RELATIONAL OPERATOR IS COMPOUND ADD CURRENTLY REFERENCED CHARACTER
                               9316 * TO THE CONTENTS OF THE OPERATOR BUCKET TO DERIVE A CHARACTER CODE
                               9317 *
    1B27 6E 00 32 00           9318 BKC060 ALC   BKC090+@Q(,@BR),B@CHAR(1,@XR)  ADD TO GET CHAR CODE
                               9319 *
                               9320 * SEARCH THE RELATIONAL OPERATOR TABLE FOR THE CONDITION CODE THAT
                               9321 * MATCHES THE CHARACTER CODE IN THE OPERATOR BUCKET-EITHER SIMPLE OR
                               9322 * COMPOUND
                               9323 *
    1B2B D2 02 8B              9324 BKC070 LA    BKCOTB(,@BR),@XR          LOAD TABLE BASE ADOR IN OR
    1B2E E2 02 02              9325 BKC080 LA    BKCLTH(,@XR),@XR          ADD LENGTH TO ADDR IN XR
    1B31 BD 00 00              9326 BKC090 CLI   BKCOD1(,@XR),*-*          IF TEXT OPERATOR = TABLE ENTRY
    1B34 D0 01 2E              9327        BNE   BKC080(,@BR)              * FALL THROUGH
                               9328 *
                               9329 * STORE CONDITION CODE IN OPERAND FIELD OF 'BRC' INSTRUCTION IMAGE
                               9330 *
    1B37 6C 00 8A 01           9331 BKC100 MVC   BKCBO2(,@BR),BKCCD2(,@XR)  SET 'BRC' CORD CODE OPERAND
                               9332 *
                               9333 * GOTO CHARACTER SCAN ROUTINE TO GENERATE PMC FOR THE SECOND CHARACTER
                               9334 * EXPRESSION
                               9335 *
    1B3B 35 02 0878            9336 BKC110 L     B$GPTR,@XR                RESTORE TEXT POINTER
    1B3F C0 87 14B0            9337        B     B$CSCN                    LINK TO GENERATE PMC
                               9338 *
                               9339 * SET PARAMETER TO SKIP EMBEDDED KEYWORD 'GOTO' OR 'THEN' TO ADVANCE
                               9340 * THE TEXT POINTER TO THE LINE NUMBER
                               9341 *
    1B43 3C 04 0873            9342 BKC120 MVI   B$NUMC,B@LTHN             SET GET RTN TO SKIP KEYWORD
    1B47 C0 87 0867            9343        B     B$GETC                    LINK TO ADVANCE POINTER
```

```
                               9344 *
                               9345 * CONVERT THE 'GOTO' LINE NUMBER TO BINARY RION DECIMAL
                               9346 *
    1B4B C0 87 19F2            9347 BKC130 B     B$ZDBN                   LINK TO CONVERT LINE NUMBER
                               9348 *
                               9349 * GENERATE A COMPARE CHARACTER PMC IN VIRTUAL MEMORY
                               9350 *
    1B4F D2 02 86              9351 BKC140 LA    BKCCMC(,@BR),@XR         LOAD CADDR OF 'CMC' INSTR
    1B52 34 02 0A40            9352        ST    B$PCAD,@XR               SET PUT RTN FOR VADDR OF 'CMC'
    1B56 3C 00 0A41            9353        MVI   B$PNBY,B@LCMC-1          SET PUT RTN FOR LENGTH OF 'CMC'
    1B5A C0 87 093A            9354        B     B$PUTC                   LINK TO GENERATE PMC
                               9355 *
                               9356 * GENERATE BRANCH ON CONDITION INSTRUCTION IMAGE IN VIRTUAL MEMORY
                               9357 *
    1B5E D2 02 87              9358 BKC150 LA    BKCBRC(,@BR),@XR         LOAD CADDR OF 'BRC' INSTR
    1B61 34 02 0A40            9359        ST    B$PCAD,@XR               SET PUT RTN FOR VADDR OF 'BRC'
    1B65 3C 03 0A41            9360        MVI   B$PNBY,B@LBRC-1          SET PUT RTN FOR LENGTH OF 'BRC'
    1B69 C0 87 093A            9361        B     B$PUTC                   LINK TO GENERATE 'BRC' INSTR
                               9362 *
                               9363 * ESTABLISH ADDRESS AND LINE NUMBER PARAMETERS FOR BRANCH TABLE
                               9364 * RESOLUTION ROUTINE
                               9365 *
    1B6D 0C 01 19EF 0A43       9366 BKC160 MVC   B$BRVA,B$PVAD(@VADDR)    SET ADDR PARAMETER
    1B73 1F 01 19EF 8C         9367        SLC   B$BRVA,BKCLNG(@VADDR,@BR)  SET PARAMETER FOR VADDR OF BRC
    1B78 0C 01 19F1 1A6A       9368        MVC   B$BRLN,B$BINO(B@LCLN)    SET LINE NO PARAMETER
    1B7E C0 87 1996            9369        B     B$BTAB                   LINK TO SET RESOLUTION COND
                               9370 *
                               9371 * RETURN CONTROL TO THE DISTRIBUTOR
                               9372 *
    1B82 C0 87 0700            9373        B     B$DIST                   RETURN TO DISTRIBUTOR

                               9375 ***********************************************************************
                               9376 * CHARACTER IF ROUTINE PMC AND STORAGE PARAMETERS
                               9377 ***********************************************************************
                               9378 *
    1B86 42               1B86 9379 BKCCMC DC    AL(B@LCOP)(B@CCMC)       COMPARE CHAR OPCODE
                               9380 *
    1B87 44               1B87 9381 BKCBRC DC    AL(B@LCOP)(B@CBRC)       BRANCH ON CONDITION OPCODE
    1B88 0000             1B89 9382 BKCBO1 DC    XL(B@LCVA)'00'           BRANCH ON CORD VADDR OPERAND
    1B8A                  1B8A 9383 BKCBO2 DS    CL(B@LCCC)               BRANCH ON COND COND CODE OPND

                               9385 ***********************************************************************
                               9386 * CHARACTER IF ROUTINE CONSTANTS
                               9387 ***********************************************************************
                               9388 *
    1B8B 0002             1B8C 9389 BKCLNG DC    AL(@VADDR)(B@LCCC+1)     LENGTH OF CONDITION CODE + 1

                               9391 ***********************************************************************
                               9392 * RELATIONAL OPERATOR - CONDITION CODE TABLE
                               9393 ***********************************************************************
                               9394 *
                          1B8D 9395 BKCTAB EQU   *                        START OF CODE TABLE
                          0000 9396 BKCOD1 EQU   0                        DISP FOR TABLE OPERATOR
                          0001 9397 BKCCD2 EQU   1                        DISP FOR TABLE COND CODE
                          0002 9398 BKCLTH EQU   2                        LENGTH OF TABLE ENTRY
                          1B8B 9399 BKCOTB EQU   BKCTAB-BKCLTH            CODE TABLE BASE ADDRESS
```

```
                               9400 *
    1B8D 7E                1B8D 9401       DC    AL1(B@EQUL)               RELATIONAL OPERATOR '='
    1B8E 84                1B8E 9402       DC    AL1(B@BREQ)               BRANCH CONDITION - EQUAL
                               9403 *
    1B8F 6E                1B8F 9404       DC    AL1(B@GRTR)               RELATIONAL OPERATOR '>'
    1B90 88                1B90 9405       DC    AL1(B@BRHI)               BRANCH CONDITION - HI
                               9406 *
    1B91 4C                1B91 9407       DC    AL1(B@LESS)               RELATIONAL OPERATOR '<'
    1B92 82                1B92 9408       DC    AL1(B@BRLO)               BRANCH CONDITION - LOW
                               9409 *
    1B93 BA                1B93 9410       DC    AL1(B@LESS+B@GRTR)        RELATIONAL OPERATOR '<>'
    1B94 94                1B94 9411       DC    AL1(B@BRNE)               BRANCH CONDITION - NOT EQUAL
                               9412 *
    1B95 CA                1B95 9413       DC    AL1(B@LESS+B@EQUL)        RELATIONAL OPERATOR '<='
    1B96 98                1B96 9414       DC    AL1(B@BRNH)               BRANCH CONDITION - NOT HIGH
                               9415 *
    1B97 EC                1B97 9416       DC    AL1(B@GRTR+B@EQUL)        RELATIONAL OPERATOR '>='
    1B98 92                1B98 9417       DC    AL1(B@BRNL)               BRANCH CONDITION - NOT LOW
                               9418 *
    1B99 7F                1B99 9419       DC    AL1(B@NEQL)               RELATIONAL OPERATOR ' '
    1B9A 94                1B9A 9420       DC    AL1(B@BRNE)               BRANCH CONDITION - NOT EQUAL
                               9421 *
                               9422 ***********************************************************************
                               9423 *
                               9424 * END OF 'CHAR IF' ROUTINE CODING
                               9425 *
```

```
9427 ***********************************************************************
9428 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
9429 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
9430 *                                                                     *
9431 ***********************************************************************
9432 *STATUS                                                              *
9433 *   VERSION 1 MODIFICATION 0                                          *
9434 *                                                                     *
9435 *FUNCTION                                                            *
9436 *   BMPUTX IS EXECUTED TO TRANSLATE MAT PUT STATEMENTS AS THEY OCCUR  *
9437 *   IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE   *
9438 *   THE PSEUDOCODE IN VIRTUAL MEMORY.                                 *
9439 *                                                                     *
9440 *ENTRY POINTS                                                        *
9441 *   BMPUTX HAS ONLY CNE ENTRY POINT:                                  *
9442 *        BMPUTX - TRANSLATE MAT PUT STATEMENT                         *
9443 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
9444 *        B     BMPUTX                                                 *
9445 *                                                                     *
9446 *INPUT                                                               *
9447 *  * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING   *
9448 *    THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE        *
9449 *    LEADING KEYWORD. MAT PUT.                                        *
9450 *  * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
9451 *    CHARACTER IN THE LEADING KEYWORD. MAT PUT.                       *
9452 *                                                                     *
9453 *OUTPUT                                                              *
9454 *  * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE         *
9455 *    GENERATED BY BMPUTX IS STORED IN THE NEXT AVAILABLE VIRTUAL      *
9456 *    MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION          *
9457 *    SEQUENCES.                                                       *
9458 *  * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE        *
9459 *    CHARACTER WHICH TERMINATES THE STATEMENT.                        *
9460 *                                                                     *
9461 *EXTERNAL REFERENCES                                                 *
9462 *        B$GETU - (B$NUNC) - ENTRY TO BASIC RETRIEVAL ROUTINE.        *
9463 *        B$PUTC - (B$PCAD, B$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY *
9464 *                 ROUTINE.                                            *
9465 *        B$MATR - ENTRY TO BASIC COMPILER MATRIX REFERENCE ROUTINE    *
9466 *        B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                *
9467 *                                                                     *
9468 *EXITS, NORMAL                                                       *
9469 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
9470 *                                                                     *
9471 *EXITS, ERROR                                                        *
9472 *   N/A                                                               *
9473 *                                                                     *
9474 *TABLES/WORK AREAS                                                   *
9475 *   N/A                                                               *
9476 *                                                                     *
9477 *ATTRIBUTES                                                          *
9478 *   BMPUTX IS NATURALLY RELOCATABLE AND REUSABLE.                     *
9479 *                                                                     *
9480 *CHARACTER CODE DEPENDENCY                                           *
9481 *   THE OPERATION OF THIS NODULE DOES NOT DEPEND ON A PARTICULAR      *
9482 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
```

```
                               9483 *                                                                     *
                               9484 *NOTES                                                                *
                               9485 *   ERROR PROCEDURES                                                  *
                               9486 *      N/A                                                            *
                               9487 *                                                                     *
                               9488 *   REGISTER USAGE                                                    *
                               9489 *      BOTH THE INNS AND BASE REGISTERS ARE USED DURING EXECUTION.    *
                               9490 *                                                                     *
                               9491 *   SAVED/RESTORED AREAS                                              *
                               9492 *      N/A                                                            *
                               9493 *                                                                     *
                               9494 *   MODIFICATION CONSIDERATIONS                                       *
                               9495 *      BMPUTX RESIDES ON A SECTOR WITH IKCRIF. ANY MODIFICATION    1-4*
                               9496 *      TO BMPUTX SHOULD CONSIDER THIS CO-RESIDENCY AND TAKE INTO   1-4*
                               9497 *      CONSIDERATION THE LIMITATION OF THE SECTOR BOUNDARY ON SIZE.1-4*
                               9498 *                                                                     *
                               9499 *   REQUIRED MODULES                                                  *
                               9500 *        @SYSEQ - COMMON SYSTEM EQUATES.                              *
                               9501 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.   *
                               9502 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.      *
                               9503 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               9504 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               9505 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               9506 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               9507 *        $B$EQU - COMPILER FIXED EQUATES.                             *
                               9508 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               9509 *                                                                     *
                               9510 *   OTHER                                                             *
                               9511 *      BMPUTX IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               9512 ***********************************************************************

                               9514 *
                               9515 * ENTER BMPUTX - MAT PUT STATEMENT ROUTINE
                               9516 *
                          1B9B 9517 BMPUTX EQU   *                         BMPUTX ENTRY POINT
                               9518 *
                               9519 * SET GET ROUTINE TO SKIP TO THE CHARACTER FOLLOWING KEYWORDS 'MAT PUT'
                               9520 *
    1B9B 3C 05 0873            9521 BMP010 MVI   B$NUMC,B@LMPT-1           SET GET TO SKIP KEYWORD
    1B9F C0 87 0867            9522        B     B$GETC                    LINK TO ADVANCE POINTER
    1BA3 C0 87 14B0            9523        B     B$CSCN                    LINK TO PROCESS FILE REFERENCE
                               9524 *
                               9525 * GENERATE THE 'ADF' PMC IN VIRT. MEM. (IF OPERAND IS ZERO, THE FILE
                               9526 * IS NOT IN ENTRY TABLE)
                               9527 *
    1BA7 D2 02 E2              9528 BMP100 LA    BMPAFC(,@BR),@XR          LOAD CADDR OF 'ADF' INSTR
    1BAA 34 02 0A40            9529        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR AVE
    1BAE 3C 01 0A41            9530        MVI   B$PNBY,B@LADF-1           SET LNG PARM OF PUT FOR 'ADF'
    1BB2 C0 87 093A            9531        B     B$PUTC                    LINK TO GENERATE 'ADF' INSTR
                               9532 *
                               9533 * CALL GET ROUTINE TO GET NEXT CHAR
                               9534 *
    1BB6 3C 00 0873            9535 BMP110 MVI   B$NUMC,B@GETS             DISABLE GET ROUTINE
    1BBA C0 87 0867            9536        B     B$GETC                    LINK TO GET CHARACTER POINTER
                               9537 *
                               9538 * CALL MATRIX REFERENCE PROCESSOR TO GENERATE DOPE VECTOR STACKING
                               9539 * INSTRUCTIONS
                               9540 *
    1BBE C0 87 18F3            9541 BMP120 B     B$MATR                    LINK TO PROCESS MAT-REFERENCE
    1BC2 74 02 D7              9542        ST    BMP140+@OP1(,@BR),@XR     SAVE TEXT POINTER
                               9543 *
                               9544 * GENERATE THE 'MF1' INSTR IN VIRTAL MEMORY.
                               9545 *
    1BC5 D2 02 E4              9546 BMP130 LA    BMPMFC(,@BR),@XR          LOAD CADDR OF 'MF1' INSTR
    1BC8 34 02 0A40            9547        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'MF1'
    1BCC 3C 02 0A41            9548        MVI   B$PNBY,B@LMF1-1           SET LNG PARM OF PUT FOR 'MF1'
    1BD0 C0 87 093A            9549        B     B$PUTC                    LINK TO GENERATE 'MF1' INSTR
                               9550 *
                               9551 * TEST THE DELIMITER FOR BEING A STATEMENT TERMINATOR
                               9552 *
    1BD4 C2 02 0000            9553 BMP140 LA    *-*,@XR                   RESTORE TEXT POINTER
    1BD8 BD 1E 00              9554        CLI   B@CHAR(,@XR),B@EOST       IF DELIMITER IS NOT EOS
    1BDB D0 01 BE              9555        BNE   BMP120(,@BR)              * GO PROCESS NEXT MAT-REFERENCE
                               9556 *
                               9557 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               9558 *
    1BDE C0 87 0700            9559 BMP150 B     B$DIST                    RETURN TO DISTRIBUTER

                               9561 ***********************************************************************
                               9562 * MAT PUT STATEMENT ROUTINE PARAMETER AND STORAGE AREAS
                               9563 ***********************************************************************
                               9564 *
    1BE2 58               1BE2 9565 BMPAFC DC    AL(B@LCOP)(B@CADF)        'ADF' INSTR OPCODE
    1BE3 01               1BE3 9566 BMPAFO DC    XL1'01'                   'ADF' INSTR OPERAND
                               9567 *
    1BE4 18               1BE4 9568 BMPMFC DC    AL(B@LCOP)(B@CMF1)        'MF1' INSTR OPCODE
    1BE5 3E0C             1BE6 9569 BMPMFO DC    AL(B@LCVA)(V$XMPT)        'MF1' INSTR OPND - PUT

                               9571 ***********************************************************************
                               9572 * MAT PUT STATEMENT CONSTANTS AND EQUATES
                               9573 ***********************************************************************
                               9574 *
                          1BE7 9575 BMPSFA EQU   *
                               9576 *
    1BE7 0001             1BE8 9577 BMPBN1 DC    IL(@CADDR)'1'             BINARY 1
                               9578 *
                               9579 ***********************************************************************
                               9580 *
                               9581 * END OF 'MAT PUT' STATEMENT ROUTINE CODING
                               9582 *
```

```
9584 ***********************************************************************
9585 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
9586 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
9587 *                                                                     *
9588 ***********************************************************************
9589 *STATUS                                                               *
9590 *   VERSION 1 MODIFICATION 0                                          *
9591 *                                                                     *
9592 *FUNCTION                                                             *
9593 *   BKMGTO IS EXECUTED TO TRANSLATE MULTIPLE GOTO STATEMENTS AS THEY  *
9594 *   OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO   *
9595 *   PLACE THE PSEUDOCODE INTO VIRTUAL MEMORY.                         *
9596 *                                                                     *
9597 *ENTRY POINTS                                                         *
9598 *   BKMGTO HAS ONLY ONE ENTRY POINT:                                  *
9599 *        BKMGTO - TRANSLATE MULTIPLE GOTO STATEMENT                   *
9600 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
9601 *        B     BKMGTO                                                 *
9602 *                                                                     *
9603 *INPUT                                                                *
9604 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
9605 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN TIE       *
9606 *     LEADING KEYWORD, GOTO.                                          *
9607 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE 1ST   *
9608 *     CHARACTER IN THE LEADING KEYWORD, GOTO.                         *
9609 *                                                                     *
9610 *OUTPUT                                                               *
9611 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
9612 *     GENERATED BY BKMGTO IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
9613 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
9614 *     SEQUENCES.                                                      *
9615 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
9616 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
9617 *   * B$BRVA - CONTAINS THE VIRTUAL ADDRESS OF THE RIGHT BYTE OF      *
9618 *     THE ADDRESS OPERAND FIELD IN THE EXCEPTION BYPASS ADDRESS       *
9619 *     STACKING INSTRUCTION.                                           *
9620 *   * B$NXSW - SET TO ON STATUS TO CAUSE RESOLUTION OF THE EXCEPTION  *
9621 *     BYPASS ADDRESS STACKING INSTRUCTION OPERAND.                    *
9622 *                                                                     *
9623 *EXTERNAL REFERENCES                                                  *
9624 *        B$GETC - (B$NUMC, B$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.*
9625 *        B$PUTC - (B$PCAD, B$PNBY, B$PVAD) - ENTRY TO COMPILER VIRT   *
9626 *                 MEMORY OUTPUT ROUTINE.                              *
9627 *        B$SCAN - ENTRY TO BASIC ARITHMETIC EXPRESSION SCAN ROUTINE.  *
9628 *        B$BTAB - (B$BRVA, B$BRLN) - ENTRY TO BASIC COMPILER BRANCH   *
9629 *                 TABLE ROUTINE.                                      *
9630 *        B$ZDBN - (B$BINO) - ENTRY TO BASIC COMPILER ZONED DECIMAL TO *
9631 *                 BINARY CONVERSION ROUTINE.                          *
9632 *        B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR                 *
9633 *                                                                     *
9634 *EXITS, NORMAL                                                        *
9635 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR                      *
9636 *                                                                     *
9637 *EXITS, ERROR                                                         *
9638 *   N/A                                                               *
9639 *                                                                     *
```

```
                               9640 *TABLES/WORK AREAS                                                    *
                               9641 *   N/A                                                               *
                               9642 *                                                                     *
                               9643 *ATTRIBUTES                                                           *
                               9644 *   BKMGTO IS NATURALLY RELOCATABLE AND REUSABLE                      *
                               9645 *                                                                     *
                               9646 *CHARACTER CODE DEPENDENCY                                            *
                               9647 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               9648 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               9649 *                                                                     *
                               9650 *NOTES                                                                *
                               9651 *   ERROR PROCEDURES                                                  *
                               9652 *      N/A                                                            *
                               9653 *                                                                     *
                               9654 *   REGISTER USAGE                                                    *
                               9655 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               9656 *                                                                     *
                               9657 *   SAVED/RESTORED AREAS                                              *
                               9658 *      N/A                                                            *
                               9659 *                                                                     *
                               9660 *   MODIFICATION CONSIDERATIONS                                       *
                               9661 *      BKMGTO RESIDES ON THE SAME SECTOR WITH BXRSET AND BTPAUS.   1-4*
                               9662 *      AND MODIFICATION TO BKMGTO SHOULD TAKE INTO CONSIDERATION   1-4*
                               9663 *      THIS CO-RESIDENCY SINCE IT WILL CHANGE THE ENTRY ADDRESSES  1-4*
                               9664 *      OF BXRSET AND BTPAUS AND MUST TAKE INTO CONSIDERATION THE   1-4*
                               9665 *      LIMITATION OF THE SECTOR BOUNDARY ON SIZE.                  1-4*
                               9666 *                                                                     *
                               9667 *   REQUIRED MODULES                                                  *
                               9668 *        @SYSEQ - COMMON SYSTEM EQUATES                               *
                               9669 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES    *
                               9670 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS               *
                               9671 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES                    *
                               9672 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES                         *
                               9673 *        @ERMEQ - ERROR MESSAGE EQUATES                               *
                               9674 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES                       *
                               9675 *        $B$EQU - COMPILER FIXED EQUATES                              *
                               9676 *        $B@EQU - COMPILER SYSTEM EQUATES                             *
                               9677 *                                                                     *
                               9678 *   OTHER                                                             *
                               9679 *      BKMGTO IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               9680 ***********************************************************************
    1C00                       9681        ORG   *,256,0                   BEGIN AT CORE PAGE BOUNDARY  1-4
                          1C00 9682        USING *,@BR                     DEFINE BASE ADDR FOR CORE PG 1-4
                               9683 *
                               9684 * ENTER BKMGTO - MULTIPLE 'GOTO' STATEMENT ROUTINE
                               9685 *
                          1C00 9686 BKMGTO EQU   *                         BKMGTO ENTRY POINT
                               9687 *
                               9688 * SET INPUT PARAMETER TO SKIP KEYWORD 'GOTO'.
                               9689 *
    1C00 3C 04 0873            9690 BKM010 MVI   B$NUMC,B@LGTO             SET GET RTN TO SKIP 'GOTO'
    1C04 C0 87 0867            9691        B     B$GETC                    LINK TO ADVANCE POINTER
                               9692 *
                               9693 * GENERATE AN 'STA' INSTRUCTION IMAGE PMC IN VIRTUAL MEMORY
                               9694 *
    1C08 D2 02 9C              9695 BKM020 LA    BKMSTC(,@BR),@XR          LOAD CADDR OF 'STA' INSTR
    1C0B 34 02 0A40            9696        ST    B$PCAD,@XR                SET PUT RTN FOR VADDR OF 'STA'
    1C0F 3C 02 0A41            9697        MVI   B$PNBY,B@LSTA-1           SET PUT RTN FOR LENGTH OF 'STA'
    1C13 C0 87 093A            9698        B     B$PUTC                    LINK TO GENERATE PMC
                               9699 *
                               9700 * SAVE THE VADDS FOLLOWING THE OPERAND OF THE 'STA' PMC
                               9701 *
    1C17 4C 01 A5 0A43         9702 BKM030 MVC   BKMVAD(,@BR),B$PVAD(@VADDR)  SAVE VADDR TO RESOLVE 'STA'
                               9703 *
                               9704 * CONVERT A LIST LINE NUMBER TO BINARY FROM DECIMAL
                               9705 *
    1C1C 35 02 0878            9706 BKM035 L     B$GPTR,@XR                RESTORE TEXT POINTER
    1C20 7C 00 A1              9707        MVI   BKMCSO(,@BR),@ZERO        INITLZ LINE NO. COUNT TO ZERO
    1C23 C0 87 19F2            9708 BKM040 B     B$ZDBN                    CONVERT LIST LN NO TO BINARY
                               9709 *
                               9710 * GENERATE AN 'STA' INSTRUCTION PMC IN VIRTUAL MEMORY
                               9711 *
    1C27 D2 02 9C              9712 BKM050 LA    BKMSTC(,@BR),@XR          LOAD CADDR OF 'STA' INSTR
    1C2A 34 02 0A40            9713        ST    B$PCAD,@XR                SET PUT RTN FOR VADDR OF 'STA'
    1C2E 3C 02 0A41            9714        MVI   B$PNBY,B@LSTA-1           SET PUT RTN FOR LENGTH OF 'STA'
    1C32 C0 87 093A            9715        B     B$PUTC                    LINK TO GENERATE 'STA' PMC
                               9716 *
                               9717 * ESTABLISH THE CURRENT 'STA' OPERAND FOR ADDRESS RESOLUTION
                               9718 *
    1C36 0C 01 19EF 0A43       9719 BKM060 MVC   B$BRVA,B$PVAD(@VADDR)     SET VADDR PARAMETER FOR BR TBL
    1C3C 1F 01 19EF A3         9720        SLC   B$BRVA,BKMBN1(@VADDR,@BR)  ADJUST VADDR TO 'STA' OPND
                               9721 *
                               9722 * ESTABLISH THE LIST LINE NUMBER AS THE RESOLUTION LINE NUMBER
                               9723 *
    1C41 0C 01 19F1 1A6A       9724 BKM070 MVC   B$BRLN,B$BINO(@VADDR)     SET LN NO PARAMETER FOR BR TBL
    1C47 C0 87 1996            9725        B     B$BTAB                    LINK TO RESOLVE *STA' OPND
                               9726 *
                               9727 * INCREMENT CURRENT LIST LINE NUMBER COUNT BY ONE
                               9728 *
    1C4B 5E 01 A1 A3           9729 BKM080 ALC   BKMCSO(,@BR),BKMBN1(@VADDR,@BR)  INCREMENT LK NO COUNT
                               9730 *
                               9731 * CHECK FOR THE END OF THE LINE NUMBER LIST
                               9732 *
    1C4F 35 02 0878            9733 BKM090 L     B$GPTR,@XR                RESTORE TEXT POINTER
    1C53 BD 6B 00              9734        CLI   B@CHAR(,@XR),B@CMMA       IF LINE NUMBER LIST AT END
    1C56 F2 01 07              9735        JNE   BKM100                    * JUMP TO PROCESS ARITH EXPR
    1C59 C0 87 0867            9736        B     B$GETC                    LINK TO GET NEXT CHAR
    1C5D D0 87 23              9737        B     BKM040(,@BR)              BRANCH TO PROCESS NEXT LN NO
                               9738 *
                               9739 * SET INPUT PARAMETER TO SKIP TO 'N' IN KEYWORD 'ON'
                               9740 *
    1C60 3C 01 0873            9741 BKM100 MVI   B$NUMC,B@LKON-1           SET GET RTN TO SKIP 'O' IN 'ON'
    1C64 C0 87 0867            9742        B     B$GETC                    LINK TO ADVANCE POINTER
                               9743 *
                               9744 * CALL ARITH SCAN RTN TO GENERATE PMC FOR ARITH EXPRESSION
                               9745 *
    1C68 C0 87 1514            9746 BKM110 B     B$SCAN                    LINK TO SCAN ARITH EXPRESSION
                               9747 *
                               9748 * GENERATE A 'CSA' INSTRUCTION WITH LIST LINE NO COUNT AS OPERAND
                               9749 *
    1C6C D2 02 A0              9750 BKM120 LA    BKMCSC(,@BR),@XR          LOAD CADDR OF 'CSA' INSTR
    1C6F 34 02 0A40            9751        ST    B$PCAD,@XR                SET PUT RTN FOR VADDR OF 'CSA'
    1C73 3C 01 0A41            9752        MVI   B$PNBY,B@LCSA-1           SET PUT RTN FOR LENGTH OF 'CSA'
    1C77 C0 87 093A            9753        B     B$PUTC                    LINK TO GENERATE 'CSA' PMC
                               9754 *
                               9755 * GENERATE A 'BRS' INSTRUCTION IN VIRTUAL MEMORY
                               9756 *
    1C7B D2 02 9F              9757 BKM125 LA    BKMBRC(,@BR),@XR          LOAD CADDR OF 'BRS' INSTR
    1C7E 34 02 0A40            9758        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR BRS
    1C82 3C 00 0A41            9759        MVI   B$PNBY,B@LBRS-1           SET LNG PARM OF PUT FOR 'BRS'
    1C86 C0 87 093A            9760        B     B$PUTC                    LINK TO GENERATE 'BRS' INSTR
                               9761 *
                               9762 * ESTABLISH THE VADDR OF THE FIRST 'STA' INSTR AS THE BRANCH ADDRESS
                               9763 * TABLE RESOLUTION ADDRESS
                               9764 *
    1C8A 1C 01 19EF A5         9765 BKM130 MVC   B$BRVA,BKMVAD(@VADDR,@BR)  SET VADDR PARAMETER FOR BR TBL
    1C8F 1F 01 19EF A3         9766        SLC   B$BRVA,BKMBN1(@VADDR,@BR)  ADJUST VADOR FOR 'STA' OPERAND
                               9767 *
                               9768 * SET 'NEXT' SW FOR RESOLUTION OF 'STA' OPERAND WITH NEXT IN NO
                               9769 *
    1C94 3A 07 071D            9770 BKM140 SBN   B$NXSW,B$NXMK             SET 'NEXT' SW TO RESOLVE LN NO
                               9771 *
                               9772 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               9773 *
    1C98 C0 87 0700            9774 BKM150 B     B$DIST                    RETURN TO DISTRIBUTOR

                               9776 ***********************************************************************
                               9777 * MULTIPLE 'GOTO' STATEMENT ROUTINE PMC STORAGE AND PARAMETERS
                               9778 ***********************************************************************
                               9779 *
    1C9C 34               1C9C 9780 BKMSTC DC    AL(B@LCOP)(B@CSTA)        'STA' INSTR IMAGE OPCODE
    1C9D 0000             1C9E 9781 BKMSTO DC    XL(B@LCVA)'00'            'STA' INSTR OPERAND IMAGE
                               9782 *
    1C9F 4C               1C9F 9783 BKMBRC DC    AL(B@LCOP)(B@CBRS)        'BRS' INSTR OPCODE
                               9784 *
    1CA0 3E               1CA0 9785 BKMCSC DC    AL(B@LCOP)(B@CCSA)        'CSA' INSTR OPCODE
    1CA1                  1CA1 9786 BKMCSO DS    CL(B@LCNN)                'CSA' OPND - LIST LN NO COUNT

                               9788 ***********************************************************************
                               9789 * MULTIPLE 'GOTO' STATEMENT ROUTINE CONSTANTS
                               9790 ***********************************************************************
                               9791 *
    1CA2 0001             1CA3 9792 BKMBN1 DC    IL(B@LCVA)'1'             BINARY 1
    1CA4                  1CA5 9793 BKMVAD DS    CL(@VADDR)                VADDR FOLLOWING 'STA' OPERAND

                               9795 ***********************************************************************
                               9796 *
                               9797 * END OF MULTIPLE 'GOTO' STATEMENT ROUTINE CODING
                               9798 *
                               9800 ***********************************************************************
                               9801 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               9802 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               9803 *                                                                     *
                               9804 ***********************************************************************
                               9805 *STATUS                                                               *
                               9806 *   VERSION 1 MODIFICATION 0                                          *
                               9807 *                                                                     *
                               9808 *FUNCTION                                                             *
                               9809 *   BXRSET IS EXECUTED TO TRANSLATE RESET STATEMENTS AS THEY OCCUR    *
                               9810 *   IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE   *
                               9811 *   THE PSEUDOCODE INTO VIRTUAL MEMORY.                               *
                               9812 *                                                                     *
                               9813 *ENTRY POINTS                                                         *
                               9814 *   BXRSET HAS ONLY ONE ENTRY POINT:                                  *
                               9815 *      BXRSET - TRANSLATE RESET STATEMENT                             *
                               9816 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               9817 *      B       BXRSET                                                 *
                               9818 *                                                                     *
                               9819 *INPUT                                                                *
                               9820 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               9821 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               9822 *     LEADING KEYWORD, RESET.                                         *
                               9823 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                               9824 *     CHARACTER IN THE LEADING KEYWORD. RESET.                        *
                               9825 *                                                                     *
                               9826 *OUTPUT                                                               *
                               9827 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               9828 *     GENERATED BY BXRSET IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               9829 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               9830 *     SEQUENCES.                                                      *
                               9831 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               9832 *   * CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               9833 *                                                                     *
                               9834 *EXTERNAL REFERENCES                                                  *
                               9835 *        B$GETC - (B$NUMC) - ENTRY TO BASIC TEXT RETRIEVAL ROUTINE.   *
                               9836 *        B$PUTC - (B$PCAD) - B$PNBY) - ENTRY TO COMPILER VIRT MEMORY  *
                               9837 *                 OUTPUT ROUTINE.                                     *
                               9838 *        B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                *
                               9839 *                                                                     *
                               9840 *EXITS, NORMAL                                                        *
                               9841 *   B$DIST - ENTRY TO THE BASIC COMPILER DISTRIBUTOR                  *
                               9842 *                                                                     *
                               9843 *EXITS, ERROR                                                         *
                               9844 *   N/A                                                               *
                               9845 *                                                                     *
                               9846 *TABLES/WORK AREAS                                                    *
                               9847 *   N/A                                                               *
                               9848 *                                                                     *
                               9849 *ATTRIBUTES                                                           *
                               9850 *   * BXRSET IS NATURALLY RELOCATABLE AND REUSABLE.                   *
                               9851 *                                                                     *
                               9852 *CHARACTER CODE DEPENDENCY                                            *
                               9853 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               9854 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               9855 *                                                                     *
                               9856 *NOTES                                                                *
                               9857 *   ERROR PROCEDURES                                                  *
                               9858 *      N/A                                                            *
                               9859 *                                                                     *
                               9860 *   REGISTER USAGE                                                    *
                               9861 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               9862 *                                                                     *
                               9863 *   SAVED/RESTORED AREAS                                              *
                               9864 *      N/A                                                            *
                               9865 *                                                                     *
                               9866 *   MODIFICATION CONSIDERATIONS                                       *
                               9867 *      BXRSET RESIDES ON THE SAME SECTOR WITH BKMGTO AND BTPAUS.   1-4*
                               9868 *      ANY MODIFICATION TO BXRSET MUST CONSTER THIS CO-RESIDENCY   1-4*
                               9869 *      SINCE WILL CHANGE THE ENTRY ADDRESS OF BTPAUS.  THE         1-4*
                               9870 *      LIMITATION OF THE SECTOR BOUNDARY ON SIZE MUST ALSO BE      1-4*
                               9871 *      CONSIDERID.                                                 1-4*
                               9872 *                                                                     *
                               9873 *   REQUIRED MODULES                                                  *
                               9874 *        @SYSEQ - COMMON SYSTEM EQUATES                               *
                               9875 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES    *
                               9876 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS               *
                               9877 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES                    *
                               9878 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES                         *
                               9879 *        @ERMEQ - ERROR MESSAGE EQUATES                               *
                               9880 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES                       *
                               9881 *        $B$EQU - COMPILER FIXED EQUATES                              *
                               9882 *        $B@EQU - COMPILER SYSTEM EQUATES                             *
                               9883 *                                                                     *
                               9884 *   OTHER                                                             *
                               9885 *      BXRSET IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               9886 ***********************************************************************

                               9888 *
                               9889 * ENTER BXRSET - 'RESET' STATEMENT ROUTINE
                               9890 *
                          1CA6 9891 BXRSET EQU    *                        BXRSET ENTRY POINT
                               9892 *
                               9893 * SET POINTER TO SKIP TO 'T' IN KEYWORD 'RESET'
                               9894 *
    1CA6 3C 04 0873            9895 BXR010 MVI    B$NUMC,B@LKRT-1           SET GET RTN TO SKIP TO 'T'
    1CAA C0 87 0867            9896        B      B$GETC                   LINK TO ADVANCE POINTER
    1CAE C0 87 14B0            9897 BXR020 B      B$CSCN                   LINK TO PROCESS FILE REFERENCE
                               9898 *
                               9899 * GENERATE THE 'ADF' PMC IN V.M. IF OPERAND IS NOT ZERO
                               9900 *
    1CB2 D2 02 E2              9901 BXR110 LA     BXRAFC(,@BR),@XR         LOAD CADDR OF 'ADF' INSTR
    1CB5 34 02 0A40            9902        ST     B$PCAD,@XR               SET VADDR PARM OF PUT FOR ADF
    1CB9 3C 01 0A41            9903        MVI    B$PNBY,B@LADF-1          SET LNG PARM OF PUT FOR 'ADP'
    1CBD C0 87 093A            9904        B      B$PUTC                   LINK TO GENERATE 'ADF' PMC
                               9905 *
                               9906 * GENERATE THE 'RST' PMC IN V.M.
                               9907 *
    1CC1 D2 02 E4              9908 BXR120 LA     BXRRTC(,@BR),@XR         LOAD CADDR OF 'RST' INSTR
    1CC4 34 02 0A40            9909        ST     B$PCAD,@XR               SET VADDR PARM OF PUT FOR RST
    1CC8 3C 00 0A41            9910        MVI    B$PNBY,B@LRST-1          SET LNG PARM OF PUT FOR 'RST'
    1CCC C0 87 093A            9911        B      B$PUTC                   LINK TO GENERATE 'RST' PMC
```

```
                               9912 *
                               9913 * TEST NEXT LIST CHARACTER FOR BEING AN END-OF-STATEMENT
                               9914 *
    1CD0 3C 00 0873            9915 BXR130 MVI    B$NUMC,B@GETS            DISABLE GET ROUTINE
    1CD4 C0 87 0867            9916        B      B$GETC                   LINK TO GET CHARACTER POINTER
    1CD8 BD 1E 00              9917        CLI    B@CHAR(,@XR),B@EOST      IF CHAR IS EOS
    1CDB D0 01 AE              9918        BNE    BXR020(,@BR)             * BRANCH TO PROCESS FILENAME
                               9919 *
                               9920 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                               9921 *
    1CDE C0 87 0700            9922 BXR140 B      B$DIST                   RETURN TO DISTRIBUTOR
                               9923 *
                               9924 ***********************************************************************
                               9925 * 'RESET' STATEMENT PARAMETER AND STORAGE AREAS
                               9926 ***********************************************************************
                               9927 *
    1CE2 58               1CE2 9928 BXRAFC DC    AL(B@LCOP)(B@CADF)        'ADF' INSTR OPCODE
    1CE3 00               1CE3 9929 BXRAFO DC    XL1'00'                   'ADF' INSTR OPERAND
                               9930 *
    1CE4 5C               1CE4 9931 BXRRTC DC    AL(B@LCOP)(B@CRST)        'RST' INSTR OPCODE

                               9933 ***********************************************************************
                               9934 * 'RESET' STATEMENT CONSTANTS AND EQUATES
                               9935 ***********************************************************************
                               9936 *
                               9937 * CONSTANTS
                               9938 *
                          1CE5 9939 BXRSFA EQU   *
                               9940 *
    1CE5 0001             1CE6 9941 BXRBNI DC    IL(@CADDR)'1'              BINARY +1
                               9942 *
                               9943 ***********************************************************************
                               9944 *
                               9945 * END OF 'RESET' STATEMENT ROUTINE CODING
                               9946 *
                               9948 ***********************************************************************
                               9949 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               9950 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               9951 *                                                                     *
                               9952 ***********************************************************************
                               9953 *STATUS                                                               *
                               9954 *   VERSION 1 MODIFICATION 0                                          *
                               9955 *                                                                     *
                               9956 *FUNCTION                                                             *
                               9957 *   BTPAUS IS EXECUTED TO TRANSLATE PAUSE STATEMENTS AS THEY OCCUR IN *
                               9958 *   A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE  *
                               9959 *   PSEUDOCODE IN VIRTUAL MEMORY.                                     *
                               9960 *                                                                     *
                               9961 *ENTRY POINTS                                                         *
                               9962 *   BTPAUS HAS ONLY ONE ENTRY POINT:                                  *
                               9963 *        BTPAUS - TRANSLATE PAUSE STATEMENT                           *
                               9964 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               9965 *        B     BTPAUS                                                 *
                               9966 *                                                                     *
                               9967 *INPUT                                                                *
                               9968 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               9969 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                               9970 *     LEADING KEYWORD, PAUSE.                                         *
                               9971 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                               9972 *     CHARACTER IN THE LEADING KEYWORD, PAUSE.                        *
                               9973 *                                                                     *
                               9974 *OUTPUT                                                               *
                               9975 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               9976 *     GENERATED BY BTPAUS IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               9977 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               9978 *     SEQUENCES.                                                      *
                               9979 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               9980 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               9981 *                                                                     *
                               9982 *EXTERNAL REFERENCES                                                  *
                               9983 *        B$PUTC(B$PCAD.B$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY     *
                               9984 *          OUTPUT.                                                    *
                               9985 *        B$RMRK - ENTRY TO BASIC COMPILER REMARK ROUTINE.             *
                               9986 *                                                                     *
                               9987 *EXITS, NORMAL                                                        *
                               9988 *   BMW - ENTRY TO BASIC COMPILER REMARK ROUTINE.                     *
                               9989 *                                                                     *
                               9990 *EXITS, ERROR                                                         *
                               9991 *    N/A                                                              *
                               9992 *                                                                     *
                               9993 *TABLES/WORK AREAS                                                    *
                               9994 *    N/A                                                              *
                               9995 *                                                                     *
                               9996 *ATTRIBUTES                                                           *
                               9997 *   BTPAUS IS NATURALLY RELOCATABLE AND REUASBLE.                     *
                               9998 *                                                                     *
                               9999 *CHARACTER CODE DEPENDENCY                                            *
                                    *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                                  1 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                                  2 *                                                                     *
                                  3 *NOTES                                                                *
                                  4 *   ERROR PROCEDURES                                                   *
                                  5 *      N/A                                                             *
                                  6 *                                                                      *
                                  7 *   REGISTER USAGE                                                     *
                                  8 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.    *
                                  9 *                                                                      *
                                 10 *   SAVED/RESTORED AREAS                                               *
                                 11 *      N/A                                                             *
                                 12 *                                                                      *
                                 13 *   MODIFICATION CONSIDERATIONS                                        *
                                 14 *      BTPAUS RESIDES ON THE SAME SECTOR WITH BKMGTO AND BXRSET.   1-4*
                                 15 *      ANY MODIFICATION OF BTPAUS MUST TAKE INTO CONSIDERATION     1-4*
                                 16 *      THIS CO-RESIDENCY AND THE LIMITATION OF THE SECTOR BOUNDARY 1-4*
                                 17 *      ON SIZE.                                                    1-4*
                                 18 *                                                                      *
                                 19 *   REQUIRED MODULES                                                   *
                                 20 *        @SYSEQ - COMMON SYSTEM EQUATES                                *
                                 21 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES     *
                                 22 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS                *
                                 23 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES                     *
                                 24 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES                          *
                                 25 *        @ERMEQ - ERROR MESSAGE EQUATES                                *
                                 26 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES                        *
                                 27 *        $B$EQU - COMPILER FIXED EQUATES                               *
                                 28 *        $B@EQU - COMPILER SYSTEM EQUATES                              *
                                 29 *                                                                      *
                                 30 *   OTHER                                                              *
                                 31 *      BTPAUS IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.       *
                                 32 ************************************************************************

                                 34 *
                                 35 * ENTER BTPAUS - 'PAUSE' STATEMENT ROUTINE
                                 36 *
                          1CE7   37 BTPAUS EQU   *                         BTPAUS ENTRY POINT
                                 38 *
                                 39 * GENERATE A HALT INSTRUCTION IN VIRTUAL MEMORY
                                 40 *
    1CE7 D2 02 FA                41 BTP010 LA    BTPHTC(,@BR),@XR          LOAD CADDR OF 'HLT' INSIR
    1CEA 34 02 0A40              42        ST    B$PCAD,@XR                SET PUT RTN FOR VADDR OF 'HLT'
    1CEE 3C 00 0A41              43        MVI   B$PNBY,B@LHLT-1           SET PUT RTN FOR LENGTH OF 'HLT'
    1CF2 C0 87 093A              44        B     B$PUTC                    LINK TO GENERATE PMC
                                 45 *
                                 46 * RETURN CONTROL TO THE REMARK STATEMENT ROUTINE
                                 47 *
    1CF6 C0 87 1AE6              48 BTP020 B     B$RMRK                    RETURN CONTROL TO REM STNNT RTN
                                 49 *
                                 50 ************************************************************************
                                 51 * 'PAUSE' STATEMENT ROUTINE PMC AND STORAGE PARAMETERS
                                 52 ************************************************************************
                                 53 *
    1CFA 04               1CFA   54 BTPHTC DC    AL(B@LCOP)(B@CHLT)        'HLT' INSTRUCTION OPCODE
                                 55 *
                                 56 ************************************************************************
                                 57 *
                                 58 * END OF 'PAUSE' STATEMENT ROUTINE CODING
                                 59 *
                                 61 ***********************************************************************
                                 62 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                                 63 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                                 64 *                                                                     *
                                 65 ***********************************************************************
                                 66 *STATUS                                                               *
                                 67 *   VERSION 1 MODIFICATION 0                                          *
                                 68 *                                                                     *
                                 69 *FUNCTION                                                             *
                                 70 *   BMUPRT IS EXECUTED TO TRANSLATE MAT PRINT USING STATEMENTS AS THEY*
                                 71 *   OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO   *
                                 72 *   PLACE THE PSEUDOCODE IN VIRTUAL MEMORY.                           *
                                 73 *                                                                     *
                                 74 *ENTRY POINTS                                                         *
                                 75 *   BMUPRT HAS ONLY ONE ENTRY POINT:                                  *
                                 76 *        BMUPRT - TRANSLATE MAT PRINT USING STATEMENT                 *
                                 77 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                                 78 *        B     BMUPRT                                                 *
                                 79 *                                                                     *
                                 80 *INPUT                                                                *
                                 81 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                                 82 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER OF THE       *
                                 83 *     LEADING KEYWORD, MAT PRINT USING.                               *
                                 84 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                                 85 *     CHARACTER IN THE LEADING KEYWORD, MAT ERINT USING.              *
                                 86 *                                                                     *
                                 87 *OUTPUT                                                               *
                                 88 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                                 89 *   * GENERATED BY BRUFRT IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                                 90 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                                 91 *     SEQUENCES.                                                      *
                                 92 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                                 93 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                                 94 *                                                                     *
                                 95 *EXTERNAL REFERENCES                                                  *
                                 96 *     B$GETC - (B$NUMC, B$GPTR) - ENTR, TO BASIC RETRIEVAL ROUTINE.   *
                                 97 *     B$PUTC - (B$PCAD, B$PNBY, B$PVAD) - ENTRY TO COMPILER VIRTUAL   *
                                 98 *              MEMORY ROUTINE.                                        *
                                 99 *     B$BTAW - B$BRVA, B$BRIN) - BASIC COMPILER BRANCH TABLE ROUTINE. *
                                100 *     B$ZDBN - (B$BINO) - ENTRY TO COMPILER ZONED DECIMAL TO BINARY   *
                                101 *              ROUTINE.                                               *
                                102 *     B$MATR - ENTRY TO BASIC COMPILER MATRIX REFERENCE ROUTINE.      *
                                103 *     B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                   *
                                104 *                                                                     *
                                105 *EXITS, NORMAL                                                        *
                                106 *   B$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                     *
                                107 *                                                                     *
                                108 *EXITS, ERROR                                                         *
                                109 *   N/A                                                               *
                                110 *                                                                     *
                                111 *TABLES/WORK AREAS                                                    *
                                112 *   N/A                                                               *
                                113 *                                                                     *
                                114 *ATTRIBUTES                                                           *
                                115 *   BRUPRT IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                                116 *                                                                     *
                                117 *CHARACTER CODE DEPENDENCY                                            *
                                118 *   THE OPERATION OF THIS MULE DOES NOT DEPEND UPON A PARTICULAR      *
                                119 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SEI.            *
                                120 *                                                                     *
                                121 *NOTES                                                                *
                                122 *   ERROR PROCEDURES                                                  *
                                123 *      N/A                                                            *
                                124 *                                                                     *
                                125 *   REGISTER USAGE                                                    *
                                126 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                                127 *                                                                     *
                                128 *   SAVED/RESTORED AREAS                                              *
                                129 *      N/A                                                            *
                                130 *                                                                     *
                                131 *   MODIFICATION CONSIDERATIONS                                       *
                                132 *      BMUPRT RESIDES ON THE SAME SECTOR WITH BXCLOS AND BTSTOP.   1-4*
                                133 *      ANY MODIFICATION TO BMUPRT MUST TAKE INTO CONSIDERATION     1-4*
                                134 *      THIS CO-RESIDENCY SINCE IT WILL CHANGE THE ENTRY ADDRESSES  1-4*
                                135 *      OF BXCLOS AND BTSTOP. THE LIMITATION OF THE SECTOR          1-4*
                                136 *      BOUNDARY ON SIZE MUST ALSO BE CONSIDERED.                   1-4*
                                137 *                                                                     *
                                138 *   REQUIRED MODULES                                                  *
                                139 *        @SYSEQ - COMMON SYSTEM EQUATES                               *
                                140 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES    *
                                141 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS               *
                                142 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES                    *
                                143 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES                         *
                                144 *        @ERMEQ - ERROR MESSAGE EQUATES                               *
                                145 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES                       *
                                146 *        $B$EQU - COMPILER FIXED EQUATES                              *
                                147 *        $B@EQU - COMPILER SYSTEM EQUATES                             *
                                148 *                                                                     *
                                149 *   OTHER                                                             *
                                150 *      BMUPRT IS ASSEMBLED WITH ALL THE STATEMENT PROCESSORS.         *
                                151 ***********************************************************************
    1D00                        152        ORG   *,256,0                   BEGIN AT CORE PAGE BOUNDARY  1-4
                          1D00  153        USING *,@BR                     DEFINE BASE ADDR FOR CORE PS 1-4
                                154 *
                                155 * ENTER BMUPRT - MAT PRINT USING STATEMENT ROUTINE
                                156 *
                          1D00  157 BMUPRT EQU   *                         BMUPRT ENTRY POINT
                                158 *
                                159 * SET GET ROUTINE TO SKIP TO CHAR FOLLOWING 'MAT PRINT USING'
                                160 *
    1D00 3C 0D 0873             161 BMU010 MVI   B$NUMC,B@LMPU             SET GET TO SKIP KEYWORDS
    1D04 C0 87 0867             162        B     B$GETC                    LINK TO ADVANCE POINTER
                                163 *
                                164 * GENERATE 'STA' INSTRUCTION 'MACE IN V.M.
                                165 *
    1D08 D2 02 88               166 BMU020 LA    BMUSTC(,@BR),@XR          LOAD CADDR OF 'STA' INSTR
    1D0B 34 02 0A40             167        ST    B$PCAD,@XR                SET VADDR PARAN OF PUT FOR STA
    1D0F 3C 02 0A41             168        MVI   B$PNBY,B@LSTA-1           SET LNG PARAN OF PUT FOR 'STA'
    1D13 C0 87 093A             169        B     B$PUTC                    LINK TO GENERATE 'STA' INSTR
                                170 *
                                171 * ESTABLISH 'STA' OPERAND FOR BRANCH TABLE ADDRESS RESOLUTION
                                172 *
    1D17 0C 01 19EF 0A43        173 BMU030 MVC   B$BRVA,B$PVAD(@VADDR)     SET VADDR FOR BR TBL RESOLUTION
    1D1D 1F 01 19EF 94          174        SLC   B$BRVA,BMUBN1(@VADDR,@BR) ADJUST TO 'STA' OPND
                                175 *
                                176 * GENERATE A 'BMX' INSTRUCTION IMAGE IN V.M.
                                177 *
    1D22 D2 02 8B               178 BMU040 LA    BMUBNC(,@BR),@XR          LOAD CADDR OF 'BMX' INSTR
    1D25 34 02 0A40             179        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR ICI
    1D29 3C 02 0A41             180        MVI   B$PNBY,B@LBNX-1           SET LNG PARM OF PUT FOR WU
    1D2D C0 87 093A             181        B     B$PUTC                    LINK TO GENERATE 'BMX' INSTR
    1D31 35 02 0878             182        L     B$GPTR,@XR                RESTORE TEXT POINTER
                                183 *
                                184 * ESTABLISH NEXT AVAILABLE ADDR IN V.M. FOR BR TBL RESOLUTION (I.E.
                                185 * THE VADDR OF 1ST INSTR IN DATA OUTPUT SEQUENCE)
                                186 *
    1D35 0C 01 19F1 0A43        187 BMU050 MVC   B$BRLN,B$PVAD(@VADDR)     SET VADDR FOR BR TBL RESOLUTION
    1D3B C0 87 1996             188        B     B$BTAB                    LINK TO RESOLVE BR TBL ADDRS
                                189 *
                                190 * ESTABLISH 'BNX' INSTR OPND FOR ADDRESS RESOLUTION
                                191 *
    1D3F 0C 01 19EF 0A43        192 BMU060 MVC   B$BRVA,B$PVAD(@VADDR)     SET VADDR FOR BR TBL RESOLUTION
    1D45 1F 01 19EF 94          193        SLC   B$BRVA,BMUBN1(@VADDR,@BR) ADJUST TO 'BNX' OPND
                                194 *
                                195 * CONVERT THE LINE NUMBER OF THE IMAGE STATEMENT TO BINARY
                                196 *
    1D4A C0 87 19F2             197 BMU070 B     B$ZDBN                    LINK TO CONVERT LINE NO TO BINARY
                                198 *
                                199 * ESTABLISH IMAGE LN NO AS RESOLUTION LN NG
                                200 *
    1D4E 0C 01 19F1 1A6A        201 BMU080 MVC   B$BRLN,B$BINO(@VADDR)     SET RESOLUTION LINE NO
    1D54 C0 87 1996             202        B     B$BTAB                    LINK TO RESOLVE BR TBL ADDRS
                                203 *
                                204 * CALL MATRIX REFERENCE PROCESSOR TO GENERATE DOPE VECTOR STACKING
                                205 * INSTRUCTIONS IN VIRTUAL MEMORY
                                206 *
    1D58 C0 87 18F3             207 BMU090 B     B$MATR                    LINK TO PROCESS MAT-REFERENCE
                                208 *
                                209 * GENERATE 'MF1' INSTRUCTION IN V.M. TO INDICATE MAT PRINT USING
                                210 *
    1D5C D2 02 8E               211 BMU100 LA    BMUMFC(,@BR),@XR          LOAD CADDR OF 'MF1' INSTR
    1D5F 34 02 0A40             212        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'MF1'
    1D63 3C 02 0A41             213        MVI   B$PNBY,B@LMF1-1           SET LNG PARM OF PUT FOR 'MF1'
    1D67 C0 87 093A             214        B     B$PUTC                    LINK TO GENERATE 'MF1' PMC
                                215 *
                                216 * TEST LIST DELIMITER FOR BEING A STATEMENT TERMINATOR
                                217 *
    1D6B 35 02 0878             218 BMU110 L     B$GPTR,@XR                RESTORE TEXT POINTER
    1D6F BD 1E 00               219        CLI   B@CHAR(,@XR),B@EOST       IF DELIMITER IS NOT EOS
    1D72 D0 01 58               220        BNE   BMU090(,@BR)              * GO PROCESS NEXT MAT REFERENCE
                                221 *
                                222 * GENERATE 'PRU' INSTRUCTION WITH OPCOEE TO INDICATE IMAGE RELEASE
                                223 *
    1D75 D2 02 91               224 BMU120 LA    BMUPRC(,@BR),@XR          LOAD CADDR OF 'PRU' INSTR
    1D78 34 02 0A40             225        ST    B$PCAD,@XR                SET VADDR PARM OF PUT FOR 'PRU'
    1D7C 3C 01 0A41             226        MVI   B$PNBY,B@LPRU-1           SET LNG PARM OF PUT FOR 'PRU'
    1D80 C0 87 093A             227        B     B$PUTC                    LINK TO GENERATE 'PRU' INSTR
                                228 *
                                229 * RETURN CONTROL TO COMPILER DISTRIBUTOR
                                230 *
    1D84 C0 87 0700             231 BMU130 B     B$DIST                    RETURN TO DISTRIBUTOR

                                233 ***********************************************************************
                                234 * MAT PRINT USING STATEMENT RTN STORAGE AND PARAMETER AREAS
                                235 ***********************************************************************
                                236 *
    1D88 34               1D88  237 BMUSTC DC    AL(B@LCOP)(B@CSTA)        'STA' INSTR OPCODE
    1D89 0000             1D8A  238 BMUSTO DC    XL(B@LCVA)'00'            'STA' INSTR OPND IMAGE
                                239 *
    1D8B 4A               1D8B  240 BMUBNC DC    AL(B@LCOP)(B@CBNX)        'BNX' INSTR OPCODE
    1D8C 0000             1D8D  241 BMURNO DC    XL(B@LCVA)'00'            'BNX' INSTR OPND IMAGE
                                242 *
    1D8E 18               1D8E  243 BMUMFC DC    AL(B@LCOP)(B@CMF1)        'MF1' INSTR OPCODE
    1D8F 3F13             1D90  244 BMUMFO DC    AL(B@LCVA)(V$XMPU)        'MF1' INSTR OPERAND
                                245 *
    1D91 62               1D91  246 BMUPRC DC    AL(B@LCOP)(B@CPRU)        'PRU' INSTR OPCODE
    1D92 10               1D92  247 BMUPRO DC    AL(B@LCXX)(B@PUTM)        'PRU' INSTR OPND
                                248 *
                                249 * CONSTANTS
                                250 *
    1D93 0001             1D94  251 BMUBN1 DC    IL(@CADDR)'1'             BINARY 1
                                252 *
                                253 ***********************************************************************
                                254 *
                                255 * END OF MAT PRINT USING STATEMENT ROUTINE CODING
                                256 *
                                258 ***********************************************************************
                                259 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                                260 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                                261 *                                                                     *
                                262 ***********************************************************************
                                263 *STATUS                                                               *
                                264 *   VERSION 1 MODIFICATION 0                                          *
                                265 *                                                                     *
                                266 *FUNCTION                                                             *
                                267 *   BXCLOS IS EXECUTED TO TRANSLATE CLOSE STATEMENTS AS THEY OCCUR    *
                                268 *   IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE   *
                                269 *   THE PSEUDOCODE INTO VIRTUAL MEMORY.                               *
                                270 *                                                                     *
                                271 *ENTRY POINTS                                                         *
                                272 *   BXCLOS HAS ONLY ONE ENTRY POINT:                                  *
                                273 *      BXCLOS - TRANSLATE CLOSE STATEMENT                             *
                                274 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                                275 *      B  BXCLOS                                                      *
                                276 *                                                                     *
                                277 *INPUT                                                                *
                                278 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                                279 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE       *
                                280 *     LEADING KEYWORD. CLOSE.                                         *
                                281 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                                282 *     CHARACTER IN THE LEADING KEYWORD. CLOSE.                        *
                                283 *                                                                     *
                                284 *OUTPUT                                                               *
                                285 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                                286 *     GENERATED BY BXCLOS IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                                287 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                                288 *     SEQUENCES.                                                      *
                                289 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                                290 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                                291 *                                                                     *
                                292 *EXTERNAL REFERENCES                                                  *
                                293 *      B$GETC - (B$NUMC) - ENTRY TO BASIC TEXT RETRIEVAL ROUTINE.     *
                                294 *      B$PUTC - (B$PCAD, B$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY   *
                                295 *               OUTPUT ROUTINE.                                       *
                                296 *      BSDIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.                  *
                                297 *                                                                     *
                                298 *EXITS, NORMAL                                                        *
                                299 *    BSDIST - ENTRY TO THE BASIC COMPILER DISTRIBUTOR                 *
                                300 *                                                                     *
                                301 *EXITS, ERROR                                                         *
                                302 *   N/A                                                               *
                                303 *                                                                     *
                                304 *TABLES/WORK AREAS                                                    *
                                305 *   N/A                                                               *
                                306 *                                                                     *
                                307 *ATTRIBUTES                                                           *
                                308 *   BXCLOS IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                                309 *                                                                     *
                                310 *CHARACTER CODE DEPENDENCY                                            *
                                311 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                                312 *   INTERNAL REPRESENTATION OF TIE EXTERNAL CHARACTER SET.            *
                                313 *                                                                     *
                                314 *NOTES                                                                *
                                315 *   ERROR PROCEDURES                                                  *
                                316 *      N/A                                                            *
                                317 *                                                                     *
                                318 *   REGISTER USAGE                                                    *
                                319 *      BOTH THE INDEX AND BASE REGI,TERS ARE USED DURING EXECUTION.   *
                                320 *                                                                     *
                                321 *   SAVED/RESTORED AREAS                                              *
                                322 *      N/A                                                            *
                                323 *                                                                     *
                                324 *   MODIFICATION CONSIDERATIONS                                       *
                                325 *      BXCLOS RESIDES ON THE SAME SECTOR WITH BMUPRT AND BTSTOP.   1-4*
                                326 *      ANY MODIFICATION TO BXCLOS MUST TAKE INTO CONSIDERATION     1-4*
                                327 *      THIS CO-RESIDENCY SINCE IT WILL CHANGE THE ENTRY ADDRESS    1-4*
                                328 *      OF BTSTOP. THE LIMITATION OF THE SECTOR BOUNDARY ON SIZE    1-4*
                                329 *      MUST ALSO BE CONSIDERED.                                    1-4*
                                330 *                                                                     *
                                331 *   REQUIRED MODULES                                                  *
                                332 *        @SYSEQ - COMMON SYSTEM EQUATES                               *
                                333 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES    *
                                334 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS               *
                                335 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES                    *
                                336 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES                         *
                                337 *        @ERMEQ - ERROR MESSAGE EQUATES                               *
                                338 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES                       *
                                339 *        $B$EQU - COMPILER FIXED EQUATES                              *
                                340 *        $B@EQU - COMPILER SYSTEM EQUATES                             *
                                341 *                                                                     *
                                342 *   OTHER                                                             *
                                343 *      BXCLOS IS ASSEMOLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                                344 ***********************************************************************

                                346 *
                                347 * ENTER BXCLOS - 'CLOSE' STATEMENT ROUTINE
                                348 *
                          1D95  349 BXCLOS EQU   *                          BXCLOS ENTRY POINT
                                350 *
                                351 * SET GET ROUTINE TO SKIP TO 'E' IN KEYWORD 'CLOSE'
                                352 *
    1D95 3C 04 0873             353 BXC010 MVI   B$NUMC,B@LKCL-1           SET GET TO SKIP TO 'E'
    1D99 C0 87 0867             354        B     B$GETC                    LINK TO ADVANCE POINTER
    1D9D C0 87 14B0             355 BXC020 B     B$CSCN                    LINK TO PROCESS FILE REFERENCE
                                356 *
                                357 * GENERATE THE 'ADF' PMC IN V.M. IF OPND IS NOT ZERO
                                358 *
    1DA1 D2 02 D1               359 BXC120 LA    BXCAFC(,@BR),@XR          LOAD CADDR OF 'ADE' INSTR
    1DA4 34 02 0A40             360        ST    B$PCAD,@XR                SET VADDR PARAM OF PUT FOR 'ADE'
    1DA8 3C 01 0A41             361        MVI   B$PNBY,B@LADF-1           SET LNG PARAM OF PUT FOR 'ADE'
    1DAC C0 87 093A             362        B     B$PUTC                    LINK TO GENERATE 'ADE' PMC
                                363 *
                                364 * GENERATE THE 'CLS' PMC IN V.M.
                                365 *
    1DB0 D2 02 D3               366 BXC130 LA    BXCCLC(,@BR),@XR          LOAD CADOR OF 'CLS' INSTR
    1DB3 34 02 0A40             367        ST    B$PCAD,@XR                SET VADOR PARAM OF PUT FOR CL:
    1DB7 3C 00 0A41             368        MVI   B$PNBY,B@LCLS-1           SET LNG PARAM OF PUT FOR 'CLS'
    1DBB C0 87 093A             369        B     B$PUTC                    LINK TO GENERATE 'CLS' PMC
```

```
                                370 *
                                371 * TEST NEXT LIST CHARACTER FOR BEING AN END-OF-STATEMENT
                                372 *
    1DBF 3C 00 0873             373 BXC140 MVI   B$NUMC,B@GETS             DISABLE GET ROUTINE
    1DC3 C0 87 0867             374        B     B$GETC                    LINK TO GET CHARACTER POINTER
    1DC7 BD 1E 00               375        CLI   B@CHAR(,@XR),B@EOST       IF CHAR IS EOS
    1DCA D0 01 9D               376        BNE   BXC020(,@BR)              * BRANCH TO PROCESS FILENAME
                                377 *
                                378 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
                                379 *
    1DCD C0 87 0700             380 BXC150 B     B$DIST                    RETURN TO DISTRIBUTOR

                                382 ***********************************************************************
                                383 * 'CLOSE' STATEMENT PARAMETER AND STORAGE AREAS
                                384 ***********************************************************************
                                385 *
    1DD1 58               1DD1  386 BXCAFC DC    AL(B@LCOP)(B@CADF)        'ADF' INSTR OPCODE
    1DD2 00               1DD2  387 BXCAFO DC    XL1'00'                   'ADF' INSTR OPERAND
                                388 *
    1DD3 5E               1DD3  389 BXCCLC DC    AL(B@LCOP)(B@CCLS)        'CLS' INSTR OPCODE

                                391 ***********************************************************************
                                392 * 'CLOSE' STATEMENT CONSTANTS AND EQUATES
                                393 ***********************************************************************
                                394 *
                                395 * CONSTANTS
                                396 *
                          1DD4  397 BXCSFA EQU   *
                                398 *
    1DD4 0001             1DD5  399 BXCBN1 DC    IL(@CADDR)'1'             BINARY '1'
                                400 *
                                401 * END OF 'CLOSE' STATEMENT ROUTINE CODING
                                402 *
                                404 ***********************************************************************
                                405 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                                406 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                                407 *                                                                     *
                                408 ***********************************************************************
                                409 *STATUS                                                               *
                                410 *   VERSION 1 MODIFICATION 0                                          *
                                411 *                                                                     *
                                412 *FUNCTION                                                             *
                                413 *   BTSTOP IS EXECUTED TO TRANSLATE STOP STATEMENTS AS THEY OCCUR IN  *
                                414 *   A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE  *
                                415 *   PSEUDOCODE IN VIRTUAL MEMORY.                                     *
                                416 *                                                                     *
                                417 *ENTRY POINTS                                                         *
                                418 *   BTSTOP HAS ONLY ONE ENTRY POINT:                                  *
                                419 *        BTSTOP - TRANSLATE STOP STATEMENT                            *
                                420 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                                421 *        B     BTSTOP                                                 *
                                422 *                                                                     *
                                423 *INPUT                                                                *
                                424 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                                425 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER OF THE       *
                                426 *     LEADING KEYWORD, STOP.                                          *
                                427 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE 1ST   *
                                428 *     CHARACTER IN THE LEADING KEYWORD, STOP.                         *
                                429 *                                                                     *
                                430 *OUTPUT                                                               *
                                431 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                                432 *     GENERATED BY BTSTOP IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                                433 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                                434 *     SEQUENCES.                                                      *
                                435 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                                436 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                                437 *                                                                     *
                                438 *EXTERNAL REFERENCES                                                  *
                                439 *        B$PUTC - (B$PCAD, B$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY *
                                440 *                 OUTPUT ROUTINE.                                     *
                                441 *        B$RMRK - ENTRY TO BASIC COMPILER REMARK ROUTINE.             *
                                442 *                                                                     *
                                443 *EXITS, NORMAL                                                        *
                                444 *   B$RMRK - ENTRY TO BASIC COMPILER REMARK ROUTINE.                  *
                                445 *                                                                     *
                                446 *EXITS, ERROR                                                         *
                                447 *   N/A                                                               *
                                448 *                                                                     *
                                449 *TABLES/WORK AREAS                                                    *
                                450 *   N/A                                                               *
                                451 *                                                                     *
                                452 *ATTRIBUTES                                                           *
                                453 *   BTSTOP IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                                454 *                                                                     *
                                455 *CHARACTER CODE DEPENDENCY                                            *
                                456 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                                457 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                                458 *                                                                     *
                                459 *NOTES                                                                *
                                460 *   ERROR PROCEDURES                                                  *
                                461 *      N/A                                                            *
                                462 *                                                                     *
                                463 *   REGISTER USAGE                                                    *
                                464 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                                465 *                                                                     *
                                466 *   SAVED/RESTORED AREAS                                              *
                                467 *      N/A                                                            *
                                468 *                                                                     *
                                469 *   MODIFICATION CONSIDERATIONS                                       *
                                470 *      BTSTOP RESIDES ON THE SAME SECTOR WITH BMUPRT AND BXCLOS.   1-4*
                                471 *      ANY MODIFICATION TO BTSTOP MUST TAKE INTO CONSIDERATION     1-4*
                                472 *      THIS CO-RESIDENCY AND ALSO THE LIMITATION OF THE SECTOR     1-4*
                                473 *      BOUNDARY ON SIZE.                                           1-4*
                                474 *                                                                     *
                                475 *   REQUIRED MODULES                                                  *
                                476 *        @SYSEQ - COMMON SYSTEM EQUATES                               *
                                477 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES    *
                                478 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS               *
                                479 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES                    *
                                480 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES                         *
                                481 *        @ERMEQ - ERROR MESSAGE EQUATES                               *
                                482 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES                       *
                                483 *        $B$EQU - COMPILER FIXED EQUATES                              *
                                484 *        $B@EQU - COMPILER SYSTEM EQUATES                             *
                                485 *                                                                     *
                                486 *   OTHER                                                             *
                                487 *      BTSTOP IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                                488 ***********************************************************************

                                490 *
                                491 * ENTER BTSTOP - 'STOP' STATEMENT ROUTINE
                                492 *
                          1DD6  493 BTSTOP EQU   *                         BTSTOP ENTRY POINT
                                494 *
                                495 * GENERATE AN 'SVC' INSTRUCTION IN VIRTUAL MEMORY
                                496 *
    1DD6 D2 02 E9               497 BTS010 LA    BTSSVC(,@BR),@XR          LOAD CADDR OF 'SVC' INSTR
    1DD9 34 02 0A40             498        ST    B$PCAD,@XR                SET PUT RTN FOR VADDR OF 'SVC'
    1DDD 3C 00 0A41             499        MVI   B$PNBY,B@LSVC-1           SET PUT RTN FOR LENGTH OF 'SVC'
    1DE1 C0 87 093A             500        B     B$PUTC                    LINK TO GENERATE PMC
                                501 *
                                502 * RETURN CONTROL TO THE REMARK STATEMENT ROUTINE
                                503 *
    1DE5 C0 87 1AE6             504 BTS020 B     B$RMRK                    RETURN TO REMARK VINT RTN

                                506 ***********************************************************************
                                507 * 'STOP' STATEMENT ROUTINE PMC AND STORAGE PARAMETERS
                                508 ***********************************************************************
                                509 *
    1DE9 02               1DE9  510 BTSSVC DC    AL(B@LCOP)(B@CSVC)        'SVC' INSTR OPCODE
                                511 *
                                512 ***********************************************************************
                                513 *
                                514 * END OF 'STOP' STATEMENT ROUTINE CODING
                                515 *
                                517 ***********************************************************************
                                518 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                                519 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                                520 *                                                                     *
                                521 ***********************************************************************
                                522 *STATUS                                                               *
                                523 *   VERSION 1 MODIFICATION 0                                          *
                                524 *                                                                     *
                                525 *FUNCTION                                                             *
                                526 *   * BTRMNT IS EXECUTED TO TRANSLATE THE FIRST END STATEMENT OR      *
                                527 *     END-OF-FILE RECORD ENCOUNTERED IN THE SOURCE PROGRAM TEXT INTO  *
                                528 *     THE APPROPRIATE PSEUDOCODE AND TO PLACE THE PSEUDOCODE IN       *
                                529 *     VIRTUAL MEMORY.                                                 *
                                530 *   * BTRMNT ALSO PERFORMS THE FOLLOWING FUNCTIONS:                   *
                                531 *   * BASIC PROGRAM PROCESSING IS ABORTED IN THE PRESENCE OF ANY      *
                                532 *     LOGGED OR CURRENTLY ENCOUNTERED COMPILER ERROR CONDITION.       *
                                533 *     RISIDUAL CORE-RESIDENT PMC AND PROGRAM GENERATED CONSTANTS ARE  *
                                534 *     WRITTEN TO DISK VIRTUAL MEMORY, PMC GENERATION IS CLOSED.       *
                                535 *   * RISIDUAL STATEMENT ADDRESS TABLE AND BRANCH ADDRESS TABLE       *
                                536 *     ENTRIES ARE WRITTEN TO THE RESPECTIVE DISK FILES, ADDRESS TABLE *
                                537 *     FILES ARE CLOSED.                                               *
                                538 *   * CRITICAL VIRTUAL ADDRESSES ARE ESTABLISHED IN A HIGH CORE       *
                                539 *     PARAMETER REGION FOR TRANSFER TO THE NEXT PROCESSOR PHASE.      *
                                540 *   * SCALAR VARIABLE SYMBOL TABLES ARE ORGANIZED AND ESTABLISHED     *
                                541 *     IN THE #LOADR PARAMETER TRANSFER AREA.                          *
                                542 *   * FUNCTION AND ARRAY SYMBOL TABLES ARE EXTRACTED FROM THE COMPILE *
                                543 *     TIME SYMBOL TABLE/ATTRIBUTE CONGLOMERATES AND ESTABLISHED IN    *
                                544 *     THE #LOADR PARAMETER TRANSFER AREA.                             *
                                545 *   * THE RUN-TIME FUNCTION AND ARRAY TABLE IS CONSTRUCTED IN THE     *
                                546 *     #LOADR PARAMETER TRANSFER AREA FROM DATA EXTRACTED FROM THE     *
                                547 *     COMPILE-TIME SYMBOL TABLE/ATTRIBUTE CONGLOMERATES; THIS TABLE   *
                                548 *     IS CONSTRUCTED AS IT WILL EVENTUALLY APPEAR IN VIRTUAL MEMORY.  *
                                549 *   * THE NEXT PROCESSOR PHASE (#LOADR) IS CORE-LOADED AND EXECUTED   *
                                550 *     USING SYSTEM ENTRY POINT #RLOAD.                                *
                                551 *                                                                     *
                                552 *ENTRY POINTS                                                         *
                                553 *   BTRMNT HAS ONLY ONE ENTRY POINT:                                  *
                                554 *        BTRMNT - TERMINATE COMPILATION                               *
                                555 *   THE FORMAT OF THE CALLING SEQUEICE IS:                            *
                                556 *        B     BTRMNT                                                 *
                                557 *                                                                     *
                                558 *INPUT                                                                *
                                559 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                                560 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER OF THE       *
                                561 *     LEADING KEYWORD, END.  IF THE END IS IMPLICIT THE RECORD        *
                                562 *     SEGMENT CONTAINS THE END-OF-STATEMENT CHARACTER.                *
                                563 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                                564 *     CHARACTER IN THE LEADING KEYWORD, END.  IF THE END IS IMPLICIT, *
                                565 *     THE CORE ADDRESS IS OF THE END-OF-STATEMENT CHARACTER.          *
                                566 *   * B$ERSN - SET TO ON STATUS WHEN COMPILE-TIME ERRORS HAVE BEEN    *
                                567 *     ENCOUNTERED AND LOGGED IN VIRTUAL MEMORY PRIOR TO BTRMNT        *
                                568 *     EXECUTION.                                                      *
                                569 *   * LOGGED ERRORS - WHEN B$ERSW IS FOUND ON, THE FIRST 3 VIRTUAL    *
                                570 *     MEMORY PAGES NORMALLY USED FOR PMC STORAGE ARE EXPECTED TO      *
                                571 *     CONTAIN FROM 1 TO 255 3-BYTE ERROR CODE RECORDS.                *
                                572 *   * DIPECT - WHEN MERU IS ON, THIS IS EXPECTED TO CONTAIN A COUNT   *
                                573 *      OF THE NUMBER OF ERROR CODE RECORDS LOGGED IN VIRTUAL MEMORY;   *
                                574 *      THIS IS NEVER PERMITTED TO EXCEED A VALUE OF 255.               *
                                575 *    * B$FTPT - CONTAINS THE CORE ADDRESS OF THE FIRST BYTE IN THE     *
                                576 *      TOP FOR TABLE ENTRY.  WHEN THIS IS NOT IDENTICAL WITH THE       *
                                577 *      ADDRESS OF THE TABLE ITSELF, AN INCOMPLETE FOR LOOP IS          *
                                578 *      INDICATED.                                                      *
                                579 *    * B$PVAD - CONTAINS THE VIRTUAL ADDRESS OF THE NEXT AVAILABLE PMC *
                                580 *      BYTE, AND IS USED TO ESTABLISH THE LAST PAGE OCCUPIED BY        *
                                581 *      PMC FOR VM REGION 1 DEFINITION.                                 *
                                582 *    * B$PCPG - CONTAINS THE VIRTUAL PAGE NUMBER OF THE PAGE CURRENTLY,*
                                583 *      BEING FILLED WITH PROGRAM GENERATED CONSTANTS, AND USED TO      *
                                584 *      DEFINE THE UPPER BOUNDARY ADDRESS OF VM REGION 19               *
                                585 *    * B$CVPD - CONTAINS THE DISPLACEMENT VALUE USED AS A CONSTANT     *
                                586 *      OUTPUT BUFFER POINTER WHEN THIS VALUE IS LESS THAN X'FF',       *
                                587 *      RISIDUAL BUFFER CONSTANTS ARE INDICATED.                        *
                                588 *    * B$BSDA - CONTAINS THE LOGICAL SECTOR ADDRESS OF THE SECTOR      *
                                589 *      CURRENTLY BEING FILLED WITH BRANCH TABLE ENTRIES.               *
                                590 *    * B$SVPB - CONTAINS THE VIRTUAL ADDRESS OF THE NEXT BYTE          *
                                591 *      AVAILABLE FOR PROGRAM VARIABLE ALLOCATION.                      *
                                592 *    * B$SFAB - CONTAINS THE VIRTUAL ADDRESS OF THE FIRST BYTE IN THE  *
                                593 *      LAST ARRAY DOPE VECTOR OR USER FUNCTION ADDRESS DEFINED IN THE  *
                                594 *      PROGRAM.                                                        *
                                595 *    * B$FAIS - CONTAINS THE VIRTUAL ADDRESS OF THE FIRST BYTE         *
                                596 *      ALLOCATED FOR INTERNAL CONSTANTS IN THE PROGRAM.                *
                                597 *    * B$FAIW - CONTAINS THE VIRTUAL ADDRESS OF THE FIRST BYTE         *
                                598 *      ALLOCATED FOR INTERNAL VARIABLES IN THE PROGRAM.                *
                                599 *    * $EXFTR - CONTAINS A COUNT OF THE NUMBER OF CORE PAGES AVAILABLE *
                                600 *      BEYCND 8K FOR GENERAL PROGRAM UTILIZATION.                      *
                                601 *    * B$SLVT - THE 58-BYTE SYMBOL TABLE CONTAINING VIRTUAL ADDRESSES  *
                                602 *      FOR EACH LETTER VARIABLE DEFINED IN THE PROGRAM.                *
                                603 *    * B$SLDT - THE 580-BYTE SYMBOL TABLE CONTAINING VIRTUAL ADDRESSES *
                                604 *      FOR EACH LETTER-DIGIT VARIABLE DEFINED IN THE PROGRAM.          *
                                605 *    * B$SCVT - THE 58-BYTE SYMBOL TABLE CONTAINING VIRTUAL ADDRESSES  *
                                606 *      FOR EACH CHARACTER VARIABLE DEFINED IN THE PROGRAM.             *
                                607 *    * B$SNAT - THE 174-BYTE SYMBOL/ATTRIBUTE TABLE CONTAINING VIRTUAL *
                                608 *      ADDRESSES AND DOPE VECTOR INFORMATION FOR EACH ARITHMETIC ARRAY *
                                609 *      DEFINED IN THE PROGRAM.                                         *
                                610 *    * B$SCAT - THE 116-BYTE SYMBOL/ATTRIBUTE TABLE CONTAINING VIRTUAL *
                                611 *      ADDRESSES AND DOPE VECTOR INFORMATION FOR EACH CHARACTER ARRAY  *
                                612 *      DEFINED IN THE PROGRAM.                                         *
                                613 *    * B$SFNT - THE 116-BYTE SYMBOL/ATTRIBUTE TABLE CONTAINING VIRTUAL.*
                                614 *      ADDRESSES AND RUN-TIME ENTRY POINTS FOR EACH USER FUNCTION      *
                                615 *      DEFINED IN THE PROGRAM.                                         *
                                616 *                                                                      *
                                617 *OUTPUT                                                                *
                                618 *    * VIRTUAL MEMORY - IN THE ABSENCE OF ANY ERROR CONDITION, THE PMC *
                                619 *      SEQUENCE GENERATED UNDER CONTROL OF BTRMNT IS STORED IN THE     *
                                620 *      NEXT AVAILABLE VIRTUAL MEMORY LOCATION FOLLOWING PREVIOUSLY     *
                                621 *      STORED INSTRUCTION SEQUENCES, VIRTUAL MEMORY IS THEN CLOSED     *
                                622 *      FOR BOTH PMC AND PROGRAM GENERATED CONSTANTS.                   *
                                623 *    * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                                624 *      CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                                625 *    * $CAERR - WHEN ERROR 2 OR ERROR 3 (SEE ERROR PROCEDURES UNDER    *
                                626 *      NOTES) IS IN EFFECT, THIS IS SET TO CONTAIN A CODE DEFINING     *
                                627 *      THE APPROPRIATE ERROR MESSAGE FOR #ERRPG.                       *
                                628 *    * #ERRPG - WHEN ERROR 1 IS IN EFFECT, THIS IS SET TO CODE $ERSTK  *
                                629 *     TO INDICATE MULTIPLE ERROR MESSAGE DISPLAY.  WHEN ERROR 2 OR    *
                                630 *     ERROR 3 IS IN EFFECT, THIS IS SET TO CODE $$$NLN TO INDICATE    *
                                631 *     THE SUPPRESSION OF LINE NUMBER DISPLAY.                         *
                                632 *   * $ERRCT - WHEN ERROR 1 IS IN EFFECT, THIS IS SET TO CONTAIN THE  *
                                633 *     VALUE IN ERROR RECORD COUNT B$PECT.                             *
                                634 *   * ERROR RECORD STACK - WHEN ERROR 1 IS IN EFFECT, CORE REGION     *
                                635 *     X'1C00' THROUGH X'1EFF' IS LOADED WITH THE ERROR RECORDS        *
                                636 *     LOGGED AT COMPILE TIME.                                         *
                                637 *   * $XIND1 - WHEN ERROR 1 IS IN EFFECT, THIS SYSTEM INDICATOR IS    *
                                638 *     CLEARED TO SPECIFY VIRTUAL MEMORY AS UNDEFINED.                 *
                                639 *   * STATEMENT ADDRESS TABLE FILE - A FINAL ENTRY (X'FFFF', X'FFFF') *
                                640 *     IS STORED IN THE LAST ENTRY POSITION OF THE STATEMENT ADDRESS   *
                                641 *     TABLE BUFFER, AND THE BUFFER IS OUTPUT TO CLOSE THE STATEMENT   *
                                642 *     ADDRESS TABLE FILE.                                             *
                                643 *   * BRANCH ADDRESS TABLE FILE - WHEN ERROR 3 IS NOT IN EFFECT, THE  *
                                644 *     BRANCH ADDRESS TABLE BUFFER IS OUTPUT TO CLOSE THE FILE.        *
                                645 *   * #LOADR PARAMETER TRANSFER AREA - A COMMON AREA FOR TRANSFER OF  *
                                646 *     INFORMATION BETWEEN THE COMPILER AND LOADER PHASES.             *
                                647 *                                                                     *
                                648 *EXTERNAL REFEREACES                                                  *
                                649 *     B$PUTC - (B$PFNC, B$PCAD, B$PNBY, B$PVAD, B$PCPG, B$ERSW) -     *
                                650 *              ENTRY TO COMPILER VIRTUAL MEMORY OUTPUT ROUTINE.       *
                                651 *     B$FCON - (B$CVPD) - ENTRY TO BASIC COMPILER CONSTANT ROUTINE.   *
                                652 *     B$SYMB - (B$SLVT, B$SLDT, B$SCVT, B$SNAT, B$SCAT, B$SFNT,       *
                                653 *              B$SVBB, B$SFAB) - ENTRY TO BASIC COMPILER SYMBOL       *
                                654 *              TRANSLATION ROUTINE.                                   *
                                655 *     B$SCAN - (B$FAIS, B$FAIW) - ENTRY TO BASIC COMPILER ARITHMETIC  *
                                656 *              EXPRESSION SCAN ROUTINE.                               *
                                657 *     B$BTAB - (B$BSDA, B$BDPL) - ENTRY TO BASIC COMPILER BRANCH      *
                                658 *              TABLE ROUTINE.                                         *
                                659 *     B$DIST - (B$DST2, B$SDPL) - ENTRY TO BASIC COMPILER DISTRIBUTOR *
                                660 *              BVDL4T.                                                *
                                661 *     COMMOM - (B$FORT, B$FTPT. B$LDRP, B$CSBF, B$CSXA) - ENTRY TO    *
                                662 *              COMMON CORE LOCATIONS OUTSIDE NUCLEUS.                 *
                                663 *     NUCLEUS - ($XIND1, $ERRPG, $ERRCT, $CAERR, $CAERK, $DISKN,      *
                                664 *              $WAITF, $EXFTR, $RLOAD) - ENTRY TO INDICATORS AND      *
                                665 *              ADDRESSES IN NUCLEUS.                                  *
                                666 *                                                                     *
                                667 *EXITS, NORMAL                                                        *
                                668 *   IN THE ABSENCE OF COMPILER ERRORS, CONTROL IS ALWAYS PASSED TO    *
                                669 *   SYSTEM LOADER                                                     *
                                670 *     $RLOAD                                                          *
                                671 *                                                                     *
                                672 *EXITS, ERROR                                                         *
                                673 *   THE FIRST ERROR CONDITION TO BE DISCOVERED CAUSES AN EXIT         *
                                674 *   TO SYSTEM ERROR MESSAGE ROUTINE                                   *
                                675 *     #ERRPG  VIA                                                     *
                                676 *     $CAERK  WITH APPROPRIATE ERROR CODE IN                          *
                                677 *     $CAERR                                                          *
                                678 *                                                                     *
                                679 *TABLES/WORK AREAS                                                    *
                                680 *   * SEE INPUT AND OUTPUT SECTIONS ABOVE.                            *
                                681 *   * BTREPL - THE DISK PARAMETER LIST USED TO CORELOAD ERROR RECORDS *
                                682 *     LOGGED IN VIRTUAL MEMORY WHEN B$ERSW IS ON.                     *
                                683 *   * BTRDPL - THE DISK PARAMETER LIST USED AS ARGUMENT FOR $RLOAD    *
                                684 *     DEFINING #LOADR DISK AND CORELOAD PARAMETERS.                   *
                                685 *                                                                     *
                                686 *ATTRIBUTES                                                           *
                                687 *   BTRMNT IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                                688 *                                                                     *
                                689 **CHARACTER CODE DEPENDENCY                                           *
                                690 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                                691 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                                692 *                                                                     *
                                693 *NOTES                                                                *
                                694 *   ERROR PROCEDURES                                                  *
                                695 *      ERROR 1 - SWITCH B$ERSW IS FOUND ON, INDICATING THAT AT LEAST  *
                                696 *        ONE COMPILE-TIME ERROR HAS BEEN GENERATED IN VIRTUAL MEMORY, *
                                697 *        VIRTUAL MEMORY IS SET UNDEFINED AND THE FIRST 3 PMC VIRTUAL  *
                                698 *        PAGES ARE READ INTO CORE.                                    *
                                699 *      ERROR 2 - THE FOR TABLE IS FOUND TO CONTAIN AT LEAST ONE ENTRY *
                                700 *        WHICH HAS NOT BEEN PAIRED WITH A MATCHING NEXT STATEMENT.    *
                                701 *        AN ERROR CODE IS ESTABLISHED FOR 'FOR/NEXT LOOP INCOMPLETE'. *
                                702 *      ERROR 3 - THE BRANCH ADDRESS TABLE FILE IS FILLED TO CAPACITY  *
                                703 *        AND MORE TABLE ENTRIES REMAIN TO BE OUTPUT.  AN ERROR CODE   *
                                704 *        IS ESTABLISHED FOR 'TOO MANY LINE NUMBER REFERENCES'.        *
                                705 *                                                                     *
                                706 *   REGISTER USAGE                                                    *
                                707 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                                708 *                                                                     *
                                709 *   SAVED/RESTORED AREAS                                              *
                                710 *      N/A                                                            *
                                711 *                                                                     *
                                712 *   MODIFICATION CONSIDERATIONS                                       *
                                713 *      BTRMNT RESIDES ON TWO SECTORS, CO-RESIDENT ON THE SECOND    1-4*
                                714 *      SECTOR WITH BKRTRN AND BPXRSR.  ANY MODIFICATION TO BTRMNT  1-4*
                                715 *      MUST MAINTAIN THE LINKAGE BETWEEN THE TWO SECTORS AND ALSO  1-4*
                                716 *      TAKE INTO CONSIDERATION THE CO-RESIDENCY SINCE A CHANGE     1-4*
                                717 *      TO BTRMNT CAN CHANGE THE ENTRY ADDRESSES OF BKRTRN AND      1-4*
                                718 *      BPXRSR.  THE LIMITATION OF THE SECTOR BOUNDARY ON SIZE      1-4*
                                719 *      MUST ALSO BE CONSIDERED.                                    1-4*
                                720 *                                                                     *
                                721 *  REQUIRED MODULE                                                    *
                                722 *        @$YSEQ - COMMON SYSTEM EQUATES.                              *
                                723 *        @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.   *
                                724 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS.              *
                                725 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                                726 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                                727 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                                728 *        $V$EQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                                729 *        $BSEQU - COMPILER FIXED EQUATES.                             *
                                730 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                                731 *                                                                     *
                                732 *   OTHER                                                             *
                                733 *      BTRMNT IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                                734 ***********************************************************************

    1E00                        736        ORG   *,256,0                   BEGIN AT CORE PAGE BOUNDARY
                          1E00  737        USING *,@BR                     DEFINE BASE ADDR FOR CORE PAGE
                                738 *
                                739 * ENTER BTRMNT - COMPILER TERMINATOR
                                740 *
                          1E00  741 BTRMNT EQU   *                         BTRMNT ENTRY POINT
    1E00 74 01 FB               742        ST    BTRCA2(,@BR),@BR          SAVE BTRMNT BASE ADDRESS
                                743 *
                                744 * TEST FOR COMPILER-GENERATED ERRORS
                                745 *
    1E03 38 07 0993             746 BTR010 TBN   B$ERSW,B$ERMK             TEST THE COMPILER ERROR SWITCH
    1E07 F2 90 21               747        JF    BTR040                    BRANCH IF NO COMPILER ERRORS
                                748 *
                                749 * COMPILER ERRORS - CORELOAD ERROR CODES FROM VIRTUAL MEMORY
                                750 *
    1E0A 3C 9D 094E             751 BTR020 MVI   B$PFNC,B$PFCL             SET PUT ROUTINE 'CLOSE' FUNC
    1E0E C0 87 093A             752        B     B$PUTC                    LINK TO CLOSE THE ERROR FILE
                                753 *
    1E12 D2 02 F2               754        LA    BTREPL(,@BR),@XR          LOAD COMPILER ERROR DPL CADDR
    1E15 C0 87 1A6B             755        B     B$DL4T                    LINK TO READ ERRORS FROM VM
                                756 *
                                757 * ERROR EXIT 1 - PRINT COMPILER-GENERATED STACKED ERROR MESSAGES
                                758 *
    1E19 3C 00 03D0             759 BTR030 MVI   $XIND1,@ZERO              DELETE VM DEFINITION INDICATOR
    1E1D 3C 30 03CE             760        MVI   $ERRPG,$ERSTK             SET ERROR RTN FOR STACKED CODE
    1E21 0C 00 03CF 0A44        761        MVC   $ERRCT,B$PECT(1)          SET ERROR RTN MESSAGE COUNT
    1E27 C0 87 0469             762        B     $CAERK                    EXIT TO SYSTEM ERROR ROUTINE
                                763 *
                                764 * TEST FOR AN INCOMPLETE 'FOR' LOOP IN THE PROGRAM
                                765 *
    1E2B 1D 01 1B0D ED          766 BTR040 CLC   B$FTPT,BTRFTA(@CADDR,@BR)  TEST FOR AN EMPTY 'FOR' TABLE
    1E30 F2 81 0C               767        JE    BTR060                     BRANCH IF NO ACTIVE 'FOR' ENTRY
                                768 *
                                769 * ERROR EXIT 2 - PRINT 'INCOMPLETE 'FOR' LOOP' ERROR MESSAGE
                                770 *
    1E33 3C A0 03CE             771 BTR050 MVI   $ERRPG,$$$NLN             SET FOR NO LINE NO. PRINTOUT
    1E37 3C AE 03CD             772        MVI   $CAERR,@@E609             SET THE ERROR MESSAGE CODE
    1E3B C0 87 0469             773        B     $CAERK                    EXIT TO SYSTEM ERROR ROUTINE
                                774 *
                                775 * GENERATE THE FINAL PROGRAM PSEUDO INSTRUCTION SEQUENCE - AN ERROR
                                776 * CONDITION (PROGRAM TOO LARGE) IS POSSIBLE AT THIS POINT
                                777 *
    1E3F D2 02 F8               778 BTR060 LA    BTRPCA(,@BR),@XR          LOAD FINAL PMC SEQUENCE CADDR
    1E42 34 02 0A40             779        ST    B$PCAD,@XR                SET PUT RTN CORE ADDR PARAMETER
    1E46 3C 01 0A41             780        MVI   B$PNBY,B@LSVC+B@LEOF-1    SET PUT RTN LENGTH PARAMETER
    1E4A C0 87 093A             781        B     B$PUTC                    LINK TO OUTPUT THE FINAL PMC
                                782 *
                                783 * CLOSE OUTPUT OF PSEUDO INSTRUCTIONS TO VIRTUAL MEMORY - AN ERROR
                                784 * CONDITION (PROGRAM TOO LARGE) IS POSSIBLE AT THIS POINT
                                785 *
    1E4E 3C 9D 094E             786 BTR070 MVI   B$PFNC,B$PFCL             SET PUT ROUTINE 'CLOSE' FUNC
    1E52 C0 87 093A             787        B     B$PUTC                    LINK TO CLOSE THE PMC FILE
                                788 *
                                789 * TEST FOR ANY CONSTANTS REMAINING TO BE OUTPUT
                                790 *
    1E56 3D FF 0C5D             791 BTR080 CLI   B$CVPD,BTRBND             TEST FOR AN EMPTY CONSTANT BFR
    1E5A F2 81 08               792        JE    BTR100                    BRANCH WHEN BUFFER IS EMPTY
                                793 *
                                794 * OUTPUT THE FINAL PAGE OF PROGRAM CONSTANTS - AN ERROR CONDITION
                                795 * (PROGRAM TOO LARGE) IS POSSIBLE AT THIS POINT
                                796 *
    1E5D 3C 15 094E             797 BTR090 MVI   B$PFNC,B$PFWP             SET PUT RTN TO WRITE A PAGE
    1E61 C0 87 093A             798        B     B$PUTC                    LINK TO OUTPUT CONSTANT BUFFER
                                799 *
                                800 * TEST FOR POSSIBLE OVERFLOW OF THE BRANCH ADDRESS TABLE FILE
                                801 *
    1E65 C2 02 19E8             802 BTR100 LA    B$BDPL,@XR                LOAD BRANCH TABLE DPL CADDR
    1E69 3D 60 19EA             803        CLI   B$BDSA,B@DTB1+B@DTBN      IF BRANCH ADDR FILE NOT FULL
    1E6D F2 82 0C               804        JL    BTR120                    * GO OUTPUT THE FINAL FILE BFR
                                805 *
                                806 * ERROR EXIT 3 - PRINT 'TOO MANY LINE NO. REFERENCES' ERROR MESSAGE
                                807 *
    1E70 3C A0 03CE             808 BTR110 MVI   $ERRPG,$$$NLN             SET FOR NO LINE NO. PRINTOUT
    1E74 3C B1 03CD             809        MVI   $CAERR,@@E612             SET THE ERROR MESSAGE CODE
    1E78 C0 87 0469             810        B     $CAERK                    EXIT TO SYSTEM ERROR ROUTINE
                                811 *
                                812 * OUTPUT THE FINAL BRANCH ADDRESS TABLE BUFFER TO DISK
                                813 *
    1E7C C0 87 1A6B             814 BTR120 B     B$DL4T                    LINK TO WRITE BRANCH TABLE BFR
                                815 *
                                816 * OUTPUT THE FINAL STATEMENT ADDRESS TABLE BUFFER TO DISK
                                817 *
    1E80 1C 03 1CFF F1          818 BTR130 MVC   BTRSHA,BTRSHE(BTRSEL,@BR) SET STMT TABLE MAXIMUM ENTRY
                                819 *
    1E85 C2 02 07DA             820        LA    B$SDPL,@XR                LOAD STATEMENT TABLE DPL CADDR
    1E89 C0 87 1A6B             821        B     B$DL4T                    LINK TO WRITE STMT TABLE BUFF
                                822 *
    1E8D C0 87 0025             823        B     $DISKN                    LINK TO WAIT OUTPUT COMPLETED
    1E91 057F             1E92  824        DC    AL(@CADDR)($WAITF)        CADDR OF DISK IOCR 'WAIT' DPL

                                826 ************************************************************************
                                827 * ESTABLISH CRITICAL COMPILER-GENERATED VIRTUAL ADDRESSES FOR LOADER
                                828 ************************************************************************
                                829 *
                                830 * CLEAR THE VIRTUAL MEMORY REGION INDICATOR AREAS
                                831 *
    1E93 0F 07 1A07 1A07        832 BTR150 SLC   B$LDRP+B@DL04,B$LDRP+B@DL04(4*@VADDR)  CLEAR REGION ADDRS
                                833 *
                                834 * ESTABLISH VIRTUAL MEMORY REGION-1 BEGINNING ADDRESS
                                835 *
    1E99 0C 00 1A00 0A42        836 BTR160 MVC   B$LDRP+B@DL01-1,B$PVAD-1(@VADDR-1)  SET UP PAGE AFTER PMC
                                837 *
                                838 * ESTABLISH VIRTUAL MEMORY REGION-1 ENDING ADDRESS
                                839 *
    1E9F 0C 00 1A02 0A35        840 BTR170 MVC   B$LDRP+B@DL02-1,B$PCPG(@VADDR-1)  SET UP LOW CONSTANT PAGE
                                841 *
                                842 * ESTABLISH VIRTUAL MEMORY REGION-2 BEGINNING ADDRESS
                                843 *
    1EA5 1E 01 0E46 E9          844 BTR180 ALC   B$SVRB,BTRVBA(@VADDR,@BR)  ADJUST VARIABLE BASE VADDR
                                845 *                                       * TO INDICATE 1ST FREE PAGE
    1EAA 0C 00 1A04 0E45        846        MVC   B$LDRP+B@DL03-1,B$SVRB-1(@VADDR-1)  SET UP PAGE AFTER VARS
                                847 *
                                848 * ESTABLISH VIRTUAL MEMORY REGION-2 ENDING ADDRESS
                                849 *
    1EB0 0C 00 1A06 0E47        850 BTR190 MVC   B$LDRP+B@DL04-1,B$SFAB-1(@VADDR-1)  SET UP LOW NAT PAGE
                                851 *
                                852 * ESTABLISH VIRTUAL ADDRESSES FOR SYSTEM INTERNAL ELEMENTS
                                853 *
    1EB6 0C 01 1A09 15AC        854 BTR200 MVC   B$LDRP+B@DL05,B$FAIS(@VADDR) SET UP 1ST CONSTANT VADDR
    1EBC 0C 01 1A0B 15A0        855        MVC   B$LDRP+B@DL06,B$FAIW(@VADDR) SET UP 1ST VARIABLE VADDR

                                857 ***********************************************************************
                                858 * TERMINATOR 2ND SEGMENT CALLING SEQUENCE ROUTINE
                                859 ***********************************************************************
                                860 *
                                861 * TEST WHETHER CURRENT SEGMENT WAS DISK OR CORE RESIDENT
                                862 *
    1EC2 5D 01 FB EB            863 BTR250 CLC   BTRCA2(,@BR),BTRPBA(@CADDR,@BR)  IF CURR SEG CAME FR DISK
    1EC6 F2 81 10               864        JE    BTR280                    * GO LOAD & EXEC 2ND SEG
                                865 *
                                866 * CURRENT SEGMENT WAS CORE RESIDENT - TEST WHETHER 2ND SEGMENT HAS
                                867 * ALSO BEEN LOADED INTO CORE
                                868 *
    1EC9 4E 00 FD 043B          869 BTR260 ALC   BTRFCP-1(,@BR),$EXFTR(1)  CALC MAX PROCESSOR CORE PAGE
    1ECE 5D 01 FB FE            870        CLC   BTRCA2(,@BR),BTRFCP(@CADDR,@BR)  IF 2ND SEGMENT IN CORE
    1ED2 F2 82 0B               871        JL    BTR290                    * GO SET TO EXEC 2ND SEG
                                872 *
                                873 * 2ND SEGMENT IS DISK RESIDENT - ESTABLISH DISTRIBUTOR PARAMETERS FOR
                                874 * CORELOADING AND EXECUTING THE 2ND SEGMENT
                                875 *
    1ED5 5C 01 FB EB            876 BTR270 MVC   BTRCA2(,@BR),BTRPBA(@CADDR,@BR)  SET UP DISKLOAD CADDR
                                877 *
                                878 * EXIT TO DISTRIBUTOR FOR 2ND SEGMENT CORELOAD AND EXECUTION
                                879 *
    1ED9 D2 02 FA               880 BTR280 LA    BTRAD2(,@BR),@XR          LOAD DISTRIBUTOR PARM CADDR
    1EDC C0 87 073A             881        B     B$DST2                    GO LOAD & EXECUTE 2ND SEGMENT
                                882 *
                                883 * 2ND SEGMENT IS CORE RESIDENT - BRANCH TO NEXT CONSECUTIVE CORE PAGE
                                884 * AND CONTINUE TERMINATOR EXECUTION
                                885 *
    1EE0 76 01 E7               886 BTR290 A     BTRBLS(,@BR),@BR          SET 2ND SEGMENT BASE CORE ADDR
    1EE3 D0 87 00               887        B     BTRSG2(,@BR)              GO EXECUTE THE 2ND SEGMENT

                                889 ***********************************************************************
                                890 * COMPILER TERMINATOR SEGMENT-1 CONSTANTS
                                891 ***********************************************************************
                                892 *
    1EE6 0100             1EE7  893 BTRBLS DC    AL(@CADDR)(B@BLSZ)        LENGTH OF CORE BLOCK OR PAGE
    1EE8 00FF             1EE9  894 BTRVBA DC    AL(@VADDR)(B@BLSZ-1)      REGION-2 VIRTUAL ADDR ADJUSTER
    1EEA 0600             1EEB  895 BTRPBA DC    AL(@CADDR)(B$CSBF)        PROCESSSR DISK BUFFER CADDR
                                896 *
    1EEC 1B0E             1EED  897 BTRFTA DC    AL(@CADDR)(B$FORT)        CADDR OF 1ST 'FOR' TABLE ENTRY
                                898 *
                          1CFF  899 BTRSHA EQU   B$SABF+B@BLSZ-1           CADDR OF STMNT TBL BFR RH BYTE
                          0004  900 BTRSEL EQU   @VADDR+B@LSNO             LENGTH OF A STATEMENT TBL ENTRY
    1EEE FFFFFFFF         1EF1  901 BTRSHE DC    XL(BTRSEL)'FFFFFFFF'      MAXIMUM ENTRY FOR STMNT TABLE

                                903 ***********************************************************************
                                904 * COMPILER TERMINATOR SEGMENT-1 DISK PARAMETER LIST
                                905 ***********************************************************************
                                906 *
                          1EF2  907 BTREPL EQU   *                         ERROR STACK CORELOAD DPL ADDR
    1EF2 01               1EF2  908 BTREFN DC    AL1(@DGET)                DISK IOCR 'READ' FUNCTION
    1EF3 07               1EF3  909 BTRECY DC    AL1(B@DVCY)               ERROR STACK BASE CYLINDER ADDR
    1EF4 56               1EF4  910 BTRESA DC    AL1(B@DVC1)               ERROR STACK 1ST LOGICAL SECTOR
    1EF5 03               1EF5  911 BTRESC DC    IL1'3'                    SECTOR COUNT FOR THE ERR STACK
    1EF6 1C00             1EF7  912 BTRECA DC    AL(@CADDR)($$ERSK)        ERROR STACK CORELOAD ADDRESS

                                914 ***********************************************************************
                                915 * COMPILER TERMINATOR PSEUDO INSTRUCTION SEQUENCE
                                916 ***********************************************************************
                                917 *
                          1EF8  918 BTRPCA EQU   *                         CADDR OF ENDING PMC SEQUENCE
                                919 *
    1EF8 02               1EF8  920 BTRSVC DC    AL(B@LCOP)(B@CSVC)        'SUPERVISOR CALL' PSEUDO OPCODE
    1EF9 70               1EF9  921 BTREOF DC    AL(B@LCOP)(B@CEOF)        'END-OF-FILE' PSEUDO OPCODE

                                923 ***********************************************************************
                                924 * COMPILER TERMINATOR SEGMENT-1 MORK AREAS
                                925 ***********************************************************************
                                926 *
                          1EFA  927 BTRAD2 EQU *                           DISTR PARMS FOR SEG-2 EXEC
    1EFA                  1EFB  928 BTRCA2 DS    CL(@CADDR)                TERMINATOR SEGMENT CORE ADDRESS
    1EFC 5C               1EFC  929 BTRSA2 DC    AL1(B@DEND+BTRPSI)        BTRMNT SEG-2 PHYS SECTOR ADDR
                                930 *
    1EFD                  1EFE  931 BTRFCP DS    CL(@CADDR)                FINAL AVAILABLE CORE PAGE ADDR
    1EFD                        932        ORG   *-@CADDR                  INITIALIZE CORE PAGE ADDR TO
    1EFD 1F00             1EFE  933        DC    AL(@CADDR)(B$CSXA-B@BLSZ) * FINAL PAGE BEFORE EXTENSION

                                935 ***********************************************************************
                                936 * COMPILER TERMINATOR SECOND SEGMENT
                                937 ***********************************************************************
                                938 *
                                939 * ESTABLISH TERMINATOR SEGMENT-2 ADDRESSABILITY
                                940 *
    1F00                        941        ORG   BTRMNT+B@BLSZ             BEGIN SEGMENT-2 AT PAGE BOUND
                          1F00  942        USING *,@BR                     DEFINE SEGMENT-2 BASE ADDRESS
                                943 *
                                944 * ESTABLISH LETTER VARIABLE SYMBOL TABLE FOR THE LOADER
                                945 *
    1F00 0C 39 1A45 109B        946 BTR300 MVC   B$LDRP+B@DL07,B$SLVT+B@LL07-1(B@LL07)  SET UP LTR VAR TBL
                                947 *
                                948 * ESTABLISH LETTER-DIGIT VARIABLE SYMBOL TABLE FOR THE LOADER
                                949 *
    1F06 0C FF 1B45 119B        950 BTR310 MVC   B$LDRP+B@DL08,B$SLDT+B@LL08-1(B@LL08)  SET UP LTR-
    1F0C 0C FF 1C45 129B        951        MVC   B$LDRP+B@DL09,B$SLDT+B@LL08+B@LL09-1(B@LL09)  * DIGIT TFIL
    1F12 0C 43 1C89 12DF        952        MVC   B$LDRP+B@DL10,B$SLDT+B@LL08+B@LL09+B@LL10-1(B@LL10)
                                953 *
                                954 * ESTABLISH CHARACTER VARIABLE SYMBOL TABLE FOR THE LOADER
                                955 *
    1F18 0C 39 1CC3 1319        956 BTR320 MVC   B$LDRP+B@DL11,B$SCVT+B@LL11-1(B@LL11)  SET UP CHAR VAR TBL
                                957 *
                                958 * CLEAR THE FUNCTION AND ARRAY TABLE AREA FOR THE LOADER
                                959 *
    1F1E 0F FF 1E71 1E71        960 BTR330 SLC   B$LDRP+B@DL15,B$LDRP+B@DL15(B@LL15)  INITLZ THE FUNC AND
    1F24 0F 95 1F07 1F07        961        SLC   B$LDRP+B@DL16,B$LDRP+B@DL16(B@LL16)  * ARRAY AREA TO ZEROS

                                963 ***********************************************************************
                                964 * ESTABLISH ARITHMETIC ARRAY SYMBOL TABLE AND DOPE VECTORS FOR LOADER
                                965 ***********************************************************************
                                966 *
                                967 * GET AN ENTRY FROM THE COMPILE-TIME ARITHMETIC (NUMERIC) ARRAY TABLE
                                968 *
    1F2A 75 02 CA               969 BTR350 L     BTRCNP(,@BR),@XR          LOAD COMPILE-TIME NAT POINTER
    1F2D 6C 05 C8 05            970        MVC   BTRCNE(,@BR),@VADDR+B@ACD2(B@LCNA,@XR) SAVE THE NAT ENTRY
                                971 *
                                972 * ESTABLISH A LOADER-TIME NUMERIC ARRAY TABLE VIRTUAL ADDRESS ENTRY
                                973 *
    1F31 C2 02 1CC3             974 BTR360 LA    B$LDRP+B@DL11,@XR         LOAD LOADER-TIME NAT BASE ADDR
                                975 *
    1F35 9C 01 00 C4            976 BTR370 MVC   *-*(,@XR),BTRVAD(@VADDR,@BR) HOVE THE ARRAY VADDR INTO
    1F37                        977        ORG   BTR370+@D1                * LOADER-TIME NAT ENTRY
    1F37 3A                 1F37  978        DC    AL1(B@LL12)               INITIALIZE LOADER-TIME NAT
    1F39                        979        ORG   BTR370+@INST4             * POINTER TO RIGHTMOST ENTRY
                                980 *
                                981 * TEST WHETHER CURRENT ENTRY ARRAY WAS REFERENCED IN PROGRAM
                                982 *
    1F39 7D 56 C3               983 BTR380 CLI   BTRVAD-1(,@BR),B@DVC1     IF ARRAY WAS NOT REFERENCED
    1F3C F2 82 0A               984        JL    BTR400                    * SKIP PAST FAT PROCESSING
                                985 *
                                986 * ESTABLISH A FUNCTION AND ARRAY TABLE DOPE VECTOR FOR CURRENT ENTRY
                                987 *
    1F3F 75 02 C4               988 BTR390 L     BTRVAD(,@BR),@XR          LOAD THE ARRAY VIRTUAL ADDRESS
    1F42 76 02 B5               989        A     BTRFAC(,@BR),@XR          CONVERT THE VADDR TO A CADDR
    1F45 9C 03 03 C8            990        MVC   B@ACD2(,@XR),BTRCND(2*B@LDMN,@BR)  SET DOPE VECTOR DIMENS
                                991 *
                                992 * DECREMENT TABLE POINTERS AND TEST FOR MORE ENTRIES TO PROCESS
                                993 *
    1F49 5F 01 CA B7            994 BTR400 SLC   BTRCNP(,@BR),BTRCNL(@CADDR,@BR)  DECR COMPILE-TIME NAT PT
    1F4D 5F 00 37 BC            995        SLC   BTRNTP(,@BR),BTRSTL(1,@BR)  DECR LOADER-TIME NAT PT
    1F51 D0 84 2A               996        BH    BTR350(,@BR)              IF MORE NAT ENTRIES, GO PROCESS

                                998 ***********************************************************************
                                999 * ESTABLISH CHARACTER ARRAY SYMBOL TABLE AND DOPE VECTORS FOR LOADER
                               1000 ***********************************************************************
                               1001 *
                               1002 * GET AN ENTRY FROM THE COMPILE-TIME CHARACTER ARRAY TABLE
                               1003 *
    1F54 75 02 CC              1004 BTR410 L     BTRCCP(,@BR),@XR          LOAD COMPILE-TIME CAT POINTER
    1F57 6C 03 C6 03           1005        MVC   BTRCCE(,@BR),@VADDR+B@CDMN(B@LCCA,@XR) SAVE THE CAT ENTRY
                               1006 *
                               1007 * ESTABLISH A LOADER-TIME CHARACTER ARRAY TABLE VIRTUAL ADDRESS ENTRY
                               1008 *
    1F5B C2 02 1CFD            1009 BTR420 LA    B$LDRP+B@DL12,@XR         LOAD LOADER-TIME CAT BASE ADDR
                               1010 *
    1F5F 9C 01 00 C4           1011 BTR430 MVC   *-*(,@XR),BTRVAD(@VADDR,@BR)  MOVE THE ARRAY VADDR INTO
    1F61                       1012        ORG   BTR430+@D1                * LOADER-TIME CAT ENTRY
    1F61 3A                 1F61 1013        DC    AL1(B@LL13)               INITIALIZE LOADER-TIME CAT
    1F63                       1014        ORG   BTR430+@INST4   CHECK OBJ * POINTER TO RIGHTMOST ENTRY
                               1015 *
                               1016 * TEST WHETHER CURRENT ENTRY ARRAY WAS REFERENCED IN PROGRAM
                               1017 *
    1F63 7D 56 C3              1018 BTR440 CLI   BTRVAD-1(,@BR),B@DVC1     IF ARRAY WAS NOT REFERENCED
    1F66 F2 82 0A              1019        JL    BTR460                    * SKIP PAST FAT PROCESSING
                               1020 *
                               1021 * ESTABLISH A FUNCTION AND ARRAY TABLE DOPE VECTOR FOR CURRENT ENTRY
                               1022 *
    1F69 75 02 C4              1023 BTR450 L     BTRVAD(,@BR),@XR          LOAD THE ARRAY VIRTUAL ADDRESS
    1F6C 76 02 B5              1024        A     BTRFAC(,@BR),@XR          CONVERT THE VADDR TO A CADDR
    1F6F 9C 01 01 C6           1025        MVC   B@CDMN(,@XR),BTRCCD(B@LDMN,@BR)  SET DOPE VECTOR DIMENSION
                               1026 *
                               1027 * DECREMENT TABLE POINTERS AND TEST FOR MORE ENTRIES TO PROCESS
                               1028 *
    1F73 5F 01 CC B9           1029 BTR460 SLC   BTRCCP(,@BR),BTRCCL(@CADDR,@BR)  DECR COMPILE-TIME CAT PT
    1F77 5F 00 61 BC           1030        SLC   BTRCTP(,@BR),BTRSTL(1,@BR)  DECR LOADER-TIME CAT PT
    1F7B D0 84 54              1031        BH    BTR410(,@BR)              IF MORE CAT ENTRIES, GO PROCESS

                               1033 ***********************************************************************
                               1034 * ESTABLISH USER FUNCTION SYMBOL TABLE AND ADDRESSES FOR LOADER
                               1035 ***********************************************************************
                               1036 *
                               1037 * GET AN ENTRY FROM THE COMPILE-TIME USER FUNCTION TABLE
                               1038 *
    1F7E 75 02 CE              1039 BTR470 L     BTRCFP(,@BR),@XR          LOAD COMPILE-TIME FNT POINTER
    1F81 6C 03 C6 03           1040        MVC   BTRCFE(,@BR),@VADDR+B@FVAD(B@LCFN,@XR)  SAVE THE FNT ENTRY
                               1041 *
                               1042 * ESTABLISH A LOADER-TIME USER FUNCTION TABLE VIRTUAL ADDRESS ENTRY
                               1043 *
    1F85 C2 02 1D37            1044 BTR480 LA    B$LDRP+B@DL13,@XR         LOAD LOADER-TIME FNT BASE ADDR
                               1045 *
    1F89 9C 01 00 C4           1046 BTR490 MVC   *-*(,@XR),BTRVAD(@VADDR,@BR) MOVE THE FUNCTION VADDR
    1F8B                       1047        ORG   BTR490+@D1                * INTO LOADER-TIME FNT ENTRY
    1F8B 3A               1F8B 1048        DC    AL1(B@LL14)               INITIALIZE LOADER-TIME FNT
    1F8D                       1049        ORG   BTR490+@INST4             * POINTER TO RIGHTMOST ENTRY
                               1050 *
                               1051 * TEST WHETHER CURRENT ENTRY FUNCTION WAS REFERENCED IN PROGRAM
                               1052 *
    1F8D 7D 56 C3              1053 BTR500 CLI   BTRVAD-1(,@BR),B@DVC1     IF FUNCTION WAS NOT REFERENCED
    1F90 F2 82 0A              1054        JL    BTR520                    * SKIP PAST FAT PROCESSING
                               1055 *
                               1056 * ESTABLISH A FUNCTION AND ARRAY TABLE ADDRESS FOR CURRENT ENTRY
                               1057 *
    1F93 75 02 C4              1058 BTR510 L     BTRVAD(,@BR),@XR          LOAD THE FUNCTION VIRTUAL ADDR
    1F96 76 02 B5              1059        A     BTRFAC(,@BR),@XR          CONVERT THE VADDR TO A CADDR
    1F99 9C 01 01 C6           1060        MVC   B@FVAD(,@XR),BTRCFA(@VADDR,@BR)  SET FUNCTION VIRTUAL ADDR
                               1061 *
                               1062 * DECREMENT TABLE POINTERS AND TEST FOR MORE ENTRIES TO PROCESS
                               1063 *
    1F9D 5F 01 CE BB           1064 BTR520 SLC   BTRCFP(,@BR),BTRCFL(@CADDR,@BR)  DECR COMPILE-TIME FNT PT
    1FA1 5F 00 8B BC           1065        SLC   BTRFTP(,@BR),BTRSTL(1,@BR)  DECR LOADER-TIME FNT PT
    1FA5 D0 84 7E              1066        BH    BTR470(,@BR)              IF MORE FNT ENTRIES, GO PROCESS
```

```
                               1068 ***********************************************************************
                               1069 * NORMAL COMPILER EXIT ROUTINE
                               1070 ***********************************************************************
                               1071 *
                               1072 * LOAD AND TRANSFER CONTROL TO THE BASIC LOADER
                               1073 *
    1FA8 D2 02 BD              1074 BTR600 LA    BTRDPL(,@BR),@XR          STORE LOADER CORELOAD DPL ADDR
    1FAB 74 02 B3              1075        ST    BTRDPA(,@BR),@XR          * FOR SYSTEM LOADER PARAMETER
    1FAE C0 87 051E            1076        B     $RLOAD                    EXIT THE COMPILER
    1FB2                  1FB3 1077 BTRDPA DS    CL(@CADDR)                LOADER CORELOAD DPL ADDRESS

                               1079 ***********************************************************************
                               1080 * COMPILER TERMINATOR SEGMENT-2 CONSTANTS
                               1081 ***********************************************************************
                               1082 *
    1FB4 1F08             1FB5 1083 BTRFAC DC    AL(@CADDR)(B$LDRP+B@DL16+1) FUNC & ARRAY ADDR CONVERTER
                               1084 *
    1FB6 0006             1FB7 1085 BTRCNL DC    AL(@CADDR)(B@LCNA)        COMPILE-TIME NAT ENTRY LENGTH
    1FB8 0004             1FB9 1086 BTRCCL DC    AL(@CADDR)(B@LCCA)        COMPILE-TIME CAT ENTRY LENGTH
    1FBA 0004             1FBB 1087 BTRCFL DC    AL(@CADDR)(B@LCFN)        COMPILE-TIME FNT ENTRY LENGTH
                               1088 *
    1FBC 02               1FBC 1089 BTRSTL DC    AL1(@VADDR)               LOADER-TIME SYM TBL ENTRY LNG

                               1091 ***********************************************************************
                               1092 * COMPILER TERMINATOR SEGMENT-2 DISK PARAMETER LIST
                               1093 ***********************************************************************
                               1094 *
                               1095 *TRDPL $DPL  FUNC-@DGET,DADDR-#$LOAD,CNT-#$@LOA,CADDR-#$$LOA
                          1FBD 1096+BTRDPL EQU   *                         DISK PARAMETER LIST
    1FBD 01               1FBD 1097+       DC    AL1(@DGET)                REQUESTED FUNCTION
    1FBE 0100             1FBF 1098+       DC    AL2(#$LOAD)               DISK ADDRESS
    1FC0 13               1FC0 1099+       DC    AL1(#$@LOA)               SECTOR COUNT
    1FC1 0600             1FC2 1100+       DC    AL2(#$$LOA)               BUFFER ADDRESS
                               1101+*** END OF EXPANSION ***

                               1103 ***********************************************************************
                               1104 * COMPILER TERMINATOR SEGMENT-2 WORK AREAS
                               1105 ***********************************************************************
                               1106 *
                          1FC3 1107 BTRTEN EQU   *                         COMPILE-TIME FUNCTION & ARRAY
    1FC3                  1FC8 1108        DS    CL(B@LCNA)                * SYMBOL TABLES ENTRY SAVE AREA
                               1109 *
    1FC9                  1FCA 1110 BTRCNP DS    CL(@CADDR)                COMPILE-TIME NAT POINTER -
    1FC9                       1111        ORG   *-@CADDR                  * INITLZ TO THE
    1FC9 13C2             1FCA 1112        DC    AL(@CADDR)(B$SNAT+B@NAAR*B@LCNA-B@LCNA) * RIGHTMOST ENTRY
                               1113 *
    1FCB                  1FCC 1114 BTRCCP DS    CL(@CADDR)                COMPILE-TIME CAT POINTER -
    1FCB                       1115        ORG   *-@CADDR                  * INITLZ TO THE
    1FCB 1438             1FCC 1116        DC    AL(@CADDR)(B$SCAT+B@NCAR*B@LCCA-B@LCCA) * RIGHTMOST ENTRY
                               1117 *
    1FCD                  1FCE 1118 BTRCFP DS    CL(@CADDR)                COMPILE-TIME FNT POINTER -
    1FCD                       1119        ORG   *-@CADDR                  * INITLZ TO THE
    1FCD 14AC             1FCE 1120        DC    AL(@CADDR)(B$SFNT+B@NUFN*B@LCFN-B@LCFN) * RIGHTMOST ENTRY

                               1122 ***********************************************************************
                               1123 * COMPILER TERMINATOR EQUATES REFERENCING CONSTANTS
                               1124 ***********************************************************************
                               1125 *
                          0000 1126 BTRSG2 EQU   0                         DISP FOR BTRMNT SEG-2 ENTRY PT
                          0004 1127 BTRPSI EQU   X'04'                     PHYSICAL SECTOR ADDR INCREMENT
                          00FF 1128 BTRBND EQU   B@BLSZ-1                  DISP INDICATING EMPTY CON BFR

                               1130 ***********************************************************************
                               1131 * COMPILER TERMINATOR EQUATES REFERENCING PROGRAM LABELS
                               1132 ***********************************************************************
                               1133 *
                          1FC4 1134 BTRVAD EQU   BTRTEN+@VADDR-1           COMPILE-TIME FIA SYMBOL VADDR
                          1FC8 1135 BTRCNE EQU   BTRTEN+@VADDR+B@ACD2      COMPILE-TIME NAT ENTRY ADDR
                          1FC8 1136 BTRCND EQU   BTRCNE                    COMPILE-TIME NAT ENTRY DINERS
                          1FC6 1137 BTRCCE EQU   BTRTEN+@VADDR+B@CDMN      COMPILE-TIME CAT ENTRY ADDR
                          1FC6 1138 BTRCCD EQU   BTRCCE                    COMPILE-TIME CAT ENTRY DIMEN
                          1FC6 1139 BTRCFE EQU   BTRTEN+@VADDR+B@FVAD      COMPILE-TIME FNT ENTRY ADDR
                          1FC6 1140 BTRCFA EQU   BTRCFE                    COMPILE-TIME FNT ENTRY VADDR
                               1141 *
                          1F37 1142 BTRNTP EQU   BTR370+@D1                LOADER-TIME NAT POINTER DISP
                          1F61 1143 BTRCTP EQU   BTR430+@D1                LOADER-TIME CAT POINTER DISP
                          1F8B 1144 BTRFTP EQU   BTR490+@D1                LOADER-TIME FNT POINTER DISP
                               1145 *
                               1146 ***********************************************************************
                               1147 *
                               1148 * END OF COMPILER TERMINATOR CODING
                               1149 *
                               1151 ***********************************************************************
                               1152 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               1153 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               1154 *                                                                     *
                               1155 ***********************************************************************
                               1156 *STATUS                                                               *
                               1157 *   VERSION 1 MODIFICATION 0                                          *
                               1158 *                                                                     *
                               1159 *FUNCTION                                                             *
                               1160 *   BKRTRN IS EXECUTED TO TRANSLATE RETURN STATEMENTS AS THEY OCCUR   *
                               1161 *   IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE   *
                               1162 *   THE PSEUDOCODE IN VIRTUAL MEMORY.                                 *
                               1163 *                                                                     *
                               1164 *ENTRY POINTS                                                         *
                               1165 *   BKRTRN HAS OILY ONE ENTRY POINT:                                  *
                               1166 *        BKRTRN - TRANSLATE RETURN STATEMENT                          *
                               1167 *   THE FORMAT OF THE CALLING SEQUENCE:                               *
                               1168 *        B     BKRTRN                                                 *
                               1169 *                                                                     *
                               1170 * INPUT                                                               *
                               1171 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               1172 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER OF THE       *
                               1173 *     LEADING KEYWORD, RETURN.                                        *
                               1174 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                               1175 *     CHARACTER IN THE LEADING KEYWORD, RETURN.                       *
                               1176 *                                                                     *
                               1177 *OUTPUT                                                               *
                               1178 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               1179 *     GENERATED BY BKRTRN IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               1180 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               1181 *     SEQUENCES.                                                      *
                               1182 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               1183 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               1184 *                                                                     *
                               1185 *EXTERNAL REFERENCES                                                  *
                               1186 *        B$PUTC - (B$PCAD, B$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY *
                               1187 *                 OUTPUT ROUTINE.                                     *
                               1188 *        B$RMRK - ENTRY TO BASIC COMPILER REMARK ROUTINE.             *
                               1189 *                                                                     *
                               1190 *EXITS, NORMAL                                                        *
                               1191 *   B$RMRK - ENTRY TO BASIC COMPILER REMARK ROUTINE.                  *
                               1192 *                                                                     *
                               1193 *EXITS, ERROR                                                         *
                               1194 *   N/A                                                               *
                               1195 *                                                                     *
                               1196 *TABLES/WORK AREAS                                                    *
                               1197 *   N/A                                                               *
                               1198 *                                                                     *
                               1199 *ATTRIBUTES                                                           *
                               1200 *   BKRTRN IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               1201 *                                                                     *
                               1202 *CHARACTER CODE DEPENDENCY                                            *
                               1203 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               1204 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               1205 *                                                                     *
                               1206 *NOTES                                                                *
                               1207 *   ERROR PROCEDURES                                                  *
                               1208 *      N/A                                                            *
                               1209 *                                                                     *
                               1210 *   REGISTER USAGE                                                    *
                               1211 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.   *
                               1212 *                                                                     *
                               1213 *   SAVED/RESTORED AREAS                                              *
                               1214 *      N/A                                                            *
                               1215 *                                                                     *
                               1216 *   MODIFICATION CONSIDERATIONS                                       *
                               1217 *      BKRTRN RESIDES ON THE SAME SECTOR WITH BTRMNT AND BPXRSR.   1-4*
                               1218 *      ANY MODIFICATION TO BKRTRN MUST CONSIDER THIS CO-RESIDENCY  1-4*
                               1219 *      SINCE IT WILL CHANGE THE ENTRY ADDRESS OF BPXRSR. THE       1-4*
                               1220 *      LIMITATION OF THE SECTOR BOUNDARY ON SIZE MUST ALSO BE      1-4*
                               1221 *      CONSIDERED.                                                 1-4*
                               1222 *                                                                     *
                               1223 *   REQUIRED MODULES                                                  *
                               1224 *        @NYSEQ - COMMON SYSTEM EQUATES.                              *
                               1225 *        @FXDEQ - SYSTEM NUCLEUS AND INDICATOR EQUATES.               *
                               1226 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.      *
                               1227 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               1228 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               1229 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               1230 *        $VSEQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               1231 *        $B$EQU - COMPILER FIXED EQUATES.                             *
                               1232 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               1233 *                                                                     *
                               1234 *   OTHER                                                             *
                               1235 *      BKRTRN IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               1236 ***********************************************************************

                               1238 *
                               1239 * ENTER BKRTRN - 'RETURN' STATEMENT ROUTINE
                               1240 *
                          1FCF 1241 BKRTRN EQU   *                         BKRTRN ENTRY POINT
                               1242 *
                               1243 * GENERATE A 'BRS' INSTRUCTION IN VIRTUAL MEMORY
                               1244 *
    1FCF D2 02 E2              1245 BKR010 LA    BKRBRC(,@BR),@XR          LOAD CADDR OF 'BRS' INSTR
    1FD2 34 02 0A40            1246        ST    B$PCAD,@XR                SET PUT RTN FOR VADDR OF 'BRS'
    1FD6 3C 00 0A41            1247        MVI   B$PNBY,B@LBRS-1           SET PUT RTN FOR LENGTH OF 'BRS'
    1FDA C0 87 093A            1248        B     B$PUTC                    LINK TO GENERATE PMC
                               1249 *
                               1250 * RETURN CONTROL TO THE REM STATEMENT ROUTINE
                               1251 *
    1FDE C0 87 1AE6            1252 BKR020 B     B$RMRK                    RETURN TO REMARK STMT RTN
                               1253 *
                               1254 ***********************************************************************
                               1255 * 'RETURN' STATEMENT ROUTINE PMC AND STORAGE PARAMETERS
                               1256 ***********************************************************************
                               1257 *
    1FE2 4C               1FE2 1258 BKRBRC DC    AL(B@LCOP)(B@CBRS)        'BRS' INSTR OPCODE
                               1259 *
                               1260 ***********************************************************************
                               1261 *
                               1262 * END OF 'RETURN' STATEMENT ROUTINE CODING
```

```
       S/3 BASIC COMPILER -RETURN- ROUTINE
ERR LOC  OBJECT CODE      ADDR STMT SOURCE STATEMENT          VER 15, MOD 00  20/07/20  PAGE 170
                               1263 *
                               1265 ***********************************************************************
                               1266 *  5703-XM1 COPYRIGHT IBM CORP. 1970                                  *
                               1267 *           REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083      *
                               1268 *                                                                     *
                               1269 ***********************************************************************
                               1270 *STATUS                                                               *
                               1271 *   VERSION 1 MODIFICATION 0                                          *
                               1272 *                                                                     *
                               1273 *FUNCTION                                                             *
                               1274 *   BPXRSR IS EXECUTED TO TRANSLATE RESTORE STATEMENTS AS THEY OCCUR  *
                               1275 *   IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE   *
                               1276 *   THE PSEUDOCODE IN VIRTUAL MEMORY.                                 *
                               1277 *                                                                     *
                               1278 *ENTRY POINTS                                                         *
                               1279 *   BPXRSR HAS ONLY ONE ENTRY POINT:                                  *
                               1280 *        BPXRSR - TRANSLATE RESTORE STATEMENT                         *
                               1281 *   THE FORMAT OF THE CALLING SEQUENCE IS:                            *
                               1282 *        B     BPXRSR                                                 *
                               1283 *                                                                     *
                               1284 *INPUT                                                                *
                               1285 *   * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING  *
                               1286 *     THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER OF THE       *
                               1287 *     LEADING KEYWORD, RESTORE.                                       *
                               1288 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST *
                               1289 *     CHARACTER IN THE LEADING KEYWORD, RESTORE.                      *
                               1290 *                                                                     *
                               1291 *OUTPUT                                                               *
                               1292 *   * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE        *
                               1293 *     GENERATED BY BPXRSR IS STORED IN THE NEXT AVAILABLE VIRTUAL     *
                               1294 *     MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION         *
                               1295 *     SEQUENCES.                                                      *
                               1296 *   * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE       *
                               1297 *     CHARACTER WHICH TERMINATES THE STATEMENT.                       *
                               1298 *                                                                     *
                               1299 *EXTERNAL REFERENCES                                                  *
                               1300 *        B$PUTC - (B$PCAD, B$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY *
                               1301 *                 OUTPUT ROUTINE.                                     *
                               1302 *        B$RMRK - ENTRY TO BASIC COMPILER REMARK ROUTINE.             *
                               1303 *                                                                     *
                               1304 *EXITS, NORMAL                                                        *
                               1305 *   B$RMRK - ENTRY TO BASIC COMPILER REMARK ROUTINE.                  *
                               1306 *                                                                     *
                               1307 *EXITS, ERROR                                                         *
                               1308 *   N/A                                                               *
                               1309 *                                                                     *
                               1310 *TABLES/WORK AREAS                                                    *
                               1311 *   N/A                                                               *
                               1312 *                                                                     *
                               1313 *ATTRIBUTES                                                           *
                               1314 *   BPXRSR IS NATURALLY RELOCATABLE AND REUSABLE.                     *
                               1315 *                                                                     *
                               1316 *CHARACIER CODE DEPENDENCY                                            *
                               1317 *   THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR    *
                               1318 *   INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.            *
                               1319 *                                                                     *
                               1320 *NOTES                                                                *
                               1321 *   ERROR PROCEDURES                                  *
                               1322 *      N/A                                            *
                               1323 *                                                     *
                               1324 *   REGISTER USAGE                                    *
                               1325 *      BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.  *
                               1326 *                                                                     *
                               1327 *   SAVED/RESTORED AREAS                                              *
                               1328 *      N/A                                                            *
                               1329 *                                                                     *
                               1330 *   MODIFICATION CONSIDERATIONS                                       *
                               1331 *      BPXRSR RESIDES ON THE SAME SECTOR WITH BTRMNT AND BKRTRN.      *
                               1332 *      ANY MODIFICATION TO BPXRSR MUST TAKE INTO CONSIDERATION        *
                               1333 *      THIS CO-RESIDENCY ANY ALSO THE LIMITATION OF THE SECTOR        *
                               1334 *      BOUNDARY ON SIZE.                                              *
                               1335 *                                                                     *
                               1336 *   REQUIRED MODULES                                                  *
                               1337 *        @NYSEQ - COMMON SYSTEM EQUATES.                              *
                               1338 *        @FXDEQ - SYSTEM NUCLEUS AND INDICATOR EQUATES.               *
                               1339 *        @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.      *
                               1340 *        @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.                   *
                               1341 *        @SPFEQ - SYSTEM PROGRAM FILE EQUATES.                        *
                               1342 *        @ERMEQ - ERROR MESSAGE EQUATES.                              *
                               1343 *        $VSEQU - FIXED VIRTUAL ADDRESS EQUATES.                      *
                               1344 *        $B$EQU - COMPILER FIXED EQUATES.                             *
                               1345 *        $B@EQU - COMPILER SYSTEM EQUATES.                            *
                               1346 *                                                                     *
                               1347 *   OTHER                                                             *
                               1348 *      BPXRSR IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.      *
                               1349 ***********************************************************************

                               1351 *
                               1352 * ENTER BPXRSR - 'RESTORE' STMT ROUTINE
                               1353 *
                          1FE3 1354 BPXRSR EQU   *                         BPXRSR ENTRY POINT
                               1355 *
                               1356 * GENERATE AN 'RSR' INSTRUCTION PMC IN VIRTUAL MEMORY
                               1357 *
    1FE3 D2 02 F6              1358 BPX010 LA    BPXRSC(,@BR),@XR          LOAD CADDR OF 'RSR' INSTR
    1FE6 34 02 0A40            1359        ST    B$PCAD,@XR                SET PUT RTN VADDR FOR 'RSR'
    1FEA 3C 00 0A41            1360        MVI   B$PNBY,B@LRSR-1           SET PUT RTN LNG CODE FOR 'RSR'
    1FEE C0 87 093A            1361        B     B$PUTC                    LINK TO GENERATE 'RSR' PMC
                               1362 *
                               1363 * RETURN CONTROL TO THE REMARK ROUTINE
                               1364 *
    1FF2 C0 87 1AE6            1365 BPX020 B     B$RMRK
                               1366 *
                               1367 ***********************************************************************
                               1368 * 'RESTORE' STATEMENT ROUTINE PARAMETER AND STORAGE AREA
                               1369 ***********************************************************************
                               1370 *
    1FF6 5A               1FF6 1371 BPXRSC DC    AL(B@LCOP)(B@CRSR)        'RSR' INSTR OPCODE
                               1372 *
                               1373 ***********************************************************************
                               1374 *
                               1375 * END OF 'RESTORE' STATEMENT ROUTINE CODING
                               1376 *
                          FFFF 1377        END
TOTAL STATEMENTS IN ERROR IN THIS ASSEMBLY =     0
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 174
$$$CMD  001  0020 0659
$$$DAT  001  0040 0658
$$$EPL  001  0091 0655
$$$ERN  001  0080 0709
$$$FUN  001  0010 0660
$$$NLN  001  00A0 0705   0771  0808
$$$STD  001  0081 0654
$$BNLN  001  0605 0635   0637
$$CDBS  001  08C0 0685
$$CDND  001  0666 0644
$$CDRD  001  0890 0683   0685
$$CKEY  001  0603 0633
$$CKFF  001  0B3D 0665
$$COFF  001  0B44 0664
$$CSNS  001  209C 0694
$$DATB  001  0BBF 0666
$$EOSA  001  0AFE 0663
$$ERSK  001  1C00 0704   0912
$$FITS  001  1D00 0712
$$FLIB  001  06FF 0711
$$ILEN  001  0601 0629   0631  0635
$$ILHD  001  0600 0627   0629
$$INLN  001  0607 0642   0644  0646
$$INND  001  06FA 0646
$$KBDT  001  09E1 0653   0657
$$KBSN  001  09E2 0657   0662
$$KLD1  001  0600 0717
$$KLD2  001  0700 0719
$$KLD3  001  0C00 0721
$$LPOS  001  09EB 0662
$$PCNT  001  07E9 0678
$$PLYN  001  2004 0692
$$PRES  001  0890 0651   0653  0663  0664  0665  0666  0683
$$PRFL  001  2143 0696
$$PRNT  001  0707 0672   0673  0677  0678
$$PRTN  001  0782 0673
$$PSIO  001  07CE 0677
$$PYCD  001  2200 0698
$$PYMP  001  2000 0690   0692  0694  0696  0698
$$SLIB  001  1C00 0707
$$TPCD  001  0606 0637   0642
$$UPAR  001  0602 0631   0633
$$WSPB  001  1E00 0710
$$XIND  001  06FF 0708   0711
$$ZERO  001  0000 0223   0224  0226  0227  0228  0232  0690
$ABORT  001  0010 0336
$BASIC  001  0080 0394
$BIGCD  001  0080 0470
$BLDPL  001  0579 0603   0605
$BLNOE  001  0569 0593
$BLOAD  001  0522 0584   0586  0589  0602  0603
$BLRTN  001  0550 0592   0593
$BRSAV  001  03C5 0281   0282
$BSADR  001  0587 0608   0610
$BUFPT  001  03E3 0489   0490
$CABLD  001  04B4 0562   0563
```

```
SYMBOL  LEN VALUE DEFN   REFERENCES

$CAERK  001  0469 0539   0542  0762  0773  0810
$CAERR  001  03CD 0287   0289  0772* 0809*
$CAIPL  001  049D 0558   0560
$CALLI  001  0008 0479
$CARDI  001  0001 0250
$CARPL  001  04A1 0560   0562
$CIENT  001  0483 0549   0550
$CIEXT  001  0480 0548   0549
$CIMSK  001  0476 0545   0548
$CISUS  001  0496 0553   0558
$CLBFR  001  0010 0437
$CMDKY  001  0008 0349
$CMODE  001  0002 0399
$CONFG  001  03DD 0462   0472
$CRPOS  001  03E2 0488   0489
$CRTAD  001  044D 0527   0528
$CRTAV  001  0002 0343
$CRTDN  001  0002 0367
$CRTIN  001  03D3 0364   0371
$CRTNO  001  0004 0346
$CRTPU  001  0004 0368
$CRTSP  001  0008 0369
$CRTUP  001  0001 0366
$CRUSH  001  0080 0475
$CSDPL  001  050E 0574   0575
$C0001  001  0464 0531   0537
$DATE   001  043A 0512   0513
$DBGUF  001  03E0 0474   0483
$DBLOK  001  0001 0424
$DFDET  001  03E8 0495   0496
$DISKN  001  0025 0226   0823
$DKERR  001  0008 0405
$DKSIZ  001  03D7 0449   0457  0498
$DK100  001  0001 0451
$DK200  001  0002 0452
$DK400  001  0004 0453
$DK600  001  0008 0454
$DK800  001  0010 0455
$DPLSV  001  0449 0523   0525
$DTNMB  001  0040 0270
$DTRDR  001  0040 0358
$ENDNU  001  0600 0617   0627  0651  0672  0708  0717  0719  0721  1196
$ERDPL  001  046F 0542   0544
$ERFIL  001  0040 0297
$ERHRD  001  0004 0429
$ERKEY  001  0080 0301
$ERLOG  001  0345 0231
$ERMAD  001  0472 0544   0545
$ERPND  001  0004 0402
$ERRCT  001  03CF 0303   0761*
$ERRPG  001  03CE 0291   0760* 0771* 0808*
$ERSFL  001  0035 0296
$ERSTK  001  0030 0294   0760
$ER050  001  0363 0232
$ER1N2  001  0050 0299
$EXADR  001  0517 0577   0579
```

```
$EXCMD  001  0001 0331
$EXFTR  001  043B 0513   0518  3384  3984  4514  5128  0869
$FCIND  001  0010 0409
$FDIND  001  0040 0416
$FEARR  001  0004 0224
$FEMAP  001  0588 0610   0611
$FILIB  001  03DA 0460   0461
$FITIN  001  0010 0385
$FUIND  001  0020 0414
$GUFIO  001  0583 0607   0608
$GUFIR  001  0008 0259
$HISTE  001  042E 0510   0511
$HIST1  001  0435 0511   0512
$HRDER  001  0020 0355
$INDR1  001  03D4 0371   0397
$INDR2  001  03D5 0397   0422
$INDR3  001  03D6 0422   0449
$INLNO  001  03CF 0289   0291  0303  0310  5614  5620*
$INRPT  001  0020 0267
$IOIND  001  03D2 0338   0364
$IOPGS  001  0010 0478
$IOYES  001  0002 0253
$IPLDV  001  05FF 0614   0617
$IRKEY  001  0020 0477
$KEYBD  001  03E1 0483   0488
$KEYCD  001  03C3 0247   0281
$KEYDT  001  0040 0391
$KE090  001  00DE 0227
$KE130  001  01D5 0228
$KYBSY  001  0010 0264
$LDRTN  001  0571 0602
$LEVEL  001  03DF 0472   0474
$LIST   001  0002 0426
$LMRGN  001  03C1 0242   0244
$LNPTR  001  0080 0361
$LOADB  001  054A 0586
$LOADR  001  051A 0579   0582
$LPRIO  001  03EA 0496
$LPROS  001  03E5 0491   0493
$LPRP3  001  03E4 0490   0491
$MOUNT  001  0020 0440
$MPDWN  001  0001 0340
$NEXTB  001  03E6 0493   0494
$NEXTL  001  03E7 0494   0495
$NOENB  001  0008 0432
$NOLST  001  0004 0256
$NUCBS  001  03C0 0239   0240
$NWRKF  001  0080 0445
$NWRKR  001  0040 0442
$PASWD  001  042D 0509   0510
$PAUSD  001  04BA 0563   0565
$PAUSE  001  0002 0333
$PGMDT  001  0020 0388
$PGMST  001  0010 0352
$PKERT  001  0419 0507   0509
$PLST1  001  0454 0528   0529
```

```
$PLST2  001  045B 0529   0530
$PLST3  001  0462 0530   0531
$PRDEV  001  044B 0525   0527
$PRESN  001  0002 0376
$PROCI  001  0001 0373
$PRPOS  001  03C2 0244   0247
$PSDBR  001  04FA 0568
$PSDXR  001  04F2 0567   0568
$PSTEP  001  0004 0334
$PSTMT  001  0008 0335
$PTCH1  001  03F5 0498   0502
$READY  001  0080 0418
$REORD  001  0040 0476
$RLOAD  001  051E 0582   0584  1076
$RMRGN  001  03C0 0240   0242
$RSTR   001  04D6 0565   0567  0569  0574
$RUNIT  001  0001 0312
$SFAID  001  050D 0570
$SPRNT  001  0465 0537   0539
$SRTRN  001  04FE 0569   0570
$STEPT  001  0002 0313
$SWPCR  001  0511 0575   0577
$TABLN  001  03CB 0284   0287
$TFLOW  001  0008 0319
$TRACE  001  0004 0314
$TRALL  001  0010 0320
$TROVR  001  054E 0589   0592
$TRUNK  001  0080 0272
$TRVAR  001  0020 0321
$UNMSK  001  048D 0550   0553
$USRDR  001  03DC 0461   0462
$VMDEF  001  0080 0325
$VOLF1  001  03FE 0504   0505
$VOLF2  001  040E 0506
$VOLID  001  03F6 0502   0503  0507
$VOLR1  001  03F6 0503   0504
$VOLR2  001  0406 0505   0506
$WAITF  001  057F 0605   0607  0824
$WFDEF  001  0040 0519
$WFLOK  001  0008 0382
$WFNME  001  0443 0518   0523
$WSIND  001  0004 0379
$XIND1  001  03D0 0310   0329  5944  6859  0759*
$XIND2  001  03D1 0329   0338
$XIND3  001  03D8 0457   0460
$XPREC  001  0040 0322   5944  6859
$XRSAV  001  03C7 0282   0284
$ZTRAD  001  05A2 0611
$12K    001  0004 0466
$16CKY  001  0008 0468
$16K    001  0002 0465
$22IMP  001  0001 0463
#$$#BL  001  0000 1049
#$$#CK  001  0000 1177
#$$#CN  001  0000 1145
#$$#CO  001  0000 0937
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 178
#$$#CS  001  0000 0997
#$$#DR  001  0000 0741
#$$#ER  001  0000 0941
#$$#FS  001  0000 1037
#$$#IN  001  0000 1181
#$$#PW  001  0000 1185
#$$#RS  001  0000 1017
#$$#SA  001  0000 1005
#$$#SS  001  0000 1001
#$$#VU  001  0600 0961
#$$#0T  001  0700 0733
#$$#1T  001  0000 0737
#$$BCO  001  0600 0749
#$$BOV  001  0800 1021   3145
#$$DPR  001  0700 0757
#$$DRE  001  0889 0773
#$$DSP  001  2800 0793
#$$ECM  001  0C00 1053
#$$EFK  001  0C00 1073
#$$ERR  001  0C00 1045
#$$EXM  001  0C00 0933
#$$FIL  001  0E00 1013
#$$FIS  001  0E00 1009
#$$FML  001  0200 1141
#$$FMS  001  0200 0981
#$$GRA  001  0889 0905
#$$GUF  001  0C00 1041
#$$INL  001  0600 1121
#$$INS  001  0600 0745
#$$KAL  001  0C00 0909
#$$KCA  001  0C00 1125
#$$KCH  001  0C00 0877
#$$KCN  001  0C00 0993
#$$KCT  001  0C00 0845
#$$KDE  001  0C00 0841
#$$KDI  001  0D00 0921
#$$KDN  001  0C00 0829
#$$KDO  001  0E00 0925
#$$KED  001  0C00 0765
#$$KEN  001  0C00 0769
#$$KEX  001  0C00 0789
#$$KGO  001  0C00 0761
#$$KHE  001  0C00 0945
#$$KKE  001  0C00 1173
#$$KLI  001  0C00 0849
#$$KLL  001  0920 1149
#$$KLO  001  0C00 0853
#$$KME  001  0D00 0833
#$$KMO  001  0C00 0777
#$$KNA  001  0C00 0889
#$$KOV  001  0E00 0809
#$$KPA  001  0C00 0785
#$$KPO  001  0C00 0873
#$$KPR  001  0C00 0897
#$$KRE  001  0C00 0817
#$$KRL  001  0700 0913
```

```
#$$KRM  001  0C00 0781
#$$KRN  001  0700 0801
#$$KRO  001  0D00 0805
#$$KRS  001  0C00 1129
#$$KRU  001  0C00 0825
#$$KRV  001  0800 0917
#$$KSA  001  0C00 0861
#$$KSE  001  0E00 0901
#$$KSO  001  0C20 0953
#$$KSS  001  0C00 0885
#$$KSV  001  0980 0881
#$$KSY  001  0C00 0893
#$$KWI  001  0C00 0821
#$$KWR  001  0C00 0813
#$$LOA  001  0600 0753   1100
#$$MIP  001  0C00 0949
#$$SDS  001  0C00 1061
#$$SFF  001  0E00 1065
#$$SFL  001  0F00 1057
#$$SFO  001  1500 1029
#$$SFS  001  0C00 1025
#$$SPA  001  0C00 0865
#$$SPO  001  0806 0869
#$$SPS  001  0C00 0857
#$$STR  001  1600 1033
#$$TDC  001  1000 0837
#$$TSY  001  1000 0797
#$$TVK  001  0FC0 0973
#$$UAL  001  0C00 0989
#$$UAT  001  0900 1085
#$$UCD  001  0900 1093
#$$UCN  001  0C00 1077
#$$UCP  001  0700 1081
#$$UDE  001  0C00 1097
#$$UDI  001  0C00 1101
#$$UEX  001  0C00 0985
#$$UIN  001  0C00 1089
#$$UPA  001  0C00 1069
#$$UPO  001  0C00 1137
#$$UPT  001  0C00 1133
#$$VCR  001  2000 0929
#$$VLO  001  0600 0965
#$$VOD  001  0600 0969
#$$VVM  001  0000 0977
#$$VXI  001  0600 0957
#$$ZDU  001  1100 1109
#$$ZLB  001  1100 1153
#$$ZLO  001  1100 1113
#$$ZLV  001  0F00 1169
#$$ZL1  001  0F00 1157
#$$ZL2  001  0F00 1161
#$$ZL3  001  0C00 1165
#$$ZTR  001  1000 1105
#$$ZUT  001  0C00 1117
#$#BLN  001  18D4 1048
#$#CKT  001  2118 1176
```

```
                                CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 180
#$#CNF  001  2000 1144
#$#COR  001  0800 0936
#$#CSA  001  1000 0996
#$#DRT  001  0000 0740
#$#ERM  001  0928 0940
#$#FSP  001  1880 1036
#$#INV  001  212C 1180
#$#PWR  001  2300 1184
#$#RSP  001  1780 1016
#$#SAV  001  1180 1004
#$#SSA  001  1128 1000
#$#VUF  001  0B08 0960
#$#0TR  001  0000 0732
#$#1TR  001  0080 0736
#$@#BL  001  0001 1050
#$@#CK  001  0004 1178
#$@#CN  001  0001 1146
#$@#CO  001  003A 0938
#$@#CS  001  003A 0998
#$@#DR  001  0008 0742
#$@#ER  001  0032 0942
#$@#FS  001  0030 1038
#$@#IN  001  003A 1182
#$@#PW  001  00C0 1186
#$@#RS  001  0030 1018
#$@#SA  001  0108 1006
#$@#SS  001  0001 1002
#$@#VU  001  0002 0962
#$@#0T  001  0018 0734
#$@#1T  001  0018 0738
#$@BCO  001  0018 0750
#$@BOV  001  0018 1022
#$@DPR  001  0005 0758
#$@DRE  001  0001 0774
#$@DSP  001  0004 0794
#$@ECM  001  0006 1054
#$@EFK  001  0002 1074
#$@ERR  001  0003 1046
#$@EXM  001  0003 0934
#$@FIL  001  0009 1014
#$@FIS  001  0009 1010
#$@FML  001  0052 1142
#$@FMS  001  0052 0982
#$@GRA  001  0003 0906
#$@GUF  001  0010 1042
#$@INL  001  0010 1122
#$@INS  001  0010 0746
#$@KAL  001  000F 0910
#$@KCA  001  000C 1126
#$@KCH  001  000C 0878
#$@KCN  001  0010 0994
#$@KCT  001  0009 0846
#$@KDE  001  0010 0842
#$@KDI  001  0005 0922
#$@KDN  001  0010 0830
#$@KDO  001  000C 0926
```

```
#$@KED  001  000E 0766
#$@KEN  001  0006 0770
#$@KEX  001  0003 0790
#$@KGO  001  0002 0762
#$@KHE  001  000C 0946
#$@KKE  001  0006 1174
#$@KLI  001  0011 0850
#$@KLL  001  0001 1150
#$@KLO  001  0008 0854
#$@KME  001  0003 0834
#$@KMO  001  0004 0778
#$@KNA  001  0008 0890
#$@KOV  001  0009 0810
#$@KPA  001  0005 0786
#$@KPO  001  000D 0874
#$@KPR  001  0009 0898
#$@KRE  001  0002 0818
#$@KRL  001  0004 0914
#$@KRM  001  0003 0782
#$@KRN  001  0003 0802
#$@KRO  001  000A 0806
#$@KRS  001  000A 1130
#$@KRU  001  0003 0826
#$@KRV  001  000D 0918
#$@KSA  001  0011 0862
#$@KSE  001  0004 0902
#$@KSO  001  000D 0954
#$@KSS  001  000B 0886
#$@KSV  001  0002 0882
#$@KSY  001  000F 0894
#$@KWI  001  0002 0822
#$@KWR  001  0002 0814
#$@LOA  001  0013 0754   1099
#$@MIP  001  000D 0950
#$@SDS  001  0004 1062
#$@SFF  001  0008 1066
#$@SFL  001  0005 1058
#$@SFO  001  0003 1030
#$@SFS  001  0011 1026
#$@SPA  001  0004 0866
#$@SPO  001  0003 0870
#$@SPS  001  0001 0858
#$@STR  001  0002 1034
#$@TDC  001  0003 0838
#$@TSY  001  0003 0798
#$@TVK  001  0001 0974
#$@UAL  001  0011 0990
#$@UAT  001  000C 1086
#$@UCD  001  000B 1094
#$@UCN  001  0009 1078
#$@UCP  001  000F 1082
#$@UDE  001  000E 1098
#$@UDI  001  0008 1102
#$@UEX  001  000E 0986
#$@UIN  001  000F 1090
#$@UPA  001  0004 1070
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 182
#$@UPO  001  0005 1138
#$@UPT  001  0012 1134
#$@VCR  001  0008 0930
#$@VLO  001  0002 0966
#$@VOD  001  0016 0970
#$@VVM  001  0030 0978
#$@VXI  001  0002 0958
#$@ZDU  001  0008 1110
#$@ZLB  001  0002 1154
#$@ZLO  001  000C 1114
#$@ZLV  001  0006 1170
#$@ZL1  001  0007 1158
#$@ZL2  001  000D 1162
#$@ZL3  001  000A 1166
#$@ZTR  001  0001 1106
#$@ZUT  001  0014 1118
#$BCOM  001  0080 0748
#$BOLV  001  1780 1020
#$DPRI  001  014C 0756
#$DREA  001  0200 0772
#$DSPL  001  0240 0792
#$ECMA  001  1900 1052
#$EFKE  001  1990 1072
#$ERRP  001  18C0 1044
#$EXMS  001  07D4 0932
#$FILN  001  1724 1012
#$FIST  001  1700 1008
#$FMLN  001  1E00 1140
#$FMST  001  0D00 0980
#$GRAP  001  0690 0904
#$GUFU  001  1880 1040
#$INLN  001  1C84 1120
#$INST  001  0020 0744
#$KALL  001  06A4 0908
#$KCAL  001  1CC4 1124
#$KCHA  001  053C 0876
#$KCND  001  0F80 0992
#$KCTL  001  03BC 0844
#$KDEL  001  035C 0840
#$KDIS  001  0744 0920
#$KDNT  001  0300 0828
#$KDOV  001  0780 0924
#$KEDI  001  0188 0764
#$KENA  001  01C4 0768
#$KEXT  001  0234 0788
#$KGOS  001  0180 0760
#$KHEL  001  0A30 0944
#$KKEY  001  2100 1172
#$KLIS  001  0400 0848
#$KLLA  001  2004 1148
#$KLOG  001  0444 0852
#$KMER  001  030C 0832
#$KMOU  001  0204 0776
#$KNAM  001  05C0 0888
#$KOVM  001  0290 0808
#$KPAS  001  0220 0784
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 183
#$KPOO  001  0508 0872
#$KPRT  001  063C 0896
#$KREA  001  02BC 0816
#$KRLA  001  0700 0912
#$KRMO  001  0214 0780
#$KRNU  001  0280 0800
#$KROV  001  028C 0804
#$KRSU  001  1D24 1128
#$KRUN  001  02CC 0824
#$KRVL  001  0710 0916
#$KSAV  001  0488 0860
#$KSET  001  0680 0900
#$KSOV  001  0AC8 0952
#$KSSP  001  0594 0884
#$KSVL  001  058C 0880
#$KSYM  001  0600 0892
#$KWID  001  02C4 0820
#$KWRI  001  02B4 0812
#$LOAD  001  0100 0752   1098
#$MIPP  001  0A80 0948
#$SDSY  001  192C 1060
#$SFFI  001  193C 1064
#$SFLO  001  1918 1056
#$SFOV  001  1844 1028
#$SFSY  001  1800 1024
#$SPAC  001  04CC 0864
#$SPOV  001  04DC 0868
#$SPSY  001  0484 0856
#$STRO  001  1850 1032
#$TDCK  001  0350 0836
#$TSYK  001  0250 0796
#$TVKB  001  0BAC 0972
#$UALL  001  0F00 0988
#$UATR  001  1A38 1084
#$UCDI  001  1AD8 1092
#$UCNF  001  19B8 1076
#$UCPL  001  19DC 1080
#$UDEL  001  1B24 1096
#$UDIS  001  1B5C 1100
#$UEXL  001  0EA8 0984
#$UINI  001  1A88 1088
#$UPAC  001  1980 1068
#$UPOV  001  1D24 1136
#$UPTF  001  1D5C 1132
#$VCRT  001  07B4 0928
#$VLOA  001  0B80 0964
#$VODK  001  0B88 0968
#$VVMR  001  0C00 0976
#$VXIT  001  0B00 0956
#$ZDUM  001  1BA4 1108
#$ZLBM  001  2008 1152
#$ZLOA  001  1BC4 1112
#$ZLVR  001  20B0 1168
#$ZL1M  001  2010 1156
#$ZL2M  001  2030 1160
#$ZL3M  001  2088 1164
```

```
#$ZTRA  001  1B9C 1104
#$ZUTM  001  1C14 1116
#BOVLY  001  0000 0001
@@E001  001  0000 2624   2626
@@E003  001  0001 2626   2628
@@E004  001  0002 2628   2630
@@E005  001  0003 2630   2632
@@E006  001  0004 2632   2634
@@E007  001  0005 2634   2636
@@E008  001  0006 2636   2638
@@E009  001  0007 2638   2640
@@E010  001  0008 2640   2642
@@E011  001  0009 2642   2644
@@E012  001  000A 2644   2646
@@E013  001  000B 2646   2648
@@E014  001  000C 2648   2650
@@E015  001  000D 2650   2652
@@E016  001  000E 2652   2654
@@E017  001  000F 2654   2656
@@E018  001  0010 2656   2658
@@E019  001  0011 2658   2660
@@E020  001  0012 2660   2662
@@E021  001  0013 2662   2664
@@E023  001  0014 2664   2666
@@E024  001  0015 2666   2668
@@E025  001  0016 2668   2670
@@E026  001  0017 2670   2672
@@E027  001  0018 2672   2674
@@E028  001  0019 2674   2676
@@E029  001  001A 2676   2678
@@E030  001  001B 2678   2680
@@E031  001  001C 2680   2682
@@E032  001  001D 2682   2684
@@E035  001  001E 2684   2686
@@E036  001  001F 2686   2688
@@E037  001  0020 2688   2690
@@E038  001  0021 2690   2692
@@E039  001  0022 2692   2694
@@E040  001  0023 2694   2696
@@E041  001  0024 2696   2698
@@E042  001  0025 2698   2700
@@E043  001  0026 2700   2702
@@E044  001  0027 2702   2704
@@E045  001  0028 2704   2706
@@E046  001  0029 2706   2708
@@E060  001  002A 2708   2710
@@E080  001  002B 2710
@@E100  001  0000 2096   2098
@@E101  001  0001 2098   2100
@@E102  001  0002 2100   2102
@@E103  001  0003 2102   2104
@@E110  001  0004 2104   2106
@@E112  001  0005 2106   2108
@@E113  001  0006 2108   2110
@@E114  001  0007 2110   2112
@@E115  001  0008 2112   2114
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 185
@@E116  001  0009 2114   2116
@@E117  001  000A 2116   2118
@@E120  001  000B 2118   2120
@@E122  001  000C 2120   2122
@@E123  001  000D 2122   2124
@@E124  001  000E 2124   2126
@@E129  001  000F 2126   2128
@@E130  001  0010 2128   2130
@@E131  001  0011 2130   2132
@@E133  001  0012 2132   2134
@@E134  001  0013 2134   2136
@@E135  001  0014 2136   2138
@@E136  001  0015 2138   2140
@@E137  001  0016 2140   2142
@@E138  001  0017 2142   2144
@@E139  001  0018 2144   2146
@@E142  001  0019 2146   2148
@@E143  001  001A 2148   2150
@@E150  001  001B 2150   2152
@@E151  001  001C 2152   2154
@@E160  001  001D 2154   2156
@@E162  001  001E 2156   2158
@@E163  001  001F 2158   2160
@@E164  001  0020 2160   2162
@@E200  001  0021 2162   2164
@@E205  001  0022 2164   2166
@@E210  001  0023 2166   2168
@@E211  001  0024 2168   2170
@@E212  001  0025 2170   2172
@@E213  001  0026 2172   2174
@@E215  001  0027 2174   2176
@@E216  001  0028 2176   2178
@@E217  001  0029 2178   2180
@@E220  001  002A 2180   2182
@@E221  001  002B 2182   2184
@@E222  001  002C 2184   2186
@@E223  001  002D 2186   2188
@@E225  001  002E 2188   2190
@@E226  001  002F 2190   2192
@@E227  001  0030 2192   2194
@@E228  001  0031 2194   2196
@@E229  001  0032 2196   2198
@@E230  001  0033 2198   2200
@@E232  001  0034 2200   2202
@@E234  001  0035 2202   2204
@@E237  001  0036 2204   2206
@@E240  001  0037 2206   2208
@@E241  001  0038 2208   2210  3099
@@E242  001  0039 2210   2212
@@E248  001  003A 2212   2214
@@E249  001  003B 2214   2216
@@E250  001  003C 2216   2218
@@E251  001  003D 2218   2220
@@E252  001  003E 2220   2222
@@E253  001  003F 2222   2224
@@E254  001  0040 2224   2226
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                          VER 15, MOD 00  20/07/20  PAGE 186
@@E255  001  0041 2226   2228
@@E256  001  0042 2228   2230
@@E300  001  0043 2230   2232
@@E301  001  0044 2232   2234
@@E302  001  0045 2234   2236
@@E303  001  0046 2236   2238
@@E304  001  0047 2238   2240
@@E305  001  0048 2240   2242
@@E308  001  0049 2242   2244
@@E310  001  004A 2244   2246
@@E315  001  004B 2246   2248
@@E316  001  004C 2248   2250
@@E320  001  004D 2250   2252
@@E325  001  004E 2252   2254
@@E330  001  004F 2254   2256
@@E335  001  0050 2256   2258
@@E338  001  0051 2258   2260
@@E340  001  0052 2260   2262
@@E350  001  0053 2262   2264
@@E351  001  0054 2264   2266
@@E352  001  0055 2266   2268
@@E360  001  0056 2268   2270
@@E361  001  0057 2270   2272
@@E362  001  0058 2272   2274
@@E371  001  0059 2274   2276
@@E380  001  005A 2276   2278
@@E390  001  005B 2278   2280
@@E400  001  005C 2280   2282
@@E410  001  005D 2282   2284
@@E415  001  005E 2284   2286
@@E417  001  005F 2286   2288
@@E420  001  0060 2288   2290
@@E430  001  0061 2290   2292
@@E432  001  0062 2292   2294
@@E433  001  0063 2294   2296
@@E450  001  0064 2296   2298
@@E451  001  0065 2298   2300
@@E460  001  0066 2300   2302
@@E461  001  0067 2302   2304
@@E464  001  0068 2304   2306
@@E465  001  0069 2306   2308
@@E466  001  006A 2308   2310
@@E467  001  006B 2310   2312
@@E469  001  006C 2312   2314
@@E470  001  006D 2314   2316
@@E471  001  006E 2316   2318
@@E473  001  006F 2318   2320
@@E474  001  0070 2320   2322
@@E475  001  0071 2322   2324
@@E476  001  0072 2324   2326
@@E477  001  0073 2326   2328
@@E478  001  0074 2328   2330
@@E479  001  0075 2330   2332
@@E480  001  0076 2332   2334
@@E481  001  0077 2334   2336
@@E482  001  0078 2336   2338
```

```
@@E483  001  0079 2338   2340
@@E484  001  007A 2340   2342
@@E485  001  007B 2342   2344
@@E486  001  007C 2344   2346
@@E487  001  007D 2346   2348
@@E488  001  007E 2348   2350
@@E489  001  007F 2350   2352
@@E490  001  0080 2352   2354
@@E491  001  0081 2354   2356
@@E492  001  0082 2356   2358
@@E493  001  0083 2358   2360
@@E494  001  0084 2360   2362
@@E495  001  0085 2362   2364
@@E496  001  0086 2364   2366
@@E497  001  0087 2366   2368
@@E498  001  0088 2368   2370
@@E500  001  0089 2370   2372
@@E501  001  008A 2372   2374
@@E530  001  008B 2374   2376
@@E531  001  008C 2376   2378
@@E535  001  008D 2378   2380
@@E540  001  008E 2380   2382
@@E541  001  008F 2382   2384
@@E542  001  0090 2384   2386
@@E543  001  0091 2386   2388
@@E544  001  0092 2388   2390
@@E545  001  0093 2390   2392
@@E546  001  0094 2392   2394
@@E547  001  0095 2394   2396
@@E548  001  FFFF 2600
@@E549  001  0096 2396   2398
@@E550  001  0097 2398   2400
@@E551  001  0098 2400   2402
@@E552  001  0099 2402   2404
@@E553  001  009A 2404   2406
@@E554  001  009B 2406   2408
@@E555  001  009C 2408   2410
@@E556  001  009D 2410   2412
@@E558  001  009E 2412   2414
@@E570  001  009F 2414   2416
@@E571  001  00A0 2416   2418
@@E572  001  00A1 2418   2420
@@E573  001  00A2 2420   2422
@@E574  001  00A3 2422   2424
@@E575  001  FFFF 2602
@@E578  001  00A4 2424   2426
@@E579  001  FFFF 2604
@@E580  001  FFFF 2606
@@E585  001  00A5 2426   2428
@@E595  001  FFFF 2608
@@E597  001  FFFF 2610
@@E598  001  FFFF 2612
@@E600  001  00A6 2428   2430  3683
@@E601  001  00A7 2430   2432
@@E602  001  00A8 2432   2434
@@E603  001  00A9 2434   2436
```

```
@@E604  001  00AA 2436   2438  6854
@@E606  001  00AB 2438   2440  8416
@@E607  001  00AC 2440   2442  8411
@@E608  001  00AD 2442   2444  5968
@@E609  001  00AE 2444   2446  0772
@@E610  001  00AF 2446   2448
@@E611  001  00B0 2448   2450
@@E612  001  00B1 2450   2452  0809
@@E613  001  00B2 2452   2454
@@E614  001  00B3 2454   2456
@@E700  001  00B4 2456   2458
@@E701  001  00B5 2458   2460
@@E710  001  00B6 2460   2462
@@E712  001  00B7 2462   2464
@@E713  001  00B8 2464   2466
@@E714  001  00B9 2466   2468
@@E715  001  00BA 2468   2470
@@E716  001  00BB 2470   2472
@@E717  001  00BC 2472   2474
@@E718  001  00BD 2474   2476
@@E720  001  00BE 2476   2478
@@E721  001  00BF 2478   2480
@@E723  001  00C0 2480   2482
@@E724  001  00C1 2482   2484
@@E725  001  00C2 2484   2486
@@E726  001  00C3 2486   2488
@@E727  001  00C4 2488   2490
@@E728  001  00C5 2490   2492
@@E729  001  00C6 2492   2494
@@E730  001  00C7 2494   2496
@@E732  001  00C8 2496   2498
@@E752  001  00C9 2498   2500
@@E753  001  00CA 2500   2502
@@E754  001  00CB 2502   2504
@@E755  001  00CC 2504   2506
@@E756  001  00CD 2506   2508
@@E757  001  00CE 2508   2510
@@E758  001  00CF 2510   2512
@@E759  001  00D0 2512   2514
@@E760  001  00D1 2514   2516
@@E761  001  00D2 2516   2518
@@E762  001  00D3 2518   2520
@@E763  001  00D4 2520   2522
@@E764  001  00D5 2522   2524
@@E765  001  00D6 2524   2526
@@E766  001  00D7 2526   2528
@@E767  001  00D8 2528   2530
@@E768  001  00D9 2530   2532
@@E769  001  00DA 2532   2534
@@E770  001  00DB 2534   2536
@@E771  001  00DC 2536   2538
@@E772  001  00DD 2538   2540
@@E773  001  00DE 2540   2542
@@E774  001  00DF 2542   2544
@@E775  001  00E0 2544   2546
@@E776  001  00E1 2546   2548
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 189
@@E777  001  00E2 2548   2550
@@E778  001  00E3 2550   2552
@@E779  001  00E4 2552   2554
@@E780  001  00E5 2554   2556
@@E781  001  00E6 2556   2558
@@E782  001  00E7 2558   2560
@@E783  001  00E8 2560   2562
@@E784  001  00E9 2562   2564
@@E785  001  00EA 2564   2566
@@E786  001  00EB 2566   2568
@@E790  001  00EC 2568   2570
@@E791  001  00ED 2570   2572
@@E792  001  00EE 2572   2574
@@E793  001  00EF 2574   2576
@@E794  001  00F0 2576   2578
@@E795  001  00F1 2578   2580
@@E796  001  00F2 2580   2582
@@E797  001  00F3 2582   2584
@@E798  001  00F4 2584   2586
@@E800  001  FFFF 2614
@@E801  001  FFFF 2616
@@E802  001  FFFF 2618
@@E803  001  FFFF 2620
@@E804  001  FFFF 2622
@@E900  001  00F5 2586   2588  3095
@@E901  001  00F6 2588   2590  3097
@@E902  001  00F7 2590   2592  3096
@@E903  001  00F8 2592   2594  3098
@@E905  001  00F9 2594   2596
@@E906  001  00FA 2596   2598
@@E910  001  00FB 2598   3094
@ARR    001  0008 0016   4565  4690  4706  4868  6282  6672
@ASIGN  001  007C 0071
@ASTER  001  005C 0069
@BCRDL  001  0050 0088
@BE     001  0081 0043
@BF     001  0090 0052
@BH     001  0084 0041
@BL     001  0082 0042
@BLANK  001  0040 0065
@BM     001  0082 0054
@BNE    001  0001 0046
@BNH    001  0004 0044
@BNL    001  0002 0045
@BNM    001  0002 0057
@BNOL   001  0020 0050
@BNOZ   001  0008 0049
@BNP    001  0004 0056
@BNZ    001  0001 0058
@BOL    001  00A0 0048
@BOZ    001  0088 0047
@BP     001  0084 0053
@BR     001  0001 0013   3251  3256  3256  3265  3273  3277  3293  3298  3302  3330  3330  3342
                         3342  3348  3349  3353  3362  3366  3377  3377  3383  3383  3384  3385
                         3385  3391  3391  3395  3401  3401* 3402  3402* 3403  3455  3465  3469
                         3473  3485  3489  3496  3496  3500  3500  3504  3508  3515  3672  3707
                         3713  3714  3720  3724  3739  3754  3875  3880  3880  3904  3905  3924
                         3925  3930  3931  3968  3969  3977  3977  3983  3983  3984  3986  3986
                         3992  3992  3996  4002  4002* 4003  4042  4070  4073  4099  4119  4123
                         4290  4299  4420  4426  4426  4429  4445  4445  4453  4454  4474  4479
                         4480  4480  4486  4488  4489* 4496  4497  4498  4504  4505  4507  4507
                         4513  4514  4515  4515  4522  4522  4523  4529  4529* 4530  4530* 4531
                         4539  4540  4541  4545  4550  4551  4551  4552  4552  4553  4554  4565
                         4569  4605  4618  4620  4625  4626  4627  4627  4628  4629  4632  4634
                         4635  4636  4637  4643  4644  4645  4646  4647  4648  4653  4655  4661
                         4667  4668* 4675  4675  4677  4690  4694  4706  4707  4708  4709  4710
                         4712  4713  4714  4715  4715  4716  4717  4718  4718  4719  4720  4721
                         4722  4723  4724  4766  4773  4774  4780  4788  4789  4790  4801  4806
                         4808  4809* 4819  4820  4823  4824  4825  4826  4827  4829  4830  4831
                         4832  4833  4842  4843  4844  4849  4868  4872  5010  5018  5019  5026
                         5027  5027  5047  5061  5062  5066  5068  5073  5082  5084  5086* 5093
                         5093  5094  5095  5102  5103  5111  5121  5121  5127  5127  5128  5129
                         5129  5135  5135  5139  5140  5140  5146  5146* 5147  5147* 5148  5179
                         5191  5197  5199  5204  5209  5218  5233* 5234  5239  5249  5256  5257
                         5266  5402  5406  5410  5420  5425  5426  5587  5599  5632  5648  5655
                         5656  5658  5659  5662  5669  5672  5676  5686  5692  5696  5703  5708
                         5897  5911  5933  5936  5937  5946  5947  5951  5952  5966  5974  5977
                         6127  6140  6144  6145  6166  6171  6172  6176  6178  6182  6183  6184
                         6188  6192  6193  6197  6202  6206  6216  6218  6222  6223  6224  6226
                         6227  6231  6235  6237  6241  6242  6246  6250  6250  6251  6262  6277
                         6282  6494  6507  6509  6514  6518  6520  6535  6557  6561  6565  6573
                         6592  6603  6604  6618  6619  6623  6627  6631  6633  6637  6638  6642
                         6647  6651  6651  6655  6667  6672  6817  6830  6861  6862  6866  6868
                         6869  6875  6908  6909  6924  6932  7086  7105  7114  7118  7124  7129
                         7133  7140  7147  7153  7161  7162  7170  7170  7171  7180  7189  7198
                         7338  7347  7464  7476  7477  7478  7479  7491  7499  7503  7519  7523
                         7530  7546  7551  7555  7562  7571  7575  7575  7579  7580  7584  7721
                         7730  7841  7857  7878  7879  7896  7900  7909  8054  8059  8059  8067
                         8073  8199  8219  8228  8379  8423  8424  8431  8436  8437  8572  8585
                         8589  8598  8734  8743  8876  8894  8913  8918  8921  8925  8945  8952
                         8961  9129  9137  9146  9155  9281  9299  9318  9324  9327  9331  9351
                         9358  9367  9528  9542  9546  9555  9682  9695  9702  9707  9712  9720
                         9729  9729  9737  9750  9757  9765  9766  9901  9908  9918  0041  0153
                         0166  0174  0178  0193  0211  0220  0224  0359  0366  0376  0497  0737
                         0742  0742  0754  0766  0778  0818  0844  0863  0863  0869  0870  0870
                         0876  0876  0880  0886  0886* 0887  0942  0969  0970  0976  0983  0988
                         0989  0990  0994  0994  0995  0995  0996  1004  1005  1011  1018  1023
                         1024  1025  1029  1029  1030  1030  1031  1039  1040  1046  1053  1058
                         1059  1060  1064  1064  1065  1065  1066  1074  1075  1245  1358
@BT     001  0010 0051
@BZ     001  0081 0055
@B1     001  0001 0063   3331  3343  4461  4497  4497  4497* 4513* 4514  4644  4644  4644* 4789
                         4789  4789* 4797  5018  5043  5093  5128  5140  5191  5209  5222  5227
                         5256  5918
@CADDR  001  0002 0142   1945  1946  1947  3048  3075  3377  3383  3385  3391  3423  3425  3426
                         3432  3433  3977  3983  3986  3992  4009  4012  4013  4014  4020  4023
                         4024  4429  4480  4504  4507  4515  4522  4551  4552  4553  4589  4590
                         4594  4597  4598  4661  4667  4675  4758  4760  5094  5121  5127  5129
                         5135  5156  5158  5159  5160  5161  5283  5749  5754  5759  5764  5769
                         5774  5961  5966  6011  7932  8251  8431  8455  8456  8621  9577  9941
                         0251  0399  0766  0824  0863  0870  0876  0893  0895  0897  0912  0928
                         0931  0932  0933  0994  1029  1064  1077  1083  1085  1086  1087  1110
                         1111  1112  1114  1115  1116  1118  1119  1120
@CARDL  001  0060 0087   0644
@CHARA  001  00C1 0072
@CHARF  001  00C6 0073
@CHARR  001  00D9 0074
@CHARZ  001  00E9 0075
@CLOFF  001  0010 0094
@CLON   001  0011 0093
@COMMA  001  006B 0066
@CPLUS  001  004E 0079
@DADDR  001  0002 0140
@DBFR1  001  0004 0129
@DBFR2  001  0005 0130
@DCALK  001  0001 0081
@DCBCY  001  0009 0115   1774
@DCBT1  001  0050 0117   1777
@DCNT   001  0003 0128
@DCST1  001  0040 0116   1775
@DCTRL  001  0000 0125
@DCYL   001  0001 0126
@DD2    001  0003 0030
@DGET   001  0001 0134   0908  1097
@DOLAR  001  005B 0068
@DOP2   001  0004 0028
@DPLNG  001  0006 0132
@DPOS   001  0000 0133
@DPUT   001  0002 0135
@DSAD   001  0002 0127
@DSBCY  001  0004 0106   1712
@DSCS1  001  0000 0107   1713
@DSIVF  001  0003 0138
@DSPIN  001  0002 0131
@DTRSZ  001  0018 0085
@DVBCY  001  0007 0108   1771
@DVRFY  001  0031 0136
@DWAIT  001  00FF 0137
@DWBCY  001  0005 0103   1768
@DWSIZ  001  00C0 0105
@DWTB1  001  0003 0104   1769
@DZERO  001  00F0 0064
@D1     001  0002 0026   3298* 3302* 3330* 3342* 3349* 3473* 3500* 3905* 3925* 3931* 3969* 5103*
                         5111* 5140  6144* 6166* 6183* 6216* 6222* 0977  1012  1047  1142  1143
                         1144
@EOF    001  001C 0077
@EOFTC  001  0075 0162
@EOS    001  001E 0076   1784  5921
@FDDBC  001  0000 0195
@FDE1   001  000C 0200
@FDFNA  001  000B 0198
@FDHLN  001  0002 0208
@FDLNC  001  0002 0193
@FDNSC  001  0003 0210
@FDSD   001  0000 0206
@FLACE  001  0009 0197
@FLDBC  001  0001 0196
@FLENT  001  0004 0201
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 192
@FLFNA  001  0002 0199
@FLHLN  001  0002 0209
@FLLNC  001  0002 0194
@FLNSC  001  0001 0211
@FLSD   001  0001 0207
@HDRLN  001  0007 0092   0672
@IAR    001  0010 0017
@INDEX  001  0001 0156   0157
@INST3  001  0003 0032   3298  3349  6258
@INST4  001  0004 0033   0979  1014  1049
@INST5  001  0005 0034
@INST6  001  0006 0035
@I1IAR  001  00C0 0020
@LINSZ  001  00F4 0084   0646
@MAPEN  001  0005 0089
@MINCR  001  2000 0083
@MINUS  001  0060 0080
@NOP    001  0080 0040   6257  6391
@NUMBR  001  007B 0070
@OPD2   001  0004 0029
@OP1    001  0003 0027   3672* 3714  3724  3739  4429* 4565* 4667* 4690* 4706* 4868* 6282* 6672*
                         8585* 9542*
@OP2    001  0005 0031
@PCTRL  001  0000 0149
@PDATA  001  0003 0151
@PGCSZ  001  0020 0082   0083
@PPLNG  001  0004 0148
@PRCNT  001  0001 0150
@PRETR  001  00C0 0154
@PRINT  001  0040 0152   0154
@PSR    001  0004 0015
@PWAIT  001  00FF 0158
@P1IAR  001  0020 0018
@P2IAR  001  0040 0019
@Q      001  0001 0024   4540* 4569* 4636* 4694* 4712* 4825* 4832* 4872* 5191* 5209* 6171* 6256
                         6390  7124* 8894* 8913* 9299* 9318*
@REGL   001  0002 0012
@RETRN  001  0080 0153   0154
@RLDWN  001  004F 0159
@RTRNC  001  0080 0161
@SBLN   001  0005 0170
@SBLNL  001  0002 0184
@SCTSZ  001  0100 0100
@SDFLN  001  0007 0090
@SDF0   001  0000 0166
@SDF1   001  0001 0167
@SDF2   001  0002 0168
@SDF3   001  0003 0169
@SECCY  001  0030 0086
@SIST   001  0001 0181
@SLASH  001  0061 0067
@SLAST  001  0002 0183
@SMIDL  001  0003 0182
@SNULL  001  0080 0173
@SONLY  001  0000 0180
@STEXT  001  0007 0172
```

```
@STYPE  001  0006 0171
@SYLVL  001  0005 3130
@TBCNT  001  0000 0160
@TBLEF  001  0010 0155   0157
@TBLIX  001  0011 0157
@UCB    001  0087 0039   6391
@UPARW  001  005A 0078   3113
@VADDR  001  0002 0141   1505  1941  1953  1954  1955  1955  1969  1972  1974  1998  1999  2000
                         2038  2041  2044  2047  2050  2053  2056  2065  2068  2071  2074  2077
                         3049  3075  3272  3273  3284  3292  3293  3434  3459  3464  3465  3544
                         3761  4455  4578  4618  4627  4707  4708  4715  4718  4779  4780  4781
                         4842  4848  4849  4879  4881  4893  5162  5265  5266  5279  5406  5418
                         5419  5420  5425  5426  5451  5457  5610  5620  5626  5662  5686  5702
                         5703  5707  5708  5781  5911  5936  5974  5975  5976  5977  5991  6012
                         6014  6231  6250  6374  6384  6513  6514  6525  6534  6535  6543  6623
                         6651  6698  6700  6837  6868  6869  6875  6908  6909  6932  6970  7114
                         7118  7161  7162  7163  7170  7197  7198  7210  7498  7499  7546  7575
                         7600  7607  7610  8397  8423  8436  8437  8463  8741  8742  8743  8764
                         8960  8961  8982  9366  9367  9389  9702  9719  9720  9724  9729  9765
                         9766  9793  0173  0174  0187  0192  0193  0201  0832  0836  0840  0844
                         0846  0850  0854  0855  0894  0900  0970  0976  1005  1011  1040  1046
                         1060  1089  1134  1135  1137  1139
@VENTA  001  0056 0113   1772  2027
@VMDDV  001  00FE 0114
@VMFD1  001  0000 0109
@VMFD2  001  0001 0110
@VMRS3  001  0002 0112
@VMTRL  001  0001 0111
@VOLID  001  0006 0091
@VQ     001  0001 0025
@WSFIT  001  0500 0101
@WSTBL  001  0503 0102
@XR     001  0002 0014   3265* 3266  3277* 3278  3314* 3315  3315* 3324  3331  3331* 3336  3343
                         3343* 3344  3348  3353* 3354  3360* 3361  3366* 3367  3395* 3477* 3478
                         3478* 3480  3485  3489* 3490  3508* 3509  3515* 3516  3671* 3672  3676
                         3698* 3702  3706  3711* 3712  3714* 3719  3724* 3729  3739* 3744  3748
                         3753  3895  3900  3902  3907  3909  3914* 3915  3917  3919  3935* 3936
                         3996* 4052* 4059* 4070* 4071  4071* 4072  4077  4077* 4078  4099* 4100
                         4113* 4119* 4120  4290* 4291  4297* 4298  4453* 4456* 4486* 4487  4490*
                         4496* 4505* 4506  4523* 4539* 4542  4566  4570* 4619  4625* 4628* 4631
                         4635* 4643* 4646* 4677* 4691  4709* 4713* 4716* 4719* 4721* 4723* 4773*
                         4788* 4792  4806* 4807  4810* 4819* 4822  4824* 4829* 4831* 4843* 4869
                         5035  5060  5062* 5063  5068* 5069  5082* 5083  5087* 5139* 5187* 5191
                         5196  5198  5209  5215* 5216  5216* 5217  5222  5239* 5249* 5250  5257*
                         5258  5402* 5403  5410* 5411  5599* 5600  5632* 5633  5641  5649  5655
                         5656* 5657  5657* 5658  5660  5661  5661* 5662  5669  5671  5676* 5691
                         5696* 5697  5921  5933* 5937* 5938  5952* 5953  5958* 5959  5959* 5960
                         5974  5975  5976  5977  6171  6172* 6174  6174* 6175  6177  6182  6183
                         6211* 6225  6235* 6277* 6283  6285* 6507* 6518* 6552  6601* 6602  6631*
                         6667* 6673  6675* 6830* 6831  6842* 6847* 6848  6874* 6875  6884  6889
                         6894  6924* 6925  7105* 7106  7122* 7124  7129* 7130  7133* 7134  7153*
                         7154  7171* 7172  7180* 7181  7189* 7190  7338* 7339  7345* 7346  7479*
                         7480  7491* 7492  7503* 7523* 7524  7530* 7531  7555* 7556  7562* 7563
                         7721* 7722  7728* 7729  7857* 7858  7900* 7901  7907* 7908  8064  8073*
                         8074  8199* 8200  8219* 8220  8226* 8227  8396* 8397  8406  8423  8424*
                         8425  8572* 8573  8585  8589* 8590  8596* 8597  8734* 8735  8894  8899
                         8901  8913  8918* 8919  8919* 8920  8925  8930* 8945* 8946  8952* 8953
                         9124  9129* 9130  9137* 9138  9144* 9145  9154  9299  9304  9306  9318
                         9324* 9325  9325* 9326  9331  9336* 9351* 9352  9358* 9359  9528* 9529
                         9542  9546* 9547  9553* 9554  9695* 9696  9706* 9712* 9713  9733* 9734
                         9750* 9751  9757* 9758  9901* 9902  9908* 9909  9917  0041* 0042  0166*
                         0167  0178* 0179  0182* 0211* 0212  0218* 0219  0224* 0225  0359* 0360
                         0366* 0367  0375  0497* 0498  0754* 0778* 0779  0802* 0820* 0880* 0969*
                         0970  0974* 0976  0988* 0989* 0990  1004* 1005  1009* 1011  1023* 1024*
                         1025  1039* 1040  1044* 1046  1058* 1059* 1060  1074* 1075  1245* 1246
                         1358* 1359
@ZERO   001  0000 0062   3302  3469  3473  3480  3888  3957  4427  4463  4473  4578  4598  4799
                         4800  4879  5026  5045  5046  5058  5060  5103  5234  5920  6217  6613
                         7513  8054  9707  0759
B$ADMK  001  0001 1409
B$ADSW  001  159D 1408
B$ARMK  001  0001 1394   6161  6596  7873
B$ARSW  001  0A45 1393   6161  6596  7873
B$BABF  001  1D00 1199
B$BCKT  001  1590 1321   4618  4707  5686  5911  5936  6231  6623  7546  8397
B$BDPL  001  19E8 1273   0802
B$BDSA  001  19EA 1274   0803
B$BINO  001  1A6A 1337   3702  3729  3748  5267  5418  6543  8741  8962  9368  9724  0201
B$BRLN  001  19F1 1272   3284* 3459* 4781* 4848  5267* 5418* 5610* 5707  6525* 6543* 7163* 7197
                         8741* 8962* 9368* 9724* 0187* 0201*
B$BROP  001  1AF7 1378   4455* 4779  4842
B$BRVA  001  19EF 1271   3272* 3273* 3292* 3293* 3464* 3465* 4779* 4780* 4848* 4849* 5265* 5266*
                         5419* 5420* 5425* 5426* 5626* 5707* 5708* 6513* 6514* 6534* 6535* 6837*
                         6932* 7161* 7162* 7197* 7198* 7498* 7499* 8436* 8437* 8742* 8743* 8960*
                         8961* 9366* 9367* 9719* 9720* 9765* 9766* 0173* 0174* 0192* 0193*
B$BRVP  001  19EE 1270
B$BTAB  001  1996 1269   3288  3460  4782  5268  5421  5628  6530  6548  7165  8747  8963  9369
                         9725  0188  0202
B$CADR  001  1AF9 1379   4504* 4661  4668  5102* 5227  5233
B$CASA  001  0000 1214
B$CASC  001  0671 1218
B$CASM  001  0608 1216
B$CBAS  001  14BB 1344   4809  5086
B$CBFA  001  0CBC 1299
B$CCGT  001  0600 1224
B$CCLS  001  0695 1230
B$CCON  001  001F 1297   5643
B$CDAT  001  0600 1210
B$CDEF  001  0600 1211
B$CDIM  001  0673 1212
B$CDUM  001  0000 1248
B$CEND  001  0600 1246   1247
B$CEOF  001  0600 1247
B$CFOR  001  0600 1219
B$CGET  001  06A3 1227
B$CGSB  001  0690 1225
B$CGTO  001  06B3 1223
B$CIFA  001  0600 1221
B$CIFC  001  0600 1222
B$CIMG  001  0600 1236
B$CINP  001  0600 1231
B$CLTA  001  0000 1213
```

```
B$CLTC  001  0669 1217
B$CLTM  001  0600 1215
B$CMAT  001  0600 1237
B$CMGT  001  0665 1238
B$CMIN  001  06D3 1239
B$CMPR  001  069B 1242
B$CMPT  001  069B 1241
B$CMPU  001  0600 1243
B$CMRD  001  06D0 1240
B$CNXT  001  0600 1220
B$CPCT  001  0CA8 1302   6217  6241* 6613  6637* 7513  7579*
B$CPRT  001  0600 1234
B$CPRU  001  0600 1235
B$CPSE  001  06E7 1244
B$CPUT  001  0600 1228
B$CPWA  001  0CA6 1373
B$CRAD  001  150D 1343   4807* 5083*
B$CRBS  001  1509 1345   4808* 5084*
B$CREA  001  06CF 1232
B$CREM  001  0000 1209
B$CRMK  001  0001 1421   3319  3696
B$CRSR  001  06E3 1233
B$CRST  001  06A6 1229
B$CRSW  001  0E42 1420   3696  4473  4800  5046
B$CRTN  001  06CF 1226
B$CSBF  001  0600 1196   1210  1211  1212  1215  1216  1217  1218  1219  1220  1221  1222  1223
                         1224  1225  1226  1227  1228  1229  1230  1231  1232  1233  1234  1235
                         1236  1237  1238  1239  1240  1241  1242  1243  1244  1245  1246  1249
                         1250  1251  1252  1253  3433  4014  4590  4760  5161  5283  0895
B$CSCN  001  14B0 1318   4795  5041  5057  6157  6586  7851  7892  8072  8193  8567  9295  9337
                         9523  9897  0355
B$CSMK  001  0007 1424   6153  6581  7883
B$CSSW  001  14BC 1423   6153  6581  7883
B$CSTP  001  06D6 1245
B$CSTR  001  14CC 1342   4811  5088
B$CSXA  001  2000 1202   3426  4024  5159  0933
B$CTYP  001  0A5F 1296   5643* 6210* 6608* 7507*
B$CVPD  001  0C5D 1301   0791
B$CVPG  001  0CA5 1300
B$CWRK  001  F500 1370   4583  4741  4884
B$DIST  001  0700 1262   3526  3755  4106  4303  4856  5272  5435  5713  5981  6266  6569  6940
                         7204  7351  7541  7734  7913  8077  8232  8441  8602  8751  8967  9159
                         9373  9559  9774  9922  0231  0380
B$DLNK  001  1B37 1368   5626  5702* 5703*
B$DL4T  001  1A6B 1339   0755  0814  0821
B$DPWA  001  0E46 1374
B$DST2  001  073A 1263   3396  3997  4524  4678  5141  5240  0881
B$ERMK  001  0007 1397   7474  0746
B$ERSW  001  0993 1396   7474  0746
B$FACA  001  0E53 1305   3671  6847  6874
B$FAIS  001  15AC 1322   0854
B$FAIW  001  15A0 1323   0855
B$FCON  001  0A46 1295   5681  5935  6212  6609  7508
B$FORT  001  1B0E 1364   0897
B$FPWA  001  15AC 1375
B$FRMK  001  0007 1415
```

```
B$FRSW  001  16CC 1414
B$FSC1  001  0E4C 1306   6884*
B$FSC2  001  0E4D 1307   6894* 6902*
B$FSMK  001  0007 1406   6914  6919
B$FSSW  001  0E5C 1405   6914* 6919*
B$FSVA  001  0E4F 1308   6908* 6909*
B$FTND  001  1B0B 1366   5961
B$FTPT  001  1B0D 1365   5958  5960* 5961  5966* 8396  8431* 0766
B$FVME  001  15A2 1327   5764
B$FVMP  001  15A4 1328   5769
B$FVMS  001  15A6 1329   5774
B$FVPE  001  15A8 1324   5749
B$FVPP  001  15AA 1325   5754
B$FVPS  001  15AC 1326   5759
B$GBSW  001  08AF 1399
B$GBWK  001  0001 1400
B$GETC  001  0867 1276   3261  3306  3662  3666  3688  3733  3752  3886  3890  3901  3908  3910
                         3952  4094  4277  4281  4437  4544  4612  4654  4676  4791  4817  4855
                         5028  5056  5072  5110  5195  5393  5637  5654  5664  5670  5906  5917
                         5927  6136  6503  6826  6879  6885  6896  6903  7095  7146  7329  7487
                         7712  7850  7865  8045  8066  8192  8206  8211  8388  8566  8580  8726
                         8886  8898  8937  9115  9150  9291  9303  9343  9522  9536  9691  9736
                         9742  9896  9916  0162  0354  0374
B$GPTR  001  0878 1278   3360  3711  3935  4297  4456  4490  4570  4810  5087  5187  6211  6285
                         6601  6675  6842  7122  7345  7728  7907  8226  8930  9144  9336  9706
                         9733  0182  0218
B$GTBF  001  1E00 1200
B$IFMK  001  0007 1418
B$IFSW  001  16E5 1417
B$INVT  001  1B38 1358   3314  3445  3477
B$KWMK  001  0001 1412
B$KWSW  001  159E 1411   4463* 4799* 5045* 5058*
B$LBAS  001  185E 1349   4489
B$LBSV  001  18E7 1347   4488*
B$LDRP  001  1A00 1197   0832  0832* 0836* 0840* 0846* 0850* 0854* 0855* 0946* 0950* 0951* 0952*
                         0956* 0960  0960* 0961  0961* 0974  1009  1044  1083
B$LINE  001  07D0 1264   7478
B$LIST  001  1853 1331   3310  4285  7123  8058  8215
B$LRTN  001  18EB 1348   4487*
B$LSTR  001  1862 1346   4491
B$LTYP  001  18F2 1332   3319
B$MATR  001  18F3 1334   3889  3951  3961  4054  4060  4095  4114  7334  7716  8584  9120  9541
                         0207
B$MBMK  001  0007 1433   3960  3962
B$MBSW  001  1903 1432   3960* 3962*
B$MFBK  001  1B8F 1360   3900* 3907* 3909* 3914  3943  3945  4052  4059  4065* 4066* 4072  4113
B$MGMK  001  0007 1430   3959  3963  4053  4061  4112  4115
B$MGSW  001  18FF 1429   3959* 3963* 4053* 4061* 4112* 4115*
B$MPMK  001  0007 1436   3887  3891
B$MPSW  001  1981 1435   3887* 3891*
B$MRMK  001  0007 1427
B$MRSW  001  0DDE 1426
B$NUMC  001  0873 1277   3260* 3661* 3885* 4051* 4058* 4111* 4276* 4436* 4611* 4794* 4816* 5025*
                         5040* 5055* 5109* 5203* 5392* 5595* 5660* 5677* 5905* 5916* 5926* 5934*
                         6135* 6201* 6502* 6825* 6895* 7094* 7328* 7486* 7509* 7711* 7849* 7864*
                         7888* 8044* 8191* 8210* 8387* 8565* 8579* 8725* 8885* 8907* 8936* 9114*
                         9119* 9290* 9312* 9342* 9521* 9535* 9690* 9741* 9895* 9915* 0161* 0353*
                         0373*
B$NXMK  001  0007 1403   3522  4850  5431  5709  6936  7199  7537  8435  9770
B$NXSW  001  071D 1402   3522* 4850* 5431* 5709* 6936* 7199* 7537* 8435* 9770*
B$PARP  001  0A41 1285
B$PBNL  001  0A01 1291   7476*
B$PCAD  001  0A40 1286   3266* 3278* 3354* 3367* 3490* 3509* 3516* 4078* 4100* 4120* 4291* 4566*
                         4691* 4869* 5063* 5069* 5250* 5258* 5403* 5411* 5600* 5633* 5697* 5938*
                         5953* 6283* 6673* 6831* 6925* 7106* 7130* 7134* 7154* 7172* 7181* 7190*
                         7339* 7480* 7492* 7524* 7531* 7556* 7563* 7722* 7858* 7901* 8074* 8200*
                         8220* 8425* 8573* 8590* 8735* 8946* 8953* 9130* 9138* 9352* 9359* 9529*
                         9547* 9696* 9713* 9751* 9758* 9902* 9909* 0042* 0167* 0179* 0212* 0225*
                         0360* 0367* 0498* 0779* 1246* 1359*
B$PCDL  001  09D3 1290   5976  6869
B$PCPG  001  0A35 1289   0840
B$PECT  001  0A44 1293   0761
B$PERC  001  0A39 1292   3683* 3888* 3957  5968* 6854* 8411* 8416*
B$PFAE  001  0033 1283   3682  5967  6853  8402
B$PFCL  001  009D 1284   0751  0786
B$PFNC  001  094E 1281   3682* 5967* 6853* 8402* 0751* 0786* 0797*
B$PFWP  001  0015 1282   0797
B$PNBY  001  0A41 1287   3267* 3279* 3355* 3368* 3491* 3510* 3517* 4079* 4101* 4121* 4292* 4567*
                         4692* 4870* 5064* 5070* 5251* 5259* 5404* 5412* 5605* 5939* 5951* 6236*
                         6278* 6508* 6519* 6632* 6668* 6866* 6926* 7107* 7131* 7135* 7155* 7173*
                         7182* 7191* 7340* 7481* 7493* 7525* 7532* 7557* 7564* 7723* 7859* 7902*
                         8075* 8201* 8221* 8426* 8574* 8591* 8736* 8947* 8954* 9131* 9139* 9353*
                         9360* 9530* 9548* 9697* 9714* 9752* 9759* 9903* 9910* 0043* 0168* 0180*
                         0213* 0226* 0361* 0368* 0499* 0780* 1247* 1360*
B$PPWA  001  0A35 1372
B$PRM1  001  1AF3 1376   5222* 5256
B$PTBF  001  1F00 1201
B$PUTC  001  093A 1280   3268  3280  3356  3369  3492  3511  3518  3684  3938  3950  4047  4080
                         4089  4102  4110  4122  4293  4568  4693  4871  5065  5071  5252  5260
                         5405  5413  5606  5663  5687  5698  5940  5954  5969  6284  6674  6832
                         6855  6867  6927  7108  7132  7136  7156  7174  7183  7192  7341  7482
                         7494  7526  7533  7558  7565  7724  7860  7903  8076  8202  8222  8417
                         8427  8575  8592  8737  8948  8955  9132  9140  9354  9361  9531  9549
                         9698  9715  9753  9760  9904  9911  0044  0169  0181  0214  0227  0362
                         0369  0500  0752  0781  0787  0798  1248  1361
B$PVAD  001  0A43 1288   3272  3284  3292  3459  3464  4455  4781  5265  5406  5419  5610  5620
                         5702  5975  6513  6525  6534  6837  6868  7114  7163  7477* 7498  8742
                         8960  9366  9702  9719  0173  0187  0192  0836
B$RMRK  001  1AE6 1341   3958  0048  0504  1252  1365
B$RTRN  001  1AF5 1377   4427  4429  4506* 4553* 4667
B$SABF  001  1C00 1198   0899
B$SCAN  001  1514 1320   4093  4630  4633  4711  4821  4828  5059  5067  5915  5919  5928  6149
                         6577  6918  7179  7869  8890  8931  9746
B$SCAT  001  13C8 1315   1116
B$SCON  001  001B 1298   6210  6608  7507
B$SCVT  001  12E0 1313   0956
B$SDPL  001  07DA 1266   0820
B$SFAB  001  0E48 1310   0850
B$SFNT  001  143C 1316   1120
B$SLDT  001  109C 1312   0950  0951  0952
B$SLVT  001  1062 1311   0946
B$SNAT  001  131A 1314   1112
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 198
B$SPAT  001  07E0 1267
B$SSTA  001  1BAC 1362   4461* 4797* 5043* 5918* 5920*
B$STAS  001  061B 1251
B$STIF  001  0606 1253
B$STMA  001  061B 1252
B$STML  001  0600 1250
B$STRL  001  0600 1249
B$SVRB  001  0E46 1309   0844* 0846
B$SYMB  001  0DBC 1304   3670  4462  4613  4798  4818  5044  5910  6843  8392
B$TCD2  001  0001 1382   5222
B$TLTH  001  0002 1383   1384  5216
B$TOD1  001  0000 1381   5217
B$TOTB  001  1AF8 1384   5215
B$TTAB  001  1AFA 1380   1384
B$TYPE  001  0739 1265
B$WORK  001  15A0 1369   4497  4644  4708  4789  7118
B$ZDBN  001  19F2 1336   3692  3737  5115  5397  6539  8730  8941  9347  9708  0197
B@ABAS  001  0007 1969
B@ACD1  001  0001 1966   1967  3729*
B@ACD2  001  0003 1967   1968  3748* 0970  0990* 1135
B@AFLG  001  0000 1961   3676  3706* 3719* 3744*
B@ALLA  001  005C 1786
B@AMAX  001  0005 1968   1969
B@BLNK  001  0040 1795   4065  4066  6902
B@BLSZ  001  0100 1920   2059  2062  2065  2080  2083  3426  3432  3454  4009  4024  4041  4589
                         4597  4604  4765  5159  5160  5178  0893  0894  0899  0933  0941  1128
B@BREQ  001  0084 1575   8995  9402
B@BRHI  001  0088 1576   8998  9405
B@BRLO  001  0082 1574   9001  9408
B@BRNE  001  0094 1578   9004  9013  9411  9420
B@BRNH  001  0098 1579   9007  9414
B@BRNL  001  0092 1577   9010  9417
B@CADD  001  0006 1444
B@CADF  001  0058 1485   7919  8238  8608  9565  9928  0386
B@CBAS  001  0003 1972
B@CBNX  001  004A 1478   6685  0240
B@CBRA  001  0046 1476   3412  4577  4878  4889  5444  5719  6946  7216  7219  7593  8462  8757
B@CBRC  001  0044 1475   5276  8974  9381
B@CBRD  001  0048 1477   6960
B@CBRS  001  004C 1479   3538  7602  9783  1258
B@CCLS  001  005E 1488   0389
B@CCMC  001  0042 1474   5280  9379
B@CCMF  001  0040 1473   8973
B@CCNT  001  001F 1898
B@CCSA  001  003E 1472   9785
B@CDCA  001  006A 1494   5722
B@CDDL  001  006C 1495   5725
B@CDIV  001  000C 1447
B@CDMN  001  0001 1971   1972  3702* 1005  1025* 1137
B@CDWA  001  006E 1496   5993  6949
B@CEOF  001  0070 1497   0921
B@CEOP  001  0068 1493
B@CFCI  001  0016 1452
B@CFN0  001  0012 1450   4743  4892  5172
B@CFN1  001  0014 1451
B@CFOR  001  004E 1480   5987
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 199
B@CGET  001  0052 1482   3415  4309  8241
B@CHAR  001  0000 1911   3361  3712  3753  3895  3900  3902  3907  3909  3915  3917  3919  3936
                         4298  4542  4619  4631  4792  4822  5035  5191  5196  5198  5209  5641
                         5649  5655  5669  5671  5691  5921  6171  6225  6552  6602  6884  6889
                         6894  7124  7346  7729  7908  8064  8227  8597  8894  8899  8901  8913
                         9124  9145  9154  9299  9304  9306  9318  9554  9734  9917  0219  0375
B@CHLT  001  0004 1443   0054
B@CIEX  001  00C5 1871   5748  5763
B@CIMH  001  0066 1492   7590
B@CINI  001  0056 1484   3532
B@CIPI  001  00D7 1874   5753  5768
B@CIS2  001  00E2 1877   5758  5773
B@CMF1  001  0018 1453   4166  4170  4174  7357  7739  8611  9165  9168  9568  0243
B@CMF2  001  001A 1454   4132  4158  4162
B@CMF3  001  001C 1455   4146  4150  4154
B@CMMA  001  006B 1806   6225  6302  6323  6344  9734
B@CMPY  001  000A 1446
B@CMSM  001  001E 1456   4129
B@CNEG  001  0010 1449
B@CNXT  001  0050 1481   5990
B@COLN  001  007A 1808
B@CPMK  001  00FF 1716   1720  1724  1725  1759
B@CPRS  001  0060 1489   6380
B@CPRU  001  0062 1490   6688  7596  0246
B@CPUT  001  0054 1483   7922
B@CPWR  001  000E 1448
B@CRSR  001  005A 1486   1371
B@CRST  001  005C 1487   9931
B@CSA1  001  0036 1468
B@CSA2  001  0038 1469
B@CSB1  001  003A 1470   4749
B@CSC1  001  002A 1462   4746
B@CSD0  001  002E 1464
B@CSD1  001  0030 1465
B@CSD2  001  0032 1466
B@CSF1  001  0022 1458
B@CSF2  001  0024 1459
B@CSTA  001  0034 1467   3409  4731  4883  5441  6682  7222  9780  0237
B@CSTC  001  0028 1461   4582  4734  4740  6383  6691  7599
B@CSTF  001  0020 1457   4752  6002  7225
B@CSTH  001  0064 1491
B@CSTX  001  003C 1471   3535  4737  4886  5169
B@CSUB  001  0008 1445
B@CSVC  001  0002 1442   0510  0920
B@CTYP  001  0020 1896
B@CUSC  001  002C 1463   4580  4895  8089
B@CUSF  001  0026 1460   4755  7228
B@CVAR  001  005B 1785
B@DAMK  001  0080 1964   3676  3706
B@DASA  001  00FF 1725
B@DASC  001  0040 1729
B@DASM  001  0038 1727
B@DCGT  001  0050 1735
B@DCLS  001  0054 1741
B@DDAT  001  0024 1721
B@DDEF  001  0034 1722
```

```
B@DDIM  001  0004 1723
B@DDUM  001  00FF 1759
B@DEC0  001  00F0 1854
B@DEC1  001  00F1 1855
B@DEC2  001  00F2 1856
B@DEC3  001  00F3 1857
B@DEC4  001  00F4 1858
B@DEC5  001  00F5 1859
B@DEC6  001  00F6 1860
B@DEC7  001  00F7 1861
B@DEC8  001  00F8 1862
B@DEC9  001  00F9 1863
B@DEND  001  0058 1757   1758  0929
B@DEOF  001  0058 1758
B@DFOR  001  0028 1730
B@DGET  001  0040 1738
B@DGSB  001  0020 1736
B@DGTO  001  0044 1734
B@DIFA  001  0048 1732
B@DIFC  001  004C 1733
B@DIGS  001  007B 1788
B@DIMG  001  003C 1747
B@DINP  001  0000 1742   3424
B@DIVD  001  0061 1805
B@DLTA  001  00FF 1724
B@DLTC  001  0040 1728
B@DLTM  001  0038 1726
B@DL01  001  0001 2039   2042  0836*
B@DL02  001  0003 2042   2045  0840*
B@DL03  001  0005 2045   2048  0846*
B@DL04  001  0007 2048   2051  0832  0832* 0850*
B@DL05  001  0009 2051   2054  0854*
B@DL06  001  000B 2054   2057  0855*
B@DL07  001  0045 2057   2060  0946*
B@DL08  001  0145 2060   2063  0950*
B@DL09  001  0245 2063   2066  0951*
B@DL10  001  0289 2066   2069  0952*
B@DL11  001  02C3 2069   2072  0956* 0974
B@DL12  001  02FD 2072   2075  1009
B@DL13  001  0337 2075   2078  1044
B@DL14  001  0371 2078   2081
B@DL15  001  0471 2081   2084  0960  0960*
B@DL16  001  0507 2084   0961  0961* 1083
B@DMAT  001  0008 1748   4021
B@DMGT  001  0044 1749
B@DMIN  001  0038 1750
B@DMPR  001  0048 1753
B@DMPT  001  004C 1752
B@DMPU  001  0054 1754
B@DMRD  001  003C 1751
B@DNXT  001  0044 1731
B@DPNT  001  004B 1796
B@DPRT  001  002C 1745
B@DPRU  001  0030 1746
B@DPSE  001  0050 1755
B@DPUT  001  0040 1739
```

```
B@DREA  001  000C 1743
B@DREM  001  00FF 1720
B@DRSR  001  005C 1744
B@DRST  001  0050 1740
B@DRTN  001  005C 1737
B@DSCY  001  0004 1712
B@DSIF  001  001C 1761   5157  5284
B@DSLT  001  0010 1760
B@DSML  001  0010 1762   4586  4759
B@DSNS  001  0018 1714
B@DSS1  001  0000 1713
B@DSTP  001  0054 1756
B@DTBN  001  0010 1778   0803
B@DTB1  001  0050 1777   0803
B@DTCY  001  0009 1774
B@DTSN  001  0010 1776
B@DTS1  001  0040 1775
B@DTYP  001  0040 1890
B@DURE  001  0020 1608
B@DVCY  001  0007 1771   0909
B@DVC1  001  0056 1772   5614  6848  0910  0983  1018  1053
B@DWCY  001  0005 1768
B@DWT1  001  0003 1769
B@D1MK  001  0080 1962   3719
B@D2MK  001  00C0 1963   3744
B@EOST  001  001E 1784   3361  3753  3902  3936  4298  5691  6310  6331  6352  6552  6602  7346
                         7609  7729  7908  8227  8597  9145  9154  9554  9917  0219  0375
B@EQUL  001  007E 1810   4542  5196  7230  8064  8899  8994  9006  9009  9304  9401  9413  9416
B@EXPC  001  00C5 1787
B@FOFL  001  005C 1789
B@FVAD  001  0001 1974   6848  6875* 1040  1060* 1139
B@GETC  001  0001 1913
B@GETE  001  00FF 1914
B@GETS  001  0000 1912   4051  4058  4111  4794  5040  5203  5677  5934  6201  7509  7864  7888
                         8210  8579  8907  9119  9312  9535  9915  0373
B@GRTR  001  006E 1807   5198  8901  8997  9003  9009  9306  9404  9410  9416
B@ICON  001  0050 1869   5649  5671
B@LADD  001  0001 1513
B@LADF  001  0002 1554   7859  8201  8574  9530  9903  0361
B@LADV  001  0008 1998   2019
B@LBIN  001  0002 1923   1924  1930
B@LBNX  001  0003 1547   6519  0180
B@LBRA  001  0003 1545   3279  3368  5412  6958  6969  6976  7107  7155  7191  7493  8426  8736
B@LBRC  001  0004 1544   5259  8954  9360
B@LBRD  001  0003 1546   6926
B@LBRS  001  0001 1548   3517  7532  9759  1247
B@LCCA  001  0004 1954   1005  1086  1116  1116
B@LCCC  001  0001 1506   1544  5278  5279  8976  8982  9383  9389
B@LCDV  001  0004 1999   2020
B@LCER  001  0001 1504   1568
B@LCFN  001  0004 1955   1040  1087  1120  1120
B@LCLN  001  0002 1509   1560  1561  1568  5267  7478  7591  8962  9368
B@LCLS  001  0001 1557   0368
B@LCMC  001  0001 1543   5251  9353
B@LCMF  001  0001 1542   8947
B@LCNA  001  0006 1953   0970  1085  1108  1112  1112
```

```
B@LCNN  001  0001 1507   1532  1541  1553  1565  3496  3533  5994  5996  6373  6950  6952  8059
                         8083  8090  9786
B@LCOP  001  0001 1503   1511  1512  1513  1514  1515  1516  1517  1518  1519  1520  1521  1522
                         1523  1524  1525  1526  1527  1528  1529  1530  1531  1532  1533  1534
                         1535  1536  1537  1538  1539  1540  1541  1542  1543  1544  1545  1546
                         1547  1548  1549  1550  1551  1552  1553  1554  1555  1556  1557  1558
                         1559  1560  1561  1562  1563  1564  1565  1566  3409  3412  3415  3532
                         3535  3538  4129  4132  4146  4150  4154  4158  4162  4166  4170  4174
                         4309  4577  4580  4582  4731  4734  4737  4740  4743  4746  4749  4752
                         4755  4878  4883  4886  4889  4892  4895  5169  5172  5276  5280  5441
                         5444  5605  5719  5722  5725  5987  5990  5993  6002  6380  6383  6682
                         6685  6688  6691  6946  6949  6960  7216  7219  7222  7225  7228  7357
                         7590  7593  7596  7599  7602  7739  7919  7922  8089  8238  8241  8462
                         8608  8611  8757  8973  8974  9165  9168  9379  9381  9565  9568  9780
                         9783  9785  9928  9931  0054  0237  0240  0243  0246  0386  0389  0510
                         0920  0921  1258  1371
B@LCRV  001  0013 1997   2017  6374  6700  7610
B@LCSA  001  0002 1541   9752
B@LCVA  001  0002 1505   1519  1520  1521  1522  1523  1524  1525  1526  1527  1528  1530  1531
                         1533  1534  1535  1536  1537  1538  1539  1544  1545  1546  1547  1549
                         1550  1551  1563  1564  3410  3413  3416  4130  4133  4147  4151  4155
                         4159  4163  4167  4171  4175  4310  5277  5442  5445  5605  5720  5723
                         5726  5988  6003  6683  6686  6692  6947  6961  7217  7220  7223  7226
                         7358  7594  7740  8242  8612  8758  8975  9166  9169  9382  9569  9781
                         9792  0238  0241  0244
B@LCXX  001  0001 1508   1540  1552  1554  1558  1559  3536  6381  6689  7597  7923  0247
B@LDAT  001  0004 1667   5595
B@LDCA  001  0003 1563
B@LDDL  001  0003 1564
B@LDDM  001  0004 1927
B@LDEF  001  0003 1668   6825
B@LDIM  001  0003 1669   3661
B@LDIN  001  0004 1926   1927  1928
B@LDIV  001  0001 1516
B@LDMN  001  0002 1924   1953  1954  1966  1967  1968  1971  1998  1999  3702  3729  3748  0990
                         1025
B@LDSN  001  0004 1928
B@LDWA  001  0002 1565   6000  6018  6958  6969  6976
B@LELP  001  0010 1996   5947  5997  6018
B@LEND  001  0003 1696
B@LEOF  001  0001 1566   0780
B@LEOP  001  0001 1562
B@LERC  001  0003 1568
B@LESP  001  0008 1995   5996  6000
B@LESS  001  004C 1797   9000  9003  9006  9407  9410  9413
B@LET$  001  005B 1817
B@LET#  001  007B 1818
B@LET@  001  007C 1819
B@LETA  001  00C1 1821
B@LETB  001  00C2 1823
B@LETC  001  00C3 1824
B@LETD  001  00C4 1825
B@LETE  001  00C5 1826
B@LETF  001  00C6 1827
B@LETG  001  00C7 1828
B@LETH  001  00C8 1829
```

```
B@LETI  001  00C9 1830
B@LETJ  001  00D1 1831
B@LETK  001  00D2 1832
B@LETL  001  00D3 1833
B@LETM  001  00D4 1834
B@LETN  001  00D5 1835
B@LETO  001  00D6 1836
B@LETP  001  00D7 1837
B@LETQ  001  00D8 1838
B@LETR  001  00D9 1839
B@LETS  001  00E2 1840
B@LETT  001  00E3 1841
B@LETU  001  00E4 1842
B@LETV  001  00E5 1843
B@LETW  001  00E6 1844
B@LETX  001  00E7 1845
B@LETY  001  00E8 1846
B@LETZ  001  00E9 1847
B@LEXP  001  0008 1886
B@LFCI  001  0003 1521
B@LFNA  001  0002 2000   2021
B@LFN0  001  0003 1519   5070
B@LFN1  001  0003 1520
B@LFOR  001  0003 1549   6000  6014  6018
B@LFRT  001  0004 1941   1942  5959  6011  8455
B@LGET  001  0003 1551   3355  4292  8221
B@LGSB  001  0005 1675   5392
B@LGTO  001  0004 1674   8725  9690
B@LHLT  001  0001 1512   0043
B@LIEX  001  0002 1872   5750  5765
B@LIFN  001  0003 1935   4072  4077  4145  4149  4153  4157  4161  4165  4169  4173
B@LILP  001  0009 1994   2012  2013  2014  6862  6954  6976
B@LIMG  001  0001 1686   7486
B@LIMH  001  0003 1561   7481
B@LINI  001  0002 1553   3510
B@LINP  001  0005 1681   3260
B@LIPI  001  0003 1875   5755  5770
B@LISP  001  0005 1993   2001  2007  2008  2009  6952  6958
B@LIS2  001  0005 1878   5760  5775
B@LIVT  001  0001 1951
B@LKCL  001  0005 1680   0353
B@LKFR  001  0003 1671   5905
B@LKGT  001  0003 1677   8191
B@LKIF  001  0002 1673   5025  8885  9290
B@LKON  001  0002 1706   9741
B@LKPT  001  0003 1678   7849
B@LKPU  001  000A 1685   6502
B@LKRR  001  0007 1683
B@LKRT  001  0005 1679   9895
B@LKTO  001  0002 1700   5916
B@LLET  001  0003 1670   4436  4567  4569  4611  4692  4694  4816  4870  4872  5055  7094  8044
B@LL01  001  0002 2038   2039
B@LL02  001  0002 2041   2042
B@LL03  001  0002 2044   2045
B@LL04  001  0002 2047   2048
B@LL05  001  0002 2050   2051
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 204
B@LL06  001  0002 2053   2054
B@LL07  001  003A 2056   2057  0946  0946
B@LL08  001  0100 2059   2060  0950  0950  0951  0952
B@LL09  001  0100 2062   2063  0951  0951  0952
B@LL10  001  0044 2065   2066  0952  0952
B@LL11  001  003A 2068   2069  0956  0956
B@LL12  001  003A 2071   2072  0978
B@LL13  001  003A 2074   2075  1013
B@LL14  001  003A 2077   2078  1048
B@LL15  001  0100 2080   2081  0960
B@LL16  001  0096 2083   2084  0961
B@LMAT  001  0003 1687   3885
B@LMF1  001  0003 1522   4079  7340  7723  8591  9131  9139  9548  0213
B@LMF2  001  0003 1523   4121
B@LMF3  001  0003 1524
B@LMGT  001  0006 1688   8565
B@LMIN  001  0008 1689   7328
B@LMPR  001  0008 1692   9114
B@LMPT  001  0006 1691   9521
B@LMPU  001  000D 1693   0161
B@LMPY  001  0001 1515
B@LMRD  001  0007 1690   7711
B@LMSM  001  0003 1525   4101
B@LNEG  001  0001 1518
B@LNEX  001  0004 1672   8387
B@LNXT  001  0003 1550   6000  6018
B@LPAR  001  004D 1798   3895  4619
B@LPRS  001  0002 1558   6278
B@LPRT  001  0005 1684   6135
B@LPRU  001  0002 1559   6668  7525  7564  0226
B@LPSE  001  0005 1694
B@LPUT  001  0002 1552   7902
B@LPWR  001  0001 1517
B@LREA  001  0004 1682   4276
B@LREM  001  0003 1666
B@LRSR  001  0001 1555   1360
B@LRST  001  0001 1556   9910
B@LRTN  001  0006 1676
B@LSA1  001  0003 1537
B@LSA2  001  0003 1538
B@LSB1  001  0003 1539
B@LSC1  001  0003 1531
B@LSDF  001  0004 1921
B@LSD0  001  0003 1533
B@LSD1  001  0003 1534
B@LSD2  001  0003 1535
B@LSF1  001  0003 1527
B@LSF2  001  0003 1528
B@LSKW  001  0002 1937
B@LSNO  001  0002 1930   0900
B@LSPT  001  0003 1945   1948
B@LSTA  001  0003 1536   3267  5404  6508  7173  9697  9714  0168
B@LSTC  001  0003 1530   6236  6632  7557
B@LSTE  001  0004 1701
B@LSTF  001  0003 1526   5939  7131
B@LSTH  001  0003 1560   7608
```

```
B@LSTP  001  0004 1695
B@LSTX  001  0002 1540   3491  4636  4825  5064
B@LSUB  001  0001 1514
B@LSVC  001  0001 1511   0499  0780
B@LTHN  001  0004 1702   5109  8936  9342
B@LTYP  001  0001 1931
B@LUFN  001  0002 1938
B@LUSC  001  0002 1532   4540  4832  8075
B@LUSF  001  0001 1529   4712  7135  7182
B@LVPG  001  0100 2025   2028
B@MINS  001  0060 1804   3917  5762  5767  5772
B@MULT  001  005C 1801   3919
B@NAAR  001  001D 1989   2019  2071  1112
B@NCAR  001  001D 1990   2020  2074  1116
B@NCRV  001  001D 1988   2017  2068
B@NDGT  001  000A 1981   1987
B@NEQL  001  007F 1811   9012  9419
B@NFRT  001  000A 1940   1942
B@NICN  001  0006 1983   1985
B@NIEL  001  0007 1985   2001  2007  2012
B@NIFN  001  0018 1934
B@NIVR  001  0001 1984   1985
B@NIVT  001  0057 1950   3297  3445
B@NLDV  001  0122 1987   2009  2014  2065
B@NLRV  001  001D 1986   2008  2013  2056
B@NLTR  001  001D 1980   1986  1987  1988  1989  1990  1991
B@NSKW  001  0004 1936
B@NSPT  001  0028 1944
B@NUFN  001  001D 1991   2021  2077  1120
B@NVPG  001  0100 2024   2028
B@NXHI  001  00E3 1905
B@NXLO  001  001E 1904
B@NXZR  001  0080 1903   1904  1905
B@PLUS  001  004E 1799   3915  5648  5747  5752  5757
B@POWR  001  005A 1800
B@PREC  001  0020 1892
B@PROD  001  0023 2001
B@PRPL  001  0002 1588   6303
B@PRPN  001  0001 1587   6227  6315  6336  6349  6357
B@PRPR  001  0004 1590   6311
B@PRPS  001  0003 1589   6307
B@PRRC  001  0007 1593   6332  6353
B@PRRL  001  0008 1594   6224
B@PRSL  001  0005 1591   6324  6345
B@PRSS  001  0006 1592   6328
B@PTAB  001  0000 1946
B@PTAD  001  0001 1947
B@PTSA  001  0002 1948
B@PUD1  001  0006 1604   6592  6627
B@PUD2  001  0007 1605   6647
B@PUI0  001  0001 1598   7519
B@PUI1  001  0004 1599   7551
B@PUI2  001  0005 1600   7571
B@PUNL  001  0002 1602   6557
B@PUNS  001  0003 1603   6618
B@PUTM  001  0010 1607   6561  0247
```

```
B@RPAR  001  005D 1802   3712  4631  4822  5060  6889
B@SADV  001  00E8 2019   2022
B@SAVL  001  0B76 2015   2032
B@SAVS  001  065E 2010   2031
B@SCDV  001  0074 2020   2022
B@SCLN  001  005E 1803   6306  6327  6348  9124
B@SCRV  001  0227 2017   2031  2032
B@SDMK  001  0080 1932
B@SEXP  001  0004 1885
B@SFAT  001  0196 2022   2031  2032  2083
B@SFNA  001  003A 2021   2022
B@SFRT  001  0028 1942
B@SIEL  001  003F 2012   2015
B@SIES  001  0023 2007   2010
B@SIGN  001  0010 1894
B@SLDL  001  0A32 2014   2015
B@SLDS  001  05AA 2009   2010
B@SLVL  001  0105 2013   2015
B@SLVS  001  0091 2008   2010
B@SQUO  001  007D 1809   4792  5035  5641
B@STAT  001  0000 1884
B@TASA  001  0012 1619
B@TASC  001  001E 1625
B@TASM  001  0018 1621
B@TASS  001  007B 1626
B@TCGT  001  0030 1634
B@TCLS  001  0042 1640
B@TDAT  001  0006 1615
B@TDEF  001  0009 1616
B@TDIM  001  000C 1617
B@TDUM  001  0078 1658
B@TEND  001  0072 1656
B@TEOF  001  0075 1657
B@TFOR  001  0021 1628
B@TGET  001  0039 1637
B@TGSB  001  0033 1635
B@TGTO  001  002D 1633
B@TIFA  001  0027 1630
B@TIFC  001  002A 1631
B@TIFS  001  007D 1632
B@TIMG  001  0054 1646
B@TINP  001  0045 1641
B@TLTA  001  000F 1618
B@TLTC  001  001B 1622
B@TLTM  001  0015 1620
B@TLTS  001  0079 1623
B@TMAS  001  007C 1627
B@TMAT  001  0057 1647
B@TMGT  001  005A 1648
B@TMIN  001  005D 1649
B@TMLS  001  007A 1624
B@TMPR  001  0066 1652
B@TMPT  001  0063 1651
B@TMPU  001  0069 1653
B@TMRD  001  0060 1650
B@TNXT  001  0024 1629
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 207
B@TPRT  001  004E 1644
B@TPRU  001  0051 1645
B@TPSE  001  006C 1654
B@TPUT  001  003C 1638
B@TRAC  001  0080 1888
B@TREA  001  0048 1642
B@TREM  001  0003 1614
B@TRSR  001  004B 1643
B@TRST  001  003F 1639
B@TRTN  001  0036 1636
B@TSTP  001  006F 1655
B@VMC1  001  0056 2027
B@VMLB  001  F0CD 2032
B@VMSB  001  F5E5 2031
B@VMSZ  001  0000 2028   2030  2031  2032
B@VMTB  001  0000 2030
B@ZNEG  001  00D0 1901
B@ZPOS  001  00F0 1900
BITAD2  001  0FE7 5155   5139
BITBLS  002  0FEF 5160   5147
BITBN1  002  0FF3 5162
BITBRC  001  1086 5276   5257
BITB01  002  1088 5277
BITB02  001  1089 5278   5256*
BITCA2  002  0FE8 5156   5027* 5121  5129  5135* 5140* 5146
BITCMC  001  108C 5280   5249
BITEN2  001  0006 5281   5227
BITERM  001  104A 5248
BITFCP  002  0FEB 5158   5127* 5128* 5129
BITFNO  001  0FF8 5172   5068
BITFPE  002  0FED 5159   5127
BITLNG  002  108B 5279   5266
BITLSW  001  0FF4 5166   5018* 5026* 5093* 5094
BITOOP  002  0FFA 5173
BITPBA  002  0FF1 5161   5121  5135
BITREL  001  1000 5186
BITRE1  001  0F06 5024
BITSG2  001  0000 5153   5148
BITSTX  001  0FF6 5169   5062
BITTRM  001  004A 5154   5111
BIT001  001  0FF5 5167   5093
BIT100  003  0F0D 5027   5019
BIT110  004  0F25 5043   5036
BIT120  004  0F64 5067   5061
BIT140  003  0F68 5068   5066
BIT150  004  0F73 5071
BIT160  003  0F7E 5082   5047
BIT200  004  0F95 5093   5042  5073  5082
BIT240  004  101F 5209   5197  5199
BIT260  004  1023 5215   5204
BIT270  003  1027 5216   5218
BIT280  003  102A 5217   5191* 5209*
BIT290  003  1043 5239   5228
BIT300  004  0FA9 5109   5095
BIT340  004  0FB8 5121   5104
BIT350  004  0FBF 5127
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                            VER 15, MOD 00  20/07/20  PAGE 208
BIT360  004  0FCF 5135   5122
BIT370  003  0FD3 5139
BIT380  003  0FDE 5146   5130
BIT390  003  0FE4 5148   5103* 5111* 5140
BKABRC  001  1A87 8974   8952
BKAB01  002  1A89 8975
BKAB02  001  1A8A 8976   8925*
BKACMC  001  1A86 8973   8945
BKALNG  002  1A8C 8982   8961
BKALTH  001  0002 8991   8919  8992
BKAOD1  001  0000 8989   8920
BKAOD2  001  0001 8990   8925
BKAOT1  001  1A8B 8992   8918
BKARIF  001  1A00 8880
BKATAB  001  1A8D 8988   8992
BKA010  004  1A00 8885
BKA020  004  1A08 8890
BKA030  004  1A0C 8894
BKA040  004  1A10 8898
BKA050  004  1A20 8907
BKA060  004  1A27 8913   8900  8902
BKA070  003  1A2B 8918   8908
BKA080  003  1A2E 8919   8921
BKA090  003  1A31 8920   8894* 8913*
BKA100  004  1A37 8925
BKA110  004  1A3B 8930
BKA120  004  1A43 8936
BKA130  004  1A4B 8941
BKA140  003  1A4F 8945
BKA150  003  1A5E 8952
BKA160  006  1A6D 8960
BKA170  004  1A82 8967
BKCBO1  002  1B89 9382
BKCBO2  001  1B8A 9383   9331*
BKCBRC  001  1B87 9381   9358
BKCCD2  001  0001 9397   9331
BKCCMC  001  1B86 9379   9351
BKCLNG  002  1B8C 9389   9367
BKCLTH  001  0002 9398   9325  9399
BKCOD1  001  0000 9396   9326
BKCOTB  001  1B8B 9399   9324
BKCRIF  001  1B00 9285
BKCTAB  001  1B8D 9395   9399
BKC010  004  1B00 9290
BKC020  004  1B08 9295
BKC030  004  1B0C 9299
BKC040  004  1B10 9303
BKC050  004  1B20 9312
BKC060  004  1B27 9318   9305  9307
BKC070  003  1B2B 9324   9313
BKC080  003  1B2E 9325   9327
BKC090  003  1B31 9326   9299* 9318*
BKC100  004  1B37 9331
BKC110  004  1B3B 9336
BKC120  004  1B43 9342
BKC130  004  1B4B 9347
```

```
                                        CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 209
BKC140  003  1B4F 9351
BKC150  003  1B5E 9358
BKC160  006  1B6D 9366
BKFBN2  002  12E7 6012
BKFDAC  001  12BE 5993
BKFDAN  001  12BF 5994   5947* 5995
BKFLLP  001  0027 6018   5946
BKFLSP  001  0001 6019   5926
BKFOCV  001  0001 6020   5974*
BKFOC1  001  12E8 6013   5933
BKFOFA  001  12E0 5998   5946* 5951  5999
BKFOFC  001  12B8 5987   5952
BKFOFO  002  12BA 5988   5911* 5974
BKFONC  001  12BB 5990
BKFOND  001  0003 6021   5975* 5976* 5977*
BKFONO  002  12BD 5991
BKFOPR  032  12DF 5997
BKFORX  001  1200 5901
BKFOSC  001  12E1 6002   5937
BKFOSO  002  12E3 6003   5936*
BKFOTL  002  12E5 6011   5966
BKFOX3  002  12EA 6014   5977
BKF010  004  1200 5905
BKF020  004  1208 5910
BKF030  004  1211 5915
BKF040  004  122F 5926
BKF050  003  123E 5933   5922
BKF060  004  125D 5944   5929
BKF070  005  126A 5951   5945
BKF080  004  127A 5958
BKF090  005  128E 5966
BKF100  004  12A2 5974   5962
BKF120  004  12B4 5981   5970
BKGBN1  002  19EB 8764   8743
BKGBRC  001  19E7 8757   8734
BKGBRO  002  19E9 8758
BKGOTO  001  19B3 8721
BKG010  004  19B3 8725
BKG020  004  19BB 8730
BKG030  003  19BF 8734
BKG040  006  19CE 8741
BKG050  004  19DF 8747
BKG060  004  19E3 8751
BKMBN1  002  1CA3 9792   9720  9729  9766
BKMBRC  001  1C9F 9783   9757
BKMCSC  001  1CA0 9785   9750
BKMCSO  001  1CA1 9786   9707* 9729*
BKMGTO  001  1C00 9686
BKMSTC  001  1C9C 9780   9695  9712
BKMSTO  002  1C9E 9781
BKMVAD  002  1CA5 9793   9702* 9765
BKM010  004  1C00 9690
BKM020  003  1C08 9695
BKM030  005  1C17 9702
BKM035  004  1C1C 9706
BKM040  004  1C23 9708   9737
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 210
BKM050  003  1C27 9712
BKM060  006  1C36 9719
BKM070  006  1C41 9724
BKM080  004  1C4B 9729
BKM090  004  1C4F 9733
BKM100  004  1C60 9741   9735
BKM110  004  1C68 9746
BKM120  003  1C6C 9750
BKM125  003  1C7B 9757
BKM130  005  1C8A 9765
BKM140  004  1C94 9770
BKM150  004  1C98 9774
BKNBRC  001  1962 8462   8424
BKNBRO  002  1964 8463   8423* 8436
BKNDUM  001  0000 8450   8406
BKNEXT  001  1900 8383
BKNEX2  002  1961 8456   8437
BKNFEL  002  195F 8455   8431
BKNFTD  001  0001 8449   8397  8406
BKNNXT  001  0003 8451   8423
BKN010  004  1900 8387
BKN020  004  1908 8392
BKN030  004  190C 8396
BKN040  004  1918 8402
BKN050  003  191C 8406
BKN060  004  1922 8411
BKN070  004  1929 8416   8407
BKN080  004  192D 8417   8412
BKN090  004  1934 8423   8398
BKN100  005  1947 8431
BKN110  004  194C 8435
BKN120  004  195A 8441   8418
BKRBRC  001  1FE2 1258   1245
BKRTRN  001  1FCF 1241
BKR010  003  1FCF 1245
BKR020  004  1FDE 1252
BKSBN1  002  10ED 5451   5420  5426
BKSBRC  001  10E9 5444   5410
BKSBRO  002  10EB 5445
BKSTAC  001  10E6 5441   5402
BKSTAO  002  10E8 5442
BKSUBG  001  1090 5388
BKSVAS  002  10EF 5457   5406* 5425
BKS010  004  1090 5392
BKS020  004  1098 5397
BKS030  003  109C 5402
BKS040  003  10B0 5410
BKS050  006  10BF 5418
BKS060  005  10D4 5425
BKS070  004  10DE 5431
BKS080  004  10E2 5435
BMDM1C  001  1AEA 9165   9129
BMDM10  002  1AEC 9166
BMDM2C  001  1AED 9168   9137
BMDM20  002  1AEF 9169
BMDPRT  001  1A9B 9110
```

```
BMD010  004  1A9B 9114
BMD020  004  1AA3 9119   9155
BMD030  003  1AAB 9124
BMD040  003  1AB1 9129
BMD050  003  1AC3 9137   9125
BMD055  004  1AD2 9144
BMD060  004  1ADC 9150   9133
BMD070  003  1AE0 9154
BMD080  004  1AE6 9159   9146
BMGAFC  001  19AC 8608   8572
BMGAFO  001  19AD 8609
BMGBN1  002  19B2 8621
BMGETX  001  1965 8561
BMGMFC  001  19AE 8611   8589
BMGMFO  002  19B0 8612
BMGSFA  001  19B1 8620
BMG010  004  1965 8565
BMG100  003  1971 8572
BMG110  004  1980 8579
BMG120  004  1988 8584   8598
BMG140  003  198F 8589
BMG150  004  199E 8596   8585*
BMG160  004  19A8 8602
BMIMFC  001  16FC 7357   7338
BMIMFO  002  16FE 7358
BMINPT  001  16D3 7324
BMI010  004  16D3 7328
BMI020  004  16DB 7334   7347
BMI030  003  16DF 7338
BMI040  004  16EE 7345
BMI050  004  16F8 7351
BMMAD2  001  0AF3 4019   3996
BMMATA  001  0A00 3879   4041
BMMAT2  001  0B00 4043   3904  3905  3924  3925  3930  3931  3968  3969
BMMBK0  001  0000 4028   3900* 4052  4065* 4113
BMMBK1  001  0001 4029   3907* 3914  3943  3945
BMMBK2  001  0002 4030   3909* 4059  4066* 4072
BMMBLS  002  0AF0 4009   4002
BMMCA2  002  0AF4 4020   3880* 3977  3986  3992*
BMMFCP  002  0AF7 4023   3983* 3984* 3986
BMMFND  001  0002 4141   4072
BMMFPE  002  0AF9 4024   3983
BMMIA2  001  0AF5 4021
BMMINV  001  00D5 4016   3943
BMMMSC  001  0B99 4129   4099
BMMMSO  002  0B9B 4130
BMMM2C  001  0B9C 4132   4119
BMMM2O  002  0B9E 4133
BMMPBA  002  0AF2 4012   3904* 3924* 3930* 3968* 3977  3992
BMMPID  001  0003 4140
BMMPSI  001  0004 4033   4021
BMMSG2  001  0000 4032
BMMTAB  001  0B9F 4142   4143
BMMTBS  001  0B99 4143   4070
BMMTEL  001  0006 4139   4071  4143
BMMTRN  001  00D9 4017   3945
```

```
BMM005  005  0A3D 3907   3903
BMM010  003  0A65 3924   3916  3918
BMM020  003  0A6E 3930   3920
BMM030  004  0A85 3943   3937
BMM040  004  0A93 3950   3944
BMM050  004  0AA2 3957   3946
BMM060  003  0AC1 3968   3896
BMM070  004  0AC7 3977   3906  3926  3939  3953  3964
BMM080  004  0ADE 3992   3978
BMM090  003  0AE9 4002   3987
BMM095  003  0AEC 4003   3905* 3925* 3931* 3969*
BMM100  004  0B00 4047   3924  3925
BMM110  003  0B2C 4070   3930  3931
BMM120  003  0B2F 4071   4073
BMM130  003  0B3A 4077
BMM140  004  0B4C 4089   3968  3969
BMM150  004  0B6B 4106   4084  4123
BMM160  004  0B6F 4110   3904  3905
BMPAFC  001  1BE2 9565   9528
BMPAFO  001  1BE3 9566
BMPBN1  002  1BE8 9577
BMPMFC  001  1BE4 9568   9546
BMPMFO  002  1BE6 9569
BMPSFA  001  1BE7 9575
BMPUTX  001  1B9B 9517
BMP010  004  1B9B 9521
BMP100  003  1BA7 9528
BMP110  004  1BB6 9535
BMP120  004  1BBE 9541   9555
BMP130  003  1BC5 9546
BMP140  004  1BD4 9553   9542*
BMP150  004  1BDE 9559
BMREAD  001  17D0 7707
BMRMFC  001  17F9 7739   7721
BMRMFO  002  17FB 7740
BMR010  004  17D0 7711
BMR020  004  17D8 7716   7730
BMR030  003  17DC 7721
BMR040  004  17EB 7728
BMR050  004  17F5 7734
BMUBNC  001  1D8B 0240   0178
BMUBN1  002  1D94 0251   0174  0193
BMUMFC  001  1D8E 0243   0211
BMUMFO  002  1D90 0244
BMUPRC  001  1D91 0246   0224
BMUPRO  001  1D92 0247
BMUPRT  001  1D00 0157
BMURNO  002  1D8D 0241
BMUSTC  001  1D88 0237   0166
BMUSTO  002  1D8A 0238
BMU010  004  1D00 0161
BMU020  003  1D08 0166
BMU030  006  1D17 0173
BMU040  003  1D22 0178
BMU050  006  1D35 0187
BMU060  006  1D3F 0192
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 213
BMU070  004  1D4A 0197
BMU080  006  1D4E 0201
BMU090  004  1D58 0207   0220
BMU100  003  1D5C 0211
BMU110  004  1D6B 0218
BMU120  003  1D75 0224
BMU130  004  1D84 0231
BNABNI  002  09F7 3761
BNADIN  001  0973 3657
BNA010  004  0973 3661
BNA020  004  097B 3666   3754
BNA030  004  097F 3670
BNA040  003  098A 3676
BNA060  004  099C 3688   3678
BNA070  004  09A0 3692
BNA080  004  09A4 3696
BNA090  004  09AB 3698   3672* 3714  3724  3739
BNA100  005  09AF 3702
BNA110  003  09B4 3706
BNA120  004  09BA 3711   3697
BNA130  003  09CD 3724   3713
BNA140  005  09D0 3729
BNA150  004  09D5 3733
BNA160  004  09D9 3737
BNA170  003  09E0 3744
BNA180  005  09E3 3748   3720
BNA190  004  09E8 3752   3707
BNDATA  001  1100 5591
BNDBKL  001  0002 5741   5658  5744
BNDBKT  001  11DA 5743   5648* 5655* 5658  5669* 5676
BNDBK0  001  0000 5732   5648* 5669* 5676
BNDBK1  001  0001 5733   5655* 5658
BNDBN1  001  11FA 5781   5703  5708
BNDBRC  001  11D1 5719   5599
BNDBRO  002  11D3 5720
BNDDAC  001  11D4 5722   5632
BNDDAO  002  11D6 5723   5662* 5686*
BNDDLC  001  11D7 5725   5696
BNDDLO  002  11D9 5726
BNDICA  001  0000 5740   5662
BNDTAB  001  11DC 5746   5656
BNDTB1  001  0001 5736   5658
BNDTB3  001  0003 5737   5661
BNDTB4  001  0004 5738   5660
BNDTEL  001  0005 5735   5656  5657
BND010  004  1100 5595
BND020  003  1104 5599
BND030  006  1113 5610
BND040  004  1119 5614
BND050  006  1120 5620
BND060  006  1129 5626   5615
BND070  003  1133 5632   5621
BND080  004  113A 5637   5692
BND090  003  113E 5641
BND100  003  114B 5648   5642
BND110  004  1154 5654   5672
```

```
BND120  003  115F 5657   5659
BND130  004  1180 5669   5650
BND170  004  1195 5681   5644
BND180  005  1199 5686
BND190  003  11A2 5691   5665
BND200  003  11A8 5696
BND210  006  11B3 5702
BND220  006  11BE 5707
BND230  004  11CD 5713
BNFBDC  001  15CB 6960   6924
BNFBDO  002  15CD 6961   6868* 6869* 6875  6908
BNFBN1  001  15CF 6970   6932
BNFBRC  001  15BC 6946   6830
BNFBRO  002  15BE 6947
BNFDAC  001  15BF 6949
BNFDAN  001  15C0 6950   6862* 6951
BNFDEF  001  1500 6821
BNFLIP  001  000D 6976   6861
BNFLTH  001  15CE 6969   6909
BNFSKP  001  0002 6974   6895
BNFSPA  001  15CA 6956   6861* 6866  6957
BNFWKA  009  15C9 6954
BNF010  004  1500 6825
BNF020  003  1508 6830
BNF030  006  1513 6837
BNF040  004  1519 6842
BNF050  004  1521 6847
BNF060  004  152B 6853
BNF070  004  1537 6859   6849
BNF080  005  1544 6866   6860
BNF090  004  1557 6874
BNF100  004  155F 6879
BNF110  005  1563 6884
BNF120  003  156C 6889
BNF130  005  1572 6894
BNF140  004  1582 6902   6890
BNF150  005  158A 6908   6897
BNF160  004  1594 6914
BNF170  004  1598 6918
BNF180  003  15A0 6924
BNF190  005  15AF 6932
BNF200  004  15B4 6936
BNF210  004  15B8 6940
BNIBN1  002  17CB 7607   7499  7579
BNIBRC  001  17C1 7593   7491
BNIBRO  002  17C3 7594
BNIBSC  001  17C9 7602   7530
BNIEOS  001  17CD 7609   7503
BNIIHO  002  17C0 7591   7478*
BNIIMH  001  17BE 7590   7479
BNIMAG  001  1700 7468
BNIPRC  001  17C4 7596   7523  7562
BNIPRO  001  17C5 7597   7519* 7551* 7571*
BNISHL  001  17CC 7608   7476  7477
BNISTC  001  17C6 7599   7555
BNISTO  002  17C8 7600   7546* 7575*
```

```
BNISUB  002  17CF 7610   7575
BNI005  004  1725 7486   7475
BNI010  003  172D 7491
BNI020  006  173C 7498
BNI030  003  1747 7503
BNI040  004  174A 7507
BNI050  004  1756 7513
BNI060  003  175D 7519
BNI070  003  1760 7523
BNI080  003  176F 7530   7584
BNI090  004  177E 7537
BNI100  004  1782 7541
BNI110  005  1786 7546   7514
BNI120  003  178B 7551
BNI130  003  178E 7555   7580
BNI140  003  179D 7562
BNI150  003  17AC 7571
BNI160  004  17AF 7575
BNI170  005  17B3 7579
BNI180  003  17BB 7584
BPCASN  001  1871 8049
BPCBN1  001  18A0 8083   8059
BPCLET  001  1869 8040
BPCUCC  001  18A1 8089   8073
BPCUCO  001  18A2 8090   8054* 8059*
BPC010  004  1869 8044
BPC020  003  1871 8054
BPC030  004  1874 8058   8067
BPC040  003  187C 8064
BPC050  004  1889 8072   8065
BPMASN  001  1608 7099
BPMBIC  001  16C5 7216   7105  7153
BPMBIO  002  16C7 7217
BPMBN1  002  16C4 7210   7162  7198
BPMBRC  001  16C8 7219   7189
BPMBRO  002  16CA 7220   7114* 7161
BPMIND  001  16D2 7230   7140
BPMLET  001  1600 7090
BPMSAC  001  16CB 7222   7171
BPMSAO  002  16CD 7223   7170*
BPMSFC  001  16CE 7225   7129
BPMSFO  002  16D0 7226   7118* 7170
BPMUFC  001  16D1 7228   7133  7180
BPM010  004  1600 7094
BPM020  003  1608 7105
BPM030  005  1617 7114
BPM040  005  161C 7118
BPM045  004  1621 7122
BPM050  004  1625 7123   7147
BPM060  003  162D 7129
BPM070  003  164B 7140   7124*
BPM080  004  1651 7146
BPM090  003  1658 7153   7141
BPM100  005  1667 7161
BPM110  004  167B 7170
BPM120  004  168E 7179
```

```
BPM130  003  16A1 7189
BPM140  006  16B0 7197
BPM150  004  16BF 7204
BPREAD  001  0BCF 4272
BPRGTC  001  0BFC 4309   4290
BPRGTO  002  0BFE 4310
BPR010  004  0BCF 4276
BPR020  004  0BD7 4281   4299
BPR030  004  0BDB 4285
BPR040  003  0BDF 4290
BPR050  004  0BEE 4297
BPR060  004  0BF8 4303
BPXRSC  001  1FF6 1371   1358
BPXRSR  001  1FE3 1354
BPX010  003  1FE3 1358
BPX020  004  1FF2 1365
BRA050  004  0990 3682
BSTRAS  001  0C1B 4444
BSTRIF  001  0F00 5011   5178
BSTRLT  001  0C00 4425   4604  4765
BST010  004  0C0F 4430   4429*
BST020  004  0C13 4436   4428
BST080  003  0C1E 4453
BST100  004  0C2E 4461   4545
BST120  004  0C3A 4473
BST130  003  0C4B 4486   4474
BST131  003  0C62 4496   4486
BST132  005  0C70 4504   4481
BST134  004  0C7C 4507   4554
BST136  004  0C92 4522   4508
BST138  003  0C9D 4529   4516
BST140  003  0CA6 4539   4499  4505
BST145  003  0CBC 4550   4543
BST150  003  0CCF 4565   4454  4498  4541
BST160  004  0CD6 4567   4540* 4569*
BST170  004  0CE5 4571   4565*
BST200  004  0D00 4611
BST210  004  0D27 4630   4655
BST220  003  0D38 4635   4632
BST230  003  0D41 4643   4634
BST240  003  0D55 4653   4620
BST250  005  0D5F 4661   4648
BST260  004  0D70 4669   4667*
BST270  004  0D74 4675   4662
BST300  003  0D83 4690   4626  4629  4637  4645  4647  4710  4714  4717  4720  4722  4724
BST310  004  0D8A 4692   4636* 4694* 4712*
BST320  004  0D95 4695   4690*
BST340  003  0D99 4706   4653
BST360  004  0DD9 4725   4706*
BST400  003  0E00 4773
BST410  004  0E3B 4797   4793
BST440  003  0E4E 4806
BST460  004  0E51 4807   4820
BST500  004  0E65 4816   4801
BST540  004  0E77 4821   4819
BST545  004  0E8D 4828   4823
```

```
BST547  003  0E91 4829   4827
BST550  003  0EC2 4868   4774  4790  4826  4830  4833  4844
BST560  004  0EC9 4870   4825* 4832* 4872*
BST570  004  0ED4 4873   4868*
BST600  003  0E97 4831   4796  4806
BTPAUS  001  1CE7 0037
BTPHTC  001  1CFA 0054   0041
BTP010  003  1CE7 0041
BTP020  004  1CF6 0048
BTRAD2  001  1EFA 0927   0880
BTRBLS  002  1EE7 0893   0886
BTRBND  001  00FF 1128   0791
BTRCA2  002  1EFB 0928   0742* 0863  0870  0876*
BTRCCD  001  1FC6 1138   1025
BTRCCE  001  1FC6 1137   1005* 1138
BTRCCL  002  1FB9 1086   1029
BTRCCP  002  1FCC 1114   1004  1029*
BTRCFA  001  1FC6 1140   1060
BTRCFE  001  1FC6 1139   1040* 1140
BTRCFL  002  1FBB 1087   1064
BTRCFP  002  1FCE 1118   1039  1064*
BTRCND  001  1FC8 1136   0990
BTRCNE  001  1FC8 1135   0970* 1136
BTRCNL  002  1FB7 1085   0994
BTRCNP  002  1FCA 1110   0969  0994*
BTRCTP  004  1F61 1143   1030*
BTRDPA  002  1FB3 1077   1075*
BTRDPL  001  1FBD 1096   1074
BTRECA  002  1EF7 0912
BTRECY  001  1EF3 0909
BTREFN  001  1EF2 0908
BTREOF  001  1EF9 0921
BTREPL  001  1EF2 0907   0754
BTRESA  001  1EF4 0910
BTRESC  001  1EF5 0911
BTRFAC  002  1FB5 1083   0989  1024  1059
BTRFCP  002  1EFE 0931   0869* 0870
BTRFTA  002  1EED 0897   0766
BTRFTP  004  1F8B 1144   1065*
BTRMNT  001  1E00 0741   0941
BTRNTP  004  1F37 1142   0995*
BTRPBA  002  1EEB 0895   0863  0876
BTRPCA  001  1EF8 0918   0778
BTRPSI  001  0004 1127   0929
BTRSA2  001  1EFC 0929
BTRSEL  001  0004 0900   0818  0901
BTRSG2  001  0000 1126   0887
BTRSHA  001  1CFF 0899   0818*
BTRSHE  004  1EF1 0901   0818
BTRSTL  001  1FBC 1089   0995  1030  1065
BTRSVC  001  1EF8 0920
BTRTEN  001  1FC3 1107   1134  1135  1137  1139
BTRVAD  001  1FC4 1134   0976  0983  0988  1011  1018  1023  1046  1053  1058
BTRVBA  002  1EE9 0894   0844
BTR010  004  1E03 0746
BTR020  004  1E0A 0751
```

```
BTR030  004  1E19 0759
BTR040  005  1E2B 0766   0747
BTR050  004  1E33 0771
BTR060  003  1E3F 0778   0767
BTR070  004  1E4E 0786
BTR080  004  1E56 0791
BTR090  004  1E5D 0797
BTR100  004  1E65 0802   0792
BTR110  004  1E70 0808
BTR120  004  1E7C 0814   0804
BTR130  005  1E80 0818
BTR150  006  1E93 0832
BTR160  006  1E99 0836
BTR170  006  1E9F 0840
BTR180  005  1EA5 0844
BTR190  006  1EB0 0850
BTR200  006  1EB6 0854
BTR250  004  1EC2 0863
BTR260  005  1EC9 0869
BTR270  004  1ED5 0876
BTR280  003  1ED9 0880   0864
BTR290  003  1EE0 0886   0871
BTR300  006  1F00 0946
BTR310  006  1F06 0950
BTR320  006  1F18 0956
BTR330  006  1F1E 0960
BTR350  003  1F2A 0969   0996
BTR360  004  1F31 0974
BTR370  004  1F35 0976   0977  0979  1142
BTR380  003  1F39 0983
BTR390  003  1F3F 0988
BTR400  004  1F49 0994   0984
BTR410  003  1F54 1004   1031
BTR420  004  1F5B 1009
BTR430  004  1F5F 1011   1012  1014  1143
BTR440  003  1F63 1018
BTR450  003  1F69 1023
BTR460  004  1F73 1029   1019
BTR470  003  1F7E 1039   1066
BTR480  004  1F85 1044
BTR490  004  1F89 1046   1047  1049  1144
BTR500  003  1F8D 1053
BTR510  003  1F93 1058
BTR520  004  1F9D 1064   1054
BTR600  003  1FA8 1074
BTSSVC  001  1DE9 0510   0497
BTSTOP  001  1DD6 0493
BTS010  003  1DD6 0497
BTS020  004  1DE5 0504
BXCAFC  001  1DD1 0386   0359
BXCAFO  001  1DD2 0387
BXCBN1  002  1DD5 0399
BXCCLC  001  1DD3 0389   0366
BXCLOS  001  1D95 0349
BXCSFA  001  1DD4 0397
BXC010  004  1D95 0353
```

```
BXC020  004  1D9D 0355   0376
BXC120  003  1DA1 0359
BXC130  003  1DB0 0366
BXC140  004  1DBF 0373
BXC150  004  1DCD 0380
BXDBN1  001  13EF 6373   6241
BXDDMY  001  0009 6369
BXDDP0  001  0000 6366   6175  6177
BXDDP1  001  0001 6367   6182
BXDDP2  001  0002 6368   6183
BXDDUM  001  0000 6295   6177  6314  6335  6356
BXDLTH  001  0003 6292   6144  6166  6174  6216  6222  6369
BXDMD1  001  13CB 6301   6144  6222
BXDMD2  001  13D7 6322   6166
BXDMD3  001  13E3 6343   6216
BXDM14  001  13D6 6316   6145* 6223*
BXDPRC  001  13F2 6380   6277
BXDPRO  001  13F3 6381   6182* 6224* 6227*
BXDPRT  001  1300 6131   6144  6145  6166  6216  6222  6223  6304  6308  6312  6325  6329  6333
                         6337  6346  6350  6354  6358
BXDRM1  001  0007 6391   6140  6192
BXDROM  001  0004 6293
BXDRS1  003  13A8 6390   6140* 6192*
BXDSTC  001  13F4 6383   6235
BXDSTO  002  13F6 6384   6231* 6250*
BXDSUB  002  13F1 6374   6250
BXD010  004  1300 6135
BXD020  003  1308 6140
BXD030  003  130B 6144   6193  6202  6350
BXD040  004  1311 6149
BXD050  004  1315 6153
BXD060  004  1319 6157
BXD065  004  131D 6161
BXD070  003  1324 6166
BXD080  004  1327 6171   6162  6218  6242
BXD090  003  132B 6172   6144* 6166* 6216* 6222*
BXD095  003  132E 6174   6178
BXD100  003  1331 6175   6171*
BXD110  004  133D 6182   6176
BXD120  003  1345 6184   6183*
BXD140  003  1348 6188   6304  6308  6325  6329  6346
BXD150  003  134B 6192
BXD160  003  1351 6197   6223
BXD170  004  1354 6201   6358
BXD180  003  135B 6206   6145
BXD190  004  135E 6210   6337
BXD200  003  136A 6216
BXD210  003  1374 6222
BXD220  005  1386 6231
BXD230  003  138B 6235   6251
BXD240  005  1395 6241
BXD250  003  139D 6246
BXD260  004  13A0 6250
BXD270  003  13A7 6255   6256  6258  6333  6390
BXD280  003  13AA 6262   6312  6354
BXD290  004  13AD 6266   6255
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 220
BXD300  003  13B1 6277   6188  6197  6206  6226  6246  6262
BXD310  003  13B8 6282   6237
BXD320  004  13C7 6286   6282*
BXGAFC  001  18EB 8238   8199
BXGAFO  001  18EC 8239
BXGBN1  002  18F1 8251
BXGETX  001  18A3 8187
BXGGTC  001  18ED 8241   8219
BXGGTO  002  18EF 8242
BXGI60  004  18E7 8232
BXGSFA  001  18F0 8250
BXG010  004  18A3 8191
BXG100  003  18AF 8199
BXG110  004  18BE 8206
BXG120  004  18C6 8211   8228
BXG130  004  18CA 8215
BXG140  003  18CE 8219
BXG150  004  18DD 8226
BXIAD2  001  08EE 3422   3395
BXIBLS  002  08F6 3432   3402
BXIBN1  002  08FA 3434   3273  3293  3330  3342  3348
BXIBRC  001  08E8 3412   3277  3366
BXIBRO  002  08EA 3413
BXIBSC  001  0970 3538   3515
BXICA2  002  08EF 3423   3256* 3377  3385  3391* 3401
BXICMK  001  0080 3446   3324  3336  3344
BXIFCP  002  08F2 3425   3383* 3384* 3385
BXIFPE  002  08F4 3426   3383
BXIGTC  001  08EB 3415   3353
BXIGTO  002  08ED 3416
BXIINC  001  096C 3532   3508
BXIINO  001  096D 3533   3469* 3496*
BXILTE  001  0001 3443
BXINPT  001  0800 3255   3454
BXIONE  002  0972 3544   3465  3496  3500
BXIPBA  002  08F8 3433   3377  3391
BXIPSI  001  0004 3440   3424
BXISG2  001  0000 3441   3403
BXISTC  001  08E5 3409   3265
BXISTO  002  08E7 3410
BXISXC  001  096E 3535   3489
BXISXO  001  096F 3536   3485*
BXITB1  001  1B8E 3445   3297  3297*
BXIVTE  001  0000 3442   3324  3336  3344* 3348* 3480  3485
BXI010  004  0803 3260
BXI020  003  080B 3265
BXI030  006  081A 3272
BXI040  003  0825 3277
BXI050  006  0834 3284
BXI060  004  083A 3288
BXI070  006  083E 3292
BXI080  006  0849 3297
BXI090  003  0852 3302
BXI100  004  0855 3306   3362
BXI110  004  0859 3310
BXI120  004  085D 3314
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                          VER 15, MOD 00  20/07/20  PAGE 221
BXI130  003  0861 3315   3302* 3330* 3342*
BXI140  004  0864 3319
BXI145  003  0868 3320   3298  3298* 3349  3349*
BXI150  003  086B 3324
BXI160  004  0871 3330
BXI170  003  087B 3336   3349
BXI180  004  0881 3342
BXI185  003  0888 3344   3298
BXI190  004  088B 3348   3325  3332  3337
BXI210  003  0892 3353
BXI220  004  08A1 3360
BXI230  003  08AB 3366
BXI240  004  08BA 3377
BXI250  004  08C1 3383
BXI260  004  08D1 3391   3378
BXI270  003  08D5 3395
BXI280  003  08DC 3401   3386
BXI290  006  0900 3459
BXI300  006  090A 3464
BXI310  003  0915 3469
BXI320  003  0918 3473
BXI330  004  091B 3477   3504
BXI340  003  091F 3478   3473* 3500*
BXI350  003  0922 3480
BXI360  004  0928 3485
BXI370  003  092C 3489
BXI380  004  093B 3496
BXI390  004  093F 3500
BXI400  003  0943 3504
BXI410  003  0946 3508   3481
BXI420  003  0955 3515
BXI430  004  0964 3522
BXI440  004  0968 3526
BXPAFC  001  1863 7919   7857
BXPAFO  001  1864 7920
BXPBN1  002  1868 7932
BXPC02  001  0002 7936   7878
BXPC04  001  0004 7937   7896
BXPPTC  001  1865 7922   7900
BXPPTO  001  1866 7923   7878* 7896*
BXPSFA  001  1867 7931
BXPUTX  001  1800 7845
BXP010  004  1800 7849
BXP100  003  180C 7857
BXP120  004  181B 7864
BXP140  004  1823 7869   7909
BXP150  004  1827 7873
BXP160  003  182E 7878
BXP170  004  1834 7883   7874
BXP180  004  183B 7888
BXP190  004  183F 7892   7884
BXP200  003  1843 7896
BXP210  003  1846 7900   7879
BXP220  004  1855 7907
BXP230  004  185F 7913
BXRAFC  001  1CE2 9928   9901
```

```
BXRAFO  001  1CE3 9929
BXRBNI  002  1CE6 9941
BXRRTC  001  1CE4 9931   9908
BXRSET  001  1CA6 9891
BXRSFA  001  1CE5 9939
BXR010  004  1CA6 9895
BXR020  004  1CAE 9897   9918
BXR110  003  1CB2 9901
BXR120  003  1CC1 9908
BXR130  004  1CD0 9915
BXR140  004  1CDE 9922
BXUBNC  001  14DF 6685   6518
BXUBNO  002  14E1 6686
BXUBN1  002  14E8 6698   6514  6535  6637
BXUPRC  001  14E2 6688   6667
BXUPRO  001  14E3 6689   6557* 6561* 6592* 6618* 6627* 6647*
BXUPRT  001  1400 6498
BXUSCC  001  14E4 6691   6631
BXUSCO  002  14E6 6692   6623* 6651*
BXUSTC  001  14DC 6682   6507
BXUSTO  002  14DE 6683
BXUSUB  002  14EA 6700   6651
BXU010  004  1400 6502
BXU020  003  1408 6507
BXU025  006  1412 6513
BXU030  003  141D 6518
BXU040  006  1427 6525
BXU050  004  142D 6530
BXU060  006  1431 6534
BXU070  004  143C 6539
BXU080  006  1440 6543
BXU090  004  1446 6548
BXU100  003  144A 6552
BXU110  003  1450 6557
BXU120  003  1453 6561   6604
BXU130  003  1456 6565
BXU140  004  1459 6569
BXU150  003  145D 6573   6603
BXU170  004  1460 6577   6553
BXU180  004  1464 6581
BXU190  004  146B 6586
BXU200  003  146F 6592   6582
BXU210  004  1472 6596
BXU220  004  1479 6601   6619  6638
BXU230  004  1486 6608   6597
BXU240  004  148E 6613
BXU250  003  1495 6618
BXU260  005  149B 6623   6614
BXU270  003  14A0 6627
BXU280  003  14A3 6631   6655
BXU290  005  14AD 6637
BXU300  003  14B5 6642
BXU310  003  14B8 6647
BXU320  004  14BB 6651
BXU340  003  14BF 6655
BXU350  003  14C2 6667   6565  6573  6642
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                          VER 15, MOD 00  20/07/20  PAGE 223
BXU360  003  14C9 6672   6509  6520  6633
BXU370  004  14D8 6676   6672*
CNTAD2  001  0CF5 4593   4523
CNTBLS  002  0CF2 4589   4530
CNTBL1  002  0CFB 4597   4552
CNTBOP  002  0CEB 4578   4553
CNTBRA  001  0CE9 4577   4453
CNTCA2  002  0CF6 4594   4426* 4445* 4480  4504  4507  4522* 4551
CNTCWR  001  0CEE 4582   4496  4497*
CNTENT  001  0000 4585   4531
CNTFCP  002  0CFD 4598   4513* 4514* 4515
CNTFPE  001  001F 4599   4513
CNTPBA  002  0CF4 4590   4507  4522
CNTPSI  001  0004 4584   4586  4587
CNTSAD  001  0CF7 4595   4479* 4550*
CNTSTR  001  0014 4586   4479  4587
CNTTRM  001  0018 4587   4550
CNTUSC  001  0CEC 4580   4539
CNTWRK  002  0CF9 4596   4480* 4515  4529  4551* 4552*
STRAD2  001  0DF5 4757   4677
STRAOP  002  0DDF 4732   4618* 4627  4708* 4715
STRBOP  002  0DF0 4750   4718*
STRCA2  002  0DF6 4758   4675*
STRCOP  002  0DE2 4735   4627*
STRCWR  001  0DE5 4740   4643
STRFN2  001  0DE8 4743   4646
STRFOP  002  0DF3 4753   4715*
STRPBA  002  0DF9 4760   4661  4675
STRSB1  001  0DEE 4749   4719
STRSC1  001  0DEB 4746   4723
STRSTA  001  0DDD 4731   4625  4709
STRSTC  001  0DE0 4734   4628
STRSTF  001  0DF1 4752   4716  4721
STRSTX  001  0DE3 4737   4635
STRUSF  001  0DF4 4755   4713
STRWOP  002  0DE7 4741   4644*
STRXOP  001  0DE4 4738
STR1OP  002  0DED 4747   4707* 4718
TRMAOP  002  0EDF 4884   4789*
TRMBIC  001  0ED8 4878   4773
TRMBN1  002  0EDC 4881   4780  4849
TRMBOP  002  0EE4 4890   4842*
TRMBRC  001  0EE2 4889   4843
TRMFN1  001  0EE5 4892   4829
TRMSTA  001  0EDD 4883   4788
TRMSTX  001  0EE0 4886   4824
TRMUSC  001  0EE8 4895   4831
TWOAD2  001  108D 5282   5239
TWOCA2  002  108E 5283
V$APWR  001  0800 2757   2902
V$BFR1  001  5400 2820   3010
V$BFR2  001  5500 2821   3011
V$CBNZ  001  0CB2 2829   2909
V$CCON  001  5120 2836   3007  4744
V$CDCV  001  3100 2833   2962
V$CDSY  001  2E00 2832   2959
```

```
                                            CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                          VER 15, MOD 00  20/07/20  PAGE 224
V$CFPZ  001  0C70 2827   2908
V$CNXZ  001  0470 2830   2897
V$CSSR  001  5100 2835   3006  4893  5173
V$CZFP  001  04AD 2828   2898
V$DTLN  001  4600 2842   2994
V$DTVR  001  4700 2843   2995
V$FABS  001  1761 2728   2926
V$FACS  001  1400 2744   2918
V$FASN  001  1413 2743   2919
V$FATN  001  1100 2742   2915
V$FCOS  001  0A00 2739   2904
V$FCOT  001  0D00 2737   2910
V$FCSC  001  1725 2741   2925
V$FDEG  001  17DA 2748   2930
V$FDET  001  4540 2751   2993
V$FEXP  001  0500 2735   2899
V$FHCS  001  1500 2747   2920
V$FHSN  001  1557 2746   2921
V$FHTN  001  1593 2745   2922
V$FINT  001  176C 2729   2927
V$FLGT  001  0200 2733   2892
V$FLOG  001  0219 2732   2894
V$FLTW  001  020B 2734   2893
V$FRAD  001  17CB 2749   2929
V$FRND  001  1800 2750   2931
V$FSEC  001  1700 2740   2924
V$FSGN  001  17A7 2730   2928
V$FSIN  001  0A1A 2738   2905
V$FSQR  001  0900 2731   2903
V$FTAN  001  0D28 2736   2911
V$IFCI  001  1B00 2720   2935
V$IFIO  001  1A00 2722   2934
V$ISDN  001  1900 2721   2932
V$KBTL  001  1EAC 2864
V$KBTS  001  0DAC 2863
V$LPRB  001  4F00 2818   3004
V$LPRT  001  4D00 2816   3002
V$LPR2  001  4E00 2817   3003
V$MADD  001  4007 2765   2982  4147
V$MASN  001  43A0 2763   2989  4133
V$MCON  001  4324 2770   2987  4171
V$MIDN  001  4300 2771   2986  4175
V$MINV  001  4500 2775   2992  4159
V$MMPY  001  4100 2767   2983  4155
V$MSMY  001  4264 2768   2985  4130
V$MSUB  001  4000 2766   2981  4151
V$MTRN  001  4400 2774   2991  4163
V$MZER  001  432B 2772   2988  4167
V$PCH1  001  5200 2856   3008
V$PCH2  001  5300 2857   3009
V$SCDI  001  2A00 2813   2953
V$SCDO  001  2A96 2814   2954
V$SFA2  001  5000 2798   3005
V$SFD1  001  0000 2808   2890
V$SFD2  001  0100 2809   2891
V$SKEY  001  2500 2812   2948
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                           VER 15, MOD 00  20/07/20  PAGE 225
V$SPRT  001  2800 2811   2951
V$VMPL  001  4C06 2850   3001
V$VMPS  001  4C00 2849   3000
V$XKAF  001  1C00 2797   2936
V$XKCA  001  2400 2801   2944
V$XKCL  001  240A 2800   2945
V$XKIN  001  2B00 2796   2955
V$XKLP  001  24AD 2802
V$XKRS  001  240D 2799   2946
V$XMGT  001  3E06 2790   2976  8612
V$XMIN  001  3D00 2789   2974  7358
V$XMPL  001  3F06 2793   2979  9169
V$XMPS  001  3F00 2792   2978  9166
V$XMPT  001  3E0C 2791   2977  9569
V$XMPU  001  3F13 2794   2980  0244
V$XMRD  001  3E00 2788   2975  7740
V$XSGT  001  2100 2783   2941  8242
V$XSIN  001  2B6E 2782   2956  3416
V$XSPR  001  3400 2785   2965
V$XSPT  001  1D00 2784   2937
V$XSPU  001  3800 2786   2969
V$XSRD  001  3300 2781   2964  4310
V$00E1  001  0000 2890
V$01E1  001  0100 2891
V$02E1  001  0200 2892
V$02E2  001  020B 2893
V$02F3  001  0219 2894
V$03CC  001  0300 2895
V$04CC  001  0400 2896
V$04E1  001  0470 2897
V$04E2  001  04AD 2898
V$05E1  001  0500 2899
V$06CC  001  0600 2900
V$07CC  001  0700 2901
V$08E1  001  0800 2902
V$09E1  001  0900 2903
V$10E1  001  0A00 2904
V$10E2  001  0A1A 2905
V$11CC  001  0B00 2906
V$12CC  001  0C00 2907
V$12E1  001  0C70 2908
V$12E2  001  0CB2 2909
V$13E1  001  0D00 2910
V$13E2  001  0D28 2911
V$14CC  001  0E00 2912
V$15CC  001  0F00 2913
V$16CC  001  1000 2914
V$17E1  001  1100 2915
V$18CC  001  1200 2916
V$19CC  001  1300 2917
V$20E1  001  1400 2918
V$20E2  001  1413 2919
V$21E1  001  1500 2920
V$21E2  001  1557 2921
V$21E3  001  1593 2922
V$22CC  001  1600 2923
```

```
V$23E1  001  1700 2924
V$23E2  001  1725 2925
V$23E3  001  1761 2926
V$23E4  001  176C 2927
V$23E5  001  17A7 2928
V$23E6  001  17CB 2929
V$23E7  001  17DA 2930
V$24E1  001  1800 2931
V$25E1  001  1900 2932
V$26E1  001  1A00 2934
V$27E1  001  1B00 2935
V$28E1  001  1C00 2936
V$29E1  001  1D00 2937
V$30CC  001  1E00 2938
V$31CC  001  1F00 2939
V$32CC  001  2000 2940
V$33E1  001  2100 2941
V$34CC  001  2200 2942
V$35CC  001  2300 2943
V$36CC  001  2400 2947
V$36E1  001  2400 2944
V$36E2  001  240A 2945
V$36E3  001  240D 2946
V$37E1  001  2500 2948
V$38CC  001  2600 2949
V$39CC  001  2700 2950
V$40E1  001  2800 2951
V$41CC  001  2900 2952
V$42E1  001  2A00 2953
V$42E2  001  2A96 2954
V$43E1  001  2B00 2955
V$43E2  001  2B6E 2956
V$44CC  001  2C00 2957
V$45CC  001  2D00 2958
V$46E1  001  2E00 2959
V$47CC  001  2F00 2960
V$48CC  001  3000 2961
V$49E1  001  3100 2962
V$50CC  001  3200 2963
V$51E1  001  3300 2964
V$52E1  001  3400 2965
V$53CC  001  3500 2966
V$54CC  001  3600 2967
V$55CC  001  3700 2968
V$56E1  001  3800 2969
V$57CC  001  3900 2970
V$58CC  001  3A00 2971
V$59CC  001  3B00 2972
V$60CC  001  3C00 2973
V$61E1  001  3D00 2974
V$62E1  001  3E00 2975
V$62E2  001  3E06 2976
V$62E3  001  3E0C 2977
V$63E1  001  3F00 2978
V$63E2  001  3F06 2979
V$63E3  001  3F13 2980
```

```
SYMBOL  LEN VALUE DEFN   REFERENCES
V$64E1  001  4000 2981
V$64E2  001  4007 2982
V$65E1  001  4100 2983
V$66CC  001  4200 2984
V$66E1  001  4264 2985
V$67E1  001  4300 2986
V$67E2  001  4324 2987
V$67E3  001  432B 2988
V$67E4  001  43A0 2989
V$68E1  001  4400 2991
V$69E1  001  4500 2992
V$69E2  001  4540 2993
V$70E1  001  4600 2994
V$71E1  001  4700 2995
V$72CC  001  4800 2996
V$73CC  001  4900 2997
V$74CC  001  4A00 2998
V$75CC  001  4B00 2999
V$76E1  001  4C00 3000
V$76E2  001  4C06 3001
V$77CC  001  4D00 3002
V$78CC  001  4E00 3003
V$79CC  001  4F00 3004
V$80E1  001  5000 3005
V$81E2  001  5100 3006
V$81E3  001  5120 3007
V$82E1  001  5200 3008
V$83E2  001  5300 3009
V$84E1  001  5400 3010
V$85E2  001  5500 3011
V@CDPT  001  0007 3022
V@CHGH  001  0008 3127
V@CMIC  001  0002 3023
V@CMNI  001  00FF 3020
V@CMUL  001  0007 3128
V@CNIX  001  0080 3021
V@COEX  001  001E 3018
V@CPLS  001  00F0 3025
V@CPRC  001  000A 3027
V@CSQR  001  0003 3125
V@CSTR  001  0002 3126
V@CTTA  001  0027 3028
V@DCAD  001  0002 3048   3049
V@DEXP  001  0000 3053
V@DMAN  001  000D 3055   3056
V@DMN1  001  0001 3054
V@DPDF  001  0002 3043
V@DSAD  001  0001 3044
V@DSGN  001  000D 3056
V@DVAD  001  0004 3049
V@EART  001  0001 3026
V@ECRT  001  0038 3099
V@EFUL  001  00F8 3098
V@EINV  001  00FB 3094
V@EIPR  001  00F5 3095
V@ENSV  001  00F7 3096
```

```
                                 CROSS REFERENCE
SYMBOL  LEN VALUE DEFN   REFERENCES                         VER 15, MOD 00  20/07/20  PAGE 228
V@ENUL  001  0000 3093
V@ERPC  001  0020 3024
V@ESAV  001  00F6 3097
V@FEHN  001  0002 3123
V@FEPL  001  0091 3119
V@FERS  001  0003 3122
V@FPGS  001  0081 3118
V@FRET  001  0015 3121
V@FSPC  001  0040 3120
V@FTAB  001  0000 3124
V@KADD  001  004E 3109
V@KCLE  001  006E 3106
V@KDIV  001  0061 3112
V@KEMN  001  006C 3104
V@KEPL  001  006B 3103
V@KMUL  001  005C 3111
V@KPER  001  004B 3114
V@KPST  001  007B 3108
V@KPWR  001  005A 3113
V@KSQR  001  006F 3105
V@KSTO  001  006D 3107
V@KSUB  001  0060 3110
V@LAIP  001  0003 3074   3075
V@LDEX  001  0002 3077
V@LETE  001  0003 3081
V@LEXP  001  0001 3071   3073
V@LFKO  001  0006 3076
V@LINI  001  0200 3080
V@LLKS  001  0010 3073
V@LMAN  001  000F 3072   3073
V@LNOP  001  0015 3078
V@LTBE  001  0007 3075
V@LVPG  001  0100 3079   3080
V@MCHS  001  00C0 3060
V@MCRD  001  0010 3036
V@MDEF  001  0008 3037
V@MEXC  001  0080 3034
V@MEXT  001  0004 3063
V@MICC  001  0010 3019
V@MIPC  001  0080 3061
V@MIPL  001  0020 3067
V@MLST  001  0040 3035
V@MPND  001  0000 3066
V@MPOF  001  0080 3064
V@MPRC  001  0020 3033
V@MSFU  001  0002 3038
V@MSTN  001  0004 3032
V@OALL  001  00F4 3089
V@ONUL  001  00F0 3085   3086
V@OPM1  001  00F2 3087   3088
V@ORTN  001  00F1 3086   3087
V@OSTK  001  00F3 3088   3089
V@PEOF  001  0002 3062
V@PSQ2  001  0014 3065
```

```
TOTAL STATEMENTS IN ERROR IN THIS ASSEMBLY =     0

OL105 I  THE CODE LENGTH OF #BOVLY IS  8183 DECIMAL.
OL103 I  TOTAL NUMBER OF LIBRARY SECTORS REQUIRED IS   40
         NAME-#BOVLY,PACK-R1R1R1,UNIT-R1,RETAIN-P,LIBRARY-R,CATEGORY-000
```

```
        OVERLAY LINKAGE EDITOR CORE USAGE MAP    20/07/20


 START    CATEGORY  NAME AND       CODE LENGTH
ADDRESS               ENTRY    HEXADECIMAL DECIMAL

 0600         0     #BOVLY          1FF7    8183

OL100 I  THE TOTAL CORE USED BY #BOVLY IS  8183 DECIMAL.
OL101 I  THE START CONTROL ADDRESS OF THIS MODULE IS 0600.
OL104 I  TOTAL NUMBER OF LIBRARY SECTORS REQUIRED IS   32
         NAME-#BOVLY,PACK-R1R1R1,UNIT-R1,RETAIN-P,LIBRARY-O
```