SNAP SYMBOLIC LANGUAGE FOR THE BURROUGHS 220
VERSION 1

BY

MICHAEL J. MAHON

5/10/64

TABLE OF CONTENTS

INFORMATION REGARDING ERRORS, EITHER IN THIS MANUAL OR IN 'SNAP', MAY BE SENT TO THE AUTHOR DIRECTLY OR BY WAY OF THE COMPUTING CENTER.

INTRODUCTION

COMPUTER PROGRAMS MAY, IN GENERAL, BE WRITTEN IN THREE DIFFERENT WAYS.

THE MOST DIRECT METHOD OF PROGRAMMING IS IN BASIC 'MACHINE LANGUAGE,' AND PROGRAMS WRITTEN IN THIS FORM ARE SUITABLE FOR IMMEDIATE EXECUTION BY THE COMPUTER.

IN THE SECOND METHOD OF PROGRAMMING, ONE WRITES IN AN 'ASSEMBLY LANGUAGE' IN WHICH MACHINE OPERATION CODES ARE REPRESENTED BY MNEMONIC NAMES, AND ADDRESSES IN INTERNAL STORAGE ARE REFERRED TO BY PROGRAMMER-ASSIGNED NAMES. ASSEMBLY LANGUAGE 'SOURCE' PROGRAMS ARE NOT SUITABLE FOR DIRECT EXECUTION, BUT MUST BE 'ASSEMBLED' INTO A MACHINE LANGUAGE 'OBJECT' PROGRAM PRIOR TO EXECUTION.

FINALLY, THE PROGRAMMER MAY WRITE IN A 'COMPILER LANGUAGE,' SUCH AS 'ALGOL' OR 'FORTRAN' IN WHICH THE PROGRAM TAKES THE FORM OF A MORE-OR-LESS 'READABLE' PROBLEM-SOLVING ALGORITHM. A COMPILER LANGUAGE PROGRAM MUST BE CONVERTED INTO MACHINE LANGUAGE BEFORE IT CAN BE EXECUTED, AND THIS MAY BE DONE IN EITHER OF TWO WAYS--THE 'COMPILER' MAY TRANSLATE DIRECTLY INTO MACHINE LANGUAGE, OR IT MAY PRODUCE AN ASSEMBLY LANGUAGE PROGRAM WHICH IS THEN ASSEMBLED INTO MACHINE LANGUAGE TEXT BY AN ASSEMBLER. THIS LATTER CASE IS THE MORE USUAL FOR SOPHISTICATED COMPILERS.

COMPILER LANGUAGE, BEING THE MOST PROBLEM-ORIENTED LANGUAGE, IS GENERALLY EASIEST TO LEARN, BUT ORDINARILY HAS MUCH LESS VERSATILITY THAN BASIC MACHINE LANGUAGE. MACHINE LANGUAGE, ON THE OTHER HAND, TENDS TO BE SOMEWHAT TEDIOUS AND ERROR-PRONE BECAUSE OF THE BASIC SIMILARITY OF THE MANY NUMERIC ADDRESSES AND OPERATION CODES WHICH ENTER INTO A PIECE OF CODING. SYMBOLIC ASSEMBLER LANGUAGE PROVIDES FULL ACCESS TO THE VERSATILITY OF MACHINE LANGUAGE, WHILE ALLOWING THE USE OF MEANINGFUL SYMBOLS TO DESIGNATE LOCATIONS IN THE INTERNAL STORAGE OF THE CCMPUTER. A LONG PROGRAM MAY BE WRITTEN IN SECTIONS WITHOUT ANY SACRIFICE OF EFFICIENCY IN THE OBJECT PROGRAM, AND ERRORS OF MISTAKEN ADDRESSES AND OPERATION CODES ARE DRASTICALLY REDUCED. IN ADDITION, PROVISION IS MADE FOR INCLUSION OF EXPLANATORY REMARKS WHICH BECOME AN INTEGRAL PART OF THE SYMBOLIC SOURCE PROGRAM, AND 'PSEUDO-OPERATION' CODES ARE PROVIDED TO HANDLE COMMON PROGRAMMING NEEDS, SUCH AS ARRAY DEFINITION, SUBROUTINE CALLING, AND LISTING CONTROL.

BECAUSE OF THE COMBINATION OF SIMPLICITY AND VERSATILITY OFFERED BY SYMBOLIC ASSEMBLER LANGUAGE, IT IS ALMOST UNIVERSALLY THE STANDARD TOOL OF THE SERIOUS PROGRAMMER.

I. GENERAL DESCRIPTION

'SNAP' ('SYMBOLIC-NUMERIC ASSEMBLY PROGRAM') IS A TWO-PASS SYMBOLIC ASSEMBLER FOR THE BURROUGHS 220 PAPER-TAPE DATA PROCESSING SYSTEM. OPERATION CODES AND ADDRESSES ARE SPECIFIED SYMBOLICALLY, WHILE SIGN AND VARIANT DIGITS (EXCEPT WHERE SPECIFIED BY THE OP-CODE) ARE WRITTEN IN NUMERIC FORM. SOURCE PROGRAMS IN 'SNAP' LANGUAGE ARE PREPARED ON CODING SHEETS, KEYPUNCHED ONTO IBM CARDS, AND FINALLY CONVERTED TO PAPER TAPE. THIS 'SOURCE TAPE' IS THE ORDINARY INPUT TO 'SNAP'.

IN NORMAL OPERATION, THE SOURCE TAPE IS READ IN, ONE 11-WORD RECORD AT A TIME, IN PASS I. SIMULTANEOUSLY, 'SNAP' CONSTRUCTS THE 'SYMBOL TARLE' (OF PROGRANMER-DEFINED LABELS) AND THE 'OPDEF TABLE' (OF PROGRAMMER-DEFINED OP-CODES) AND WRITES THE SOURCE PROGRAM 'IMAGE' ON MAGNETIC TAPE. AT THE END OF PASS I, WHILE THE MAGNETIC TAPE IS REWINDING, THE SYMBOL TABLE FORMED IN PASS I IS PRINTED OUT IN ALPHABETICAL ORDER. DURING PASS II, 'SNAP' READS THE SOURCE PROGRAM FROM MAGNETIC TAPE AND, REFERRING TO THE COMPLETED SYMBOL TABLE, COMPUTES THE VALUES OF THE SYMBOLIC ADDRESS EXPRESSIONS AND GENERATES THE MACHINE LANGUAGE 'TEXT' CORRESPONDING TO THE SOURCE STATEMENTS. THESE INSTRUCTIONS ARE PUNCHED INTO THE PAPER TAPE 'OBJECT PROGRAM' AND THE LISTING OF SOURCE STATEMENTS WITH ASSEMBLED TEXT IS PRODUCED SIMULTANEOUSLY. AT THE CONCLUSION OF PASS II, 'SNAP' HALTS, READY TO RE-ENTER PASS I FOR ANOTHER ASSEMBLY.

THE CBJECT PROGRAM PRODUCED IS SUITABLE EITHER FOR IMMEDIATE EXECUTION OR FOR PHASE 0 INPUT TO THE NUMERIC ASSEMBLER IF THE PROGRAMMER HAS SO WRITTEN IT. OBJECT PROGRAMS MAY BE PREPARED EITHER IN ABSOLUTE MODE ('ABSMOD') OR RELATIVE MODE ('RELMOD') SUITABLE FOR LOADING AS A RELOCATABLE SUBPROGRAM.

A CCMPLETE DIAGNOSTIC ROUTINE IS A STANDARD PART OF PASS I AND PASS II, THOUGH SENSE SWITCH OPTIONS MAY BE USED TO LIMIT ERROR TYPEOUT. PASS II ERRORS ARE 'FLAGGED' BRIEFLY ON THE LISTING, AND ALL ERRORS ARE NORMALLY TYPED OUT ON THE TELETYPE PRINTER, TOGETHER WITH THE STATEMENT WHICH CAUSED THE ERROR.

STANDARD UNIT DESIGNATIONS FOR OUTPUT DEVICES ARE AS FOLLOWS...

```
  SPO = HIGH-SPEED PRINTER
PWR 0 = TELETYPE PRINTER
PWR 1 = HIGH-SPEED PUNCH
```

THE FOLLOWING SENSE SWITCH OPTIONS ARE PROVIDED...

1. SWITCH 5 CAUSES 'SNAP' TO TAKE ITS PASS I INPUT FROM MAGNETIC TAPE,
2. SWITCH 6 ON SUPPRESSES PUNCHING OF AN OBJECT TAPE,
3. SWITCH 7 ON ALLOWS MANUAL DEFINITION OF UNDEFINED SYMROLIC EXPRESSIONS,
4. SWITCH 8 ON SUPPRESSES ERROR TYPEOUT,
5. SWITCH 9 ON SUPPRESSES PRINTING OF THE SYMBOL DICTIONARY,
6. SWITCH 0 ON SUPPRESSES ALL USE OF MAGNETIC TAPE (PAPER-TAPE-ONLY MODE),
7. SWITCH 2 ON (OFF) RESTARTS PASS II (PASS I).

DURING BOTH PASSES, 'SNAP' MAINTAINS A CELL IN STORAGE REFERRED TO AS THE 'LOCATION COUNTER'. THIS CELL CONTAINS THE EXECUTION-TIME STORAGE ADDRESS OF THE INSTRUCTION CURRENTLY BEING ASSEMBLED (RELATIVE TO THE LOAD ADDRESS IF 'RELMOD').

THE APPEARANCE OF A SYMBOL IN THE LABEL FIELD OF ANY MACHINE OPERATION AND MOST PSEUDO-OPERATIONS CAUSES THAT SYMBOL TO BE DEFINED TO HAVE THE CURRENT VALUE OF THE LOCATION COUNTER.

ALTHOUGH THE MOST COMMON '6' OR '7' SIGN DIGIT COMMANDS ARE GENERATED BY THE TWO PSEUDO-OPERATIONS 'LOAD' AND 'TCD', THE PROGRAMMER MAY OCCASIONALLY WISH TO SPECIFY THESE SIGN DIGITS IN OTHER INSTRUCTIONS. (OF COURSE, SUCH AN INSTRUCTION INTERRUPTS NORMAL LOAOING, AND IF FURTHER LOADING IS DESIRED IT MUST BE RESTARTED MANUALLY OR BY PROGRAMMING.) 'SNAP' DOES NOT SENSE THAT A '6' OR '7' SIGN INSTRUCTION HAS BEEN PRODUCED BY A MACHINE OP-CODE STATEMENT, AND SO PROCEEDS TO DEFINE THE LABEL (IF ANY) AND INCREMENT THE LOCATION COUNTER. SINCE SUCH A WORD DOES NOT ENTER STORAGE, THE CORRESPONDENCE BETWEEN LOAD ADDRESS AND LOCATION COUNTER IS DISTURBED, AND MUST BE RESTORED OR TAKEN INTO ACCOUNT IF LOADING IS TO BE RESTARTED.

NUMERIC ASSEMBLER ('ASSEMBLER II') PSEUDO-OPS (SEE APPENDIX A) ARE, IN MOST CASES, HANDLED LIKE MACHINE OPERATIONS, EXCEPT THAT A SYMBOL IN THE LABEL FIELD OF AN 'ASSEMBLER II’ PSEUDO-OP WILL NOT BE DEFINED, AND THE LOCATION COUNTER WILL NOT BE INCREMENTED. FOR THIS REASON, SUCH PSEUDO-OPS SHOULD NOT BE INTERSPERSED WITH MACHINE OPERATIONS.

'SNAP' IS CALLED INTO STORAGE (0000-1799) FROM MAGNETIC TAPE UNDER THE SPAC CONTROL PROGRAM. IF CALL-IN OF 'SNAP' IS TO BE AUTOMATIC, APPROPRIATE MACHINE-CODE WORDS MUST BE PUNCHED ONTO PAPER TAPE BEFORE THE SOURCE PROGRAM. IF MAG-TAPE IS TO BE USED (THE NORMAL SITUATION), TAPE UNIT 1 MUST BE EDITED BEFORE 'SNAP' IS CALLED IN. THE TAPE MUST NOT BE BLOCKED, BUT MUST ALWAYS BE EDITED WHEN 'SNAP' IS LOADED OR RELOADED (NOT BETWEEN 'BATCHED' ASSEMBLIES). WHEN CONTROL IS FIRST PASSED TO 'SNAP', INITIALIZATION IS ACCOMPLISHED AND THE TITLE AND DATE OF CREATION OF THE ASSEMBLER IS PRINTED OUT, FOLLOWED BY THE LIST OF SENSE SWITCH OPTIONS. IF MAGNETIC TAPE UNIT 1 IS INOPERATIVE, OR IT IS NOT DESIREABLE TO USE MAGNETIC TAPE FOR SOME REASON, SWITCH 0 SHOULD BE TURNED ON EVEN BEFORE 'SNAP' IS CALLED INTO STORAGE. IF THIS IS NOT DONE, 'SNAP' WILL WRITE THE OP-CODE TABLE ON MAG-TAPE UNIT 1 FOR RESETTING PURPOSES.

IF IT SHOULD BECOME NECESSARY TO RESTART 'SNAP' MANUALLY, THE FOLLOWING PROCEDURE SHOULD BE USED...

1. TO RESTART PASS I, PUT SWITCH 2 UP, PRESS 'CLEAR’, AND PRESS 'START'.

2. TO RESTART PASS II, PUT SWITCH 2 DOWN, PRESS 'CLEAR', AND PRESS 'START'.

II. BASIC FORMAT

'SNAP' LANGUAGE SYMBOLIC CARDS ARE PUNCHED IN THE FOLLOWING FORMAT--

```
COLUMN   .6        .11       .16     20.      .26                         55.
         LABEL     VARIANT   OP-CODE          ADDRESS   COMMENTS
```

1) 'LABEL' (COL. 6-10) IS A LEFT-ADJUSTED STRING OF FROM ONE TO FIVE NON-BLANK ALPHAMERIC CHARACTERS, AT LEAST ONE OF WHICH IS NON-NUMERIC AND NONE OF WHICH IS AMONG THE FOLLOWING SEVEN....

```
+ (PLUS SIGN)      - (MINUS SIGN)          * (ASTERISK)
/ (SLASH)          $ (DOLLAR SIGN)         , (COMMA)
' (APOSTROPHE)
```

FOR EXAMPLE--

```
ALPHA
BETA
X
K46
3M41
.READ
.NOP.
3.14      (PERMISSIBLE, BUT NOT RECOMMENDED)
```

ARE ALL LEGAL SYMBOLS. ON THE OTHER HAND,

```
A*B
364
$XBAR
X AVG
```

ARE ILLEGAL SYMBOLS.

2) 'VARIANT' (COL. 11-15) IS A FIVE CHARACTER NUMERIC FIELD IN WHICH THE NUMBER IN COL. 11 IS INTERPRETED AS THE SIGN DIGIT OF THE INSTRUCTION AND THE DIGITS FROM COL. 12-15 ARE INTERPRETED AS THE NUMERIC VARIANT OF THE INSTRUCTION. IN ALL CASES BLANKS ARE INTERPRETED AS ZEROS. IF A NON-NUMERIC CHARACTER APPEARS IN THIS FIELD, THE LEAST SIGNIFICANT DIGIT OF THE TWO-DIGIT INTERNAL REPRESENTATION IS USED. NO ERROR IS DETECTED.

EXCEPTIONS--

IF THE OPERATION CODE UNIQUELY SPECIFIES THE LEAST SIGNIFICANT DIGIT (4,l DIGIT) OF THE VARIANT, THE ASSEMBLER IGNORES THE CONTENTS OF COL. 15.

IF A SIGN OF 1 IS DESIRED, IT MAY BE PRODUCED BY A MINUS SIGN (-) IN COL. 11. (ANY OTHER NON-NUMERIC CHARACTER IN THE SIGN COLUMN WILL RESULT IN AN ASSUMED SIGN OF ZERO.)

```
FOR EXAMPLE--   VARIANT FIELD...             ASSEMBLES AS...

                -0410                        1 0410XXAAAA
                -                            1 0000XXAAAA
                3                            3 0000XXAAAA
                 1000                        0 1000XXAAAA
                 4423                        0 4423XXAAAA
                ABCDE                        0 2345XXAAAA
```

(EACH OF THE ABOVE EXAMPLES ASSUMES THAT THE OP CODE DOES NOT SPECIFY THE LEAST SIGNIFICANT VARIANT DIGIT.)

3) 'OP-CODE' (COL. 16-20) IS A LEFT-ADJUSTED OPERATION-CODE MNEMONIC, AS DEFINED IN APPENDIX A, OR TWO LEFT-ADJUSTED DIGITS. OPERATION CODES MAY CONTAIN THE FOUR ARITHMETIC OPERATORS (+-*/) IF SO DEFINED BY THE PROGRAMMER.

```
FOR EXAMPLE--   CAD
                09
                HLT
                37
                LBC
                EJECT
                .ORG.     (DEFINED BY PROGRAMMER)
                STA-V     (DEFINED BY PROGRAMMER)
                IBB/2     (DEFINED BY PROGRAMMER)
                GO TO     (DEFINED BY PROGRAMMER)
```

ARE ALL LEGAL OPERATION CODES. NOTE THAT IN NUMERIC OP-CODES LESS THAN 10, THE LEADING ZERO IS SIGNIFICANT AND MUST BE PRESENT.

4) 'ADDRESS' (COL. 26 TO FIRST BLANK OR COL. 55) IS ANY EXPRESSION FORMED FROM LEGAL SYMBOLS, ACTUAL INTEGER NUMBERS, AND ARITHMETIC OPERATORS. EXPRESSIONS ARE EVALUATED FROM LEFT TO RIGHT USING WHOLE-WORD (10-DIGIT) INTEGER ARITHMETIC. MULTIPLICATIONS AND DIVISIONS ARE DONE FROM LEFT TO RIGHT BEFORE ADDITIONS AND SUBTRACTIONS, NO PARENTHESES ARE ALLOWED. FRACTIONAL RESULTS ARE TRUNCATED TO INTEGERS AND THE FINAL RESULT IS CONVERTED MODULO 10000 AND PLACED IN THE ADDRESS FIELD OF THE INSTRUCTION. THE * (ASTERISK) MEANS BOTH 'MULTIPLY' AND 'PRESENT VALUE OF LOCATION COUNTER' DEPENDING ON CONTEXT. ACTUAL INTEGERS APPEARING IN EXPRESSIONS WILL BE USED MODULO 10000. DIVISION BY ZERO WITHIN A TERM CAUSES THE VALUE OF THAT TERM TO BE SET TO ZERO. OTHER TERMS IN THE SAME EXPRESSION ARE NOT AFFECTED.

(IN SOME PSEUDO-OPERATIONS THE ADDRESS FIELD IS USED DIFFERENTLY. REFER TO INDIVIDUAL PSEUDO-OP DESCRIPTIONS.)

FOR EXAMPLE-- (ASSUME... ALPHA=2046, BETA=5, LOCATION COUNTER=25)

THE EXPRESSION...	HAS THE VALUE...
ALPHA	2046
37	0037
-1	9999
-ALPHA	7954
*+2	0027
ALPHA-BETA	2041
BETA*8-30	0010
2*ALPHA-2*BETA	4082
ALPHA/BETA	0409
*+ALPHA/BETA	0434
***/2	0312
**	0000
512000/1000	0012 (NOT RECOMMENDED)
51200*10/1000	0512

ARE ALL LEGAL ADDRESS FIELD EXPRESSIONS,

5) 'COMMENTS' (FIRST BLANK AFTER COL. 26 TO COL. 55) IS ANY REMARK THE PROGRAMMER WISHES TO INSERT. AT LEAST ONE BLANK MUST OCCUR BETWEEN THE START OF THE VARIABLE FIELD (COL. 26) AND THE FIRST CHARACTER OF THE REMARK. REMARKS SHOULD NOT USE SPECIAL CHARACTERS OR MULTIPLE PUNCHES WHICH WILL INTERFERE WITH LISTING, NAMELY-- DOLLAR SIGN ($), COMMA (,) AND APOSTROPHE (*).

LIBERAL USE OF MEANINGFUL COMMENTS IS STRONGLY RECOMMENDED. IF MODIFICATIONS TO A PREVIOUSLY WRITTEN PROGRAM BECOME NECESSARY, COMMENTS ARE INVALUABLE.

CARD CCLUMNS 1-5, 21-25, AND 55-80 ARE COMPLETELY IGNORED BY THE ASSEMBLER. CCLUMNS 55-80 ARE A SUITABLE PLACE FOR DECK IDENTIFICATION AND SEQUENCING IF THE PROGRAMMER DESIRES.

III. PSEUDO-OPERATIONS

A. ASSEMBLER CONTROL PSEUDO-OPERATIONS

1) 'ABS' AND 'REL'

THE PSEUDO-OPERATIONS 'ABS' AND 'REL' SPECIFY WHETHER THE SOURCE PROGRAM IS TO BE ASSEMBLED IN ABSOLUTE MODE ('ABSMOD') OR RELATIVE MODE ('RELMOD'). THE OBJECT PROGRAM PRODUCED BY AN 'ABSMOD' ASSEMBLY IS LOADED INTO THE ABSOLUTE ORIGIN(S) SPECIFIED IN THE SOURCE PROGRAM. THE OBJECT PROGRAM PRODUCED BY A 'RELMOD' ASSEMBLY MAY BE 'FLOATED' INTO CORE STORAGE WITH B-REGISTER ADDRESS MODIFICATION, IN WHICH CASE THE ADDRESSES SPECIFIED IN THE SOURCE PROGRAM ARE RELATIVE TO THE LOADING ORIGIN (MODULO 10000). THE OPERATIONS 'ABS' AND 'REL' CONSIST OF...

1. BLANKS IN THE LABEL AND VARIANT FIELDS,
2. 'ABS' OR 'REL' IN THE OP-CODE FIELD,
3. BLANKS OR COMMENTS IN THE VARIABLE FIELD.

'ABS' AND 'REL' MAY OCCUR ANY NUMBER OF TIMES IN THE SOURCE PROGRAM, THE MODE OF ASSEMBLY BEING DETERMINED BY THE LAST PRECEDING MODE CONTROL CARD. IF NO MODE CONTROL CARD APPEARS, 'ABSMOD' IS ASSUMED.

IT SHOULD BE NOTED THAT SOME MAGNETIC TAPE COMMANDS AND SEVERAL PSEUDO-OPS ARE NOT ALLOWED IN A 'RELMOD' ASSEMBLY. A LIST OF THESE OPERATION CODES ARE PROVIDED BELOW.

```
MFC            MFS            SPAC           AUX
BRO            BRU            BRZ            CHA
DEL            INS            MSR            PSR
REF            SEC            ZZZ
```

2) 'END'

THE 'END' PSEUDO-OPERATION INDICATES TO THE ASSEMBLER THE PHYSICAL END OF THE SOURCE PROGRAM. THE 'END' OPERATION CONSISTS OF...

1. BLANKS IN THE LABEL AND VARIANT FIELDS,
2. 'END' IN THE OP-CODE FIELD,
3. ANY ALPHAMERIC COMMENT IN THE VARIABLE FIELD. (NO BLANK IS NECESSARY PRECEDING COMMENTS.)

WHEN THE ASSEMBLER ENCOUNTERS THE 'END' CARD IMAGE IN PASS I, CONTROL IS TRANSFERRED TO THE INTERLUDE PASS AND THEN TO PASS II. WHEN PASS II ENCOUNTERS THE 'END' STATEMENT, THE SYMBOL TABLE IS SCANNED FOR UNREFERENCED SYMBOLS, AND THE ASSEMBLY IS TERMINATED. IF 'SNAP' IS IN THE PAPER-TAPE-ONLY MODE (SWITCH 0 ON) IN PASS I, THE 'END' STATEMENT WILL CAUSE THE COMPUTER TO HALT WITH P-T (0766 00 4740) IN THE C-REGISTER (TO REMIND THE OPERATCR TO REWIND THE SOURCE TAPE) BEFORE PROCEEDING TO PASS II.

3) 'PAUSE'

THE 'PAUSE' PSEUDO-OPERATION IS PROVIDED FOR CONVENIENCE IN CHANGING PAPER TAPE REELS IN A MULTI-REEL SOURCE PROGRAM. THE 'PAUSE' OPERATION CONSISTS OF...

1. BLANKS IN THE LABEL AND VARIANT FIELDS,
2. 'PAUSE' IN THE OP-CODE FIELD,
3. BLANKS IN THE VARIABLE FIELD.

WHEN A 'PAUSE' STATEMENT IS ENCOUNTERED DURING PASS I, THE ASSEMBLER HALTS WITH T-T (0474 00 4740) IN THE C-REGISTER. ASSEMBLY PROCEEDS NORMALLY UPON PRESSING 'START'. IF 'SNAP' IS IN THE PAPER-TAPE-ONLY MODE (SWITCH 0 ON), 'PAUSE' WILL CAUSE A SIMILAR HALT IN PASS II. THE 'PAUSE' STATEMENT DOES NOT APPEAR IN THE LISTING.

4) 'DUP'

THE 'DUP' PSEUDO-OPERATION IS NOT OPERATIVE INDEPENDENT OF THE MACRO-EXPANDER. AT PRESENT THE ONLY EFFECT OF THE 'DUP' IS TO DEFINE THE SYMBOL IN THE LABEL FIELD TO HAVE THE PRESENT VALUE OF THE LOCATION COUNTER.

5) 'TCD'

IN ACCORDANCE WITH COMMON USAGE ON CARD-ORIENTED MACHINES, THE PSEUDO-OPERATION 'TCD' (TRANSFER CARD) GENERATES A BRANCH-TO-PROGRAM INSTRUCTION. THE OPERATION 'TCD' CONSISTS OF...

1. BLANKS IN THE LABEL AND VARIANT FIELDS,
2. 'TCD' IN THE OP-CODE FIELD,
3. AN EXPRESSION REFERRING TO THE DESIRED ENTRY POINT IN THE PROGRAM.

THE CODING...

```
          6 0000 30 AAAA
```

IS GENERATED IN THE OBJECT PROGRAM, WHICH WILL TERMINATE LOADING AND TRANSFER CONTROL TO AAAA, THE VALUE OF THE ADDRESS FIELD EXPRESSION.

B. LOCATION COUNTER CONTROL PSEUDO-OPERATIONS

1) 'LOAD'

THE 'LOAD' PSEUDO-OPERATION ORIGINS THE LOCATION COUNTER AND GENERATES A '6 PRB' LOADING INSTRUCTION INTO THAT ORIGIN. THE 'LOAD' OPERATION CONSISTS OF...

1. A SYMBOL OR BLANKS IN THE LABEL FIELD,
2. BLANKS IN THE VARIANT FIELD,
3. 'LOAD' IN THE OP-CODE FIELD,
4. AN EXPRESSION (CONTAINING SYMBOLS DEFINED PHYSICALLY EARLIER IN THE SOURCE PROGRAM) IN THE ADDRESS FIELD.

THE 'LOAD' OPERATION GENERATES THE INSTRUCTION...

```
6 0000 04 AAAA
```

IN AN 'ABSMOD' ASSEMBLY, OR...

```
7 0008 O4 AAAA
```

IN A 'RELMOD' ASSEMBLY, WHERE AAAA IS THE VALUE OF THE ADDRESS FIELD EXPRESSION AND THE NEW VALUE OF THE LOCATION COUNTER. THE SYMBOL IN THE LABEL FIELD, IF ANY, IS DEFINED TO HAVE THE VALUE AAAA. 'LOAD' IS TYPICALLY THE FIRST STATEMENT OF ANY PRCGRAM.

2) 'ORG'

THE 'ORG' PSEUDO-OPERATION 'ORIGINS' THE VALUE OF THE LOCATION COUNTER WITHOUT GENERATING THE 'LOAD' (6 000O 04 AAAA) INSTRUCTION. THE 'ORG' OPERATION CONSISTS OF...

1. A SYMBOL OR BLANKS IN THE LABEL FIELD,
2. BLANKS IN THE VARIANT FIELD,
3. 'ORG' IN THE OP-CODE FIELD,
4. AN EXPRESSION (CONTAINING SYMBOLS DEFINED PHYSICALLY EARLIER IN THE SOURCE PROGRAM) IN THE ADDRESS FIELD.

THE LOCATION COUNTER IS SET TO THE VALUE OF THE ADDRESS FIELD EXPRESSION, AND THE LABEL, IF ANY, IS DEFINED TO HAVE THAT VALUE.

'ORG' IS ESPECIALLY USEFUL FOR SETTING UP REMOTE STORAGE BLOCKS AND FOR CODING INSTRUCTIONS WHICH ARE TO BE MOVED IN STORAGE PRIOR TO THEIR EXECUTION. FOR EXAMPLE, SUPPOSE THREE ARRAYS ARE TO BE DEFINED AT A REMOTE LOCATION IN STORAGE. THIS MAY PE ACCOMPLISHED EVEN IN THE MIDDLE OF A STRING OF IN-LINE CODING BY THE FOLLOWING SEQUENCE OF INSTRUCTIONS...

```
      .SAVE     SET       *       SAVE PRESENT VALUE
                ORG       REMOT   BASE ADDRESS OF ARRAYS
      A         BSS       10*10   DEFINE
      AUG       BSS       10*20     THE
      AINV      BSS       10*10       ARRAYS...
      REMOT     SET       *       KEEP RECORD OF STORAGE
                LOAD      .SAVE   RESTORE LOAD ADDRESS.
```

SUPPOSE A PROGRAM IS DESIRED WHICH WILL DUMP A SELECTED AREA OF STORAGE, GENERATE A CHECK SUM, AND PUNCH A SUM CHECKING ROUTINE ONTO THE TAPE TO VERIFY CORRECT LOADING OF THE PROGRAM WHEN THE DUMP IS LOADED. SAY THE SUM CHECKING ROUTINE WILL BE LOADED INTO 4985 WHEN IT IS EXECUTED, BUT WILL RESIDE AT SOME OTHER ADDRESS IN THE DUMPING PROGRAM. THIS MAY BE CODED AS FOLLOWS...

```
      .SAVE     SET       *       SAVE LOCATION COUNTER
      .BGIN     ORG       4985    SET ORIGIN
      * SUM CHECKING LOOP TO BE PUNCHED OUT WITH DUMP
      CKSUM     CSU       .SUM.   A-REG = -.SUM.
      LENGT     SOR       **      FILLED IN BY DUMP PROG
            0401DLB       *-1     B = LENGTH
      HIADR-    ADD       **      FILLED IN BY DUMP PROG
               1DBB       *-1     GO TIL FINISHED...
                BOF       *+1     RESET OVFLOW
                SOH               SET OVERFLOW HALT
      START     BFA       **      FILLED IN BY DUMP PROG
            9669HLT       9669    NO GOOD.  STOP
                BUN       *-1     DEAD.....
      .SUM.     DEC               FILLED IN BY DUMP PROG
      * END OF SUM CHECKING ROUTINE...
      .LGTH     SET       *-.BGIN (LENGTH OF ROUTINE)
                ORG       .SAVE+.LGTH (RESTORE COUNTER)
```

3) 'BSS'

THE 'BSS' ('BLOCK STARTED BY SYMBOL') PSEUDO-OPERATION DEFINES A BLOCK OF STORAGE. THE NAME OF THE BLOCK REFERS TO ITS FIRST LOCATION. THE 'BSS' OPERATION CONSISTS OF...

1. A SYMBOL OR BLANKS IN THE LABEL FIELD,
2. BLANKS IN THE VARIANT FIELD,
3. 'BSS' IN THE OP-CODE FIELD,
4. AN EXPRESSION (CONTAINING SYMBOLS DEFINED PHYSICALLY EARLIER IN THE SOURCE PROGRAM) IN THE ADDRESS FIELD.

THE LENGTH OF THE BLOCK DEFINED BY THE 'BSS' IS EQUAL TO THE VALUE OF THE ADDRESS FIELD EXPRESSION. THE SYMBOL IN THE LABEL FIELD, IF ANY, IS ASSIGNED THE PRESENT VALUE OF THE LOCATION COUNTER, AND THE LOCATION COUNTER IS INCREMENTED BY THE LENGTH OF THE BLOCK. A 'LOAD' INTO THE INCREMENTED VALUE OF THE LOCATION COUNTER IS THEN SIMULATED TO RESERVE THE BLOCK DURING LOADING. THE 'BSS' HAS THE SAME EFFECT AS THE SEQUENCE...

```
LABEL     EQU       *
          LOAD      *+LENGT
```

THE 'BSS' IS MOST USEFUL IN DEFINING ARRAYS TO BE REFERENCED BY B-REGISTER ADDRESS MODIFICATION WITHIN 'DBB' LOOPS, SINCE THE VALUE OF B-REGISTER MODIFIED ADDRESSES WITHIN SUCH LOOPS GOES FROM ACTUAL VALUE PLUS SOME POSITIVE MODIFICATION TO THE ACTUAL VALUE PLUS ZERO. THUS THE ARRAY IS REFERENCED SERIALLY FROM ITS HIGHEST LOCATION (=LABEL +LENGTH-1) TO ITS LOWEST LOCATION (=LABEL).

4) 'BES'

THE 'BES' ('BLOCK ENDED BY SYMBOL') PSEUDO-OPERATION DEFINES A BLOCK OF STORAGE. THE NAME OF THE BLOCK REFERS TO THE LAST LOCATION OF THE BLOCK PLUS ONE. THE 'BES' OPERATION CONSISTS OF...

1. A SYMBOL OR BLANKS IN THE LABEL FIELD,
2. BLANKS IN THE VARIANT FIELD,
3. 'BES' IN THE OP-CODE FIELD,
4. AN EXPRESSION (CONTAINING SYMBOLS DEFINED PHYSICALLY EARLIER IN THE SOURCE PROGRAM) IN THE ADDRESS FIELD.

THE OPERATION OF THE 'BES' PSEUDO-OP IS IDENTICAL WITH THAT OF THE 'BSS' DESCRIBED ABOVE, EXCEPT THAT THE SYMBOL IN THE LABEL FIELD IS ASSIGNED THE VALUE OF THE INCREMENTED LOCATION COUNTER. THE 'BES' HAS THE SAME EFFECT AS THE SEQUENCE...

```
      LABEL     LOAD      *+LENGT
OR...
                BSS       LENGT
      LABEL     EQU       *
```

THE 'BES' IS MOST USEFUL IN DEFINING ARRAYS TO BE REFERENCED BY B-REGISTER ADDRESS MODIFICATION WITHIN 'IBB' LOOPS, SINCE THE VALUE OF B-REGISTER MODIFIED ADDRESSES WITHIN SUCH LOOPS GOES FROM ACTUAL VALUE PLUS SOME NEGATIVE (MODULO 10000) MODIFICATION TO ACTUAL VALUE PLUS 9999 (MINUS 1). THUS THE ARRAY IS REFERENCED SERIALLY FROM ITS LOWEST LOCATION (=LABEL-LENGTH) TO ITS HIGHEST LOCATION (=LABEL-1).

5) 'SEX'

THE 'SEX' PSEUDO-OPERATION IS INCORPORATED FOR USE WITH THE NUMERIC ASSEMBLER. IT GENERATES A SECTION DEFINING WORD AND ORIGINS THE LOCATION COUNTER AT AN EVEN HUNDRED LOCATION. THE 'SEX' OPERATION CONSISTS OF...

1. BLANKS IN THE LABEL AND VARIANT FIELDS,
2. 'SEX' IN THE OP-CODE FIELD,
3. AN EXPRESSION (CONTAINING SYMBOLS DEFINED PHYSICALLY EARLIER IN THE SOURCE PROGRAM) IN THE ADDRESS FIELD.

THE LOCATION COUNTER IS ORIGINED TO 100 TIMES THE VALUE OF THE ADDRESS FIELD EXPRESSION, MODULO 10000, AND THE WORD...

```
9 9999 99 9AAA
```

IS GENERATED, WHERE AAA IS THE VALUE OF THE ADDRESS FIELD EXPRESSION TRUNCATED TO THREE DIGITS.

C. SYMBOL DEFINING PSEUDO-OPERATIONS

1) 'EQU'

THE 'EQU' PSEUDO-OPERATION DEFINES THE SYMBOL IN THE LABEL FIELD TO BE EQUAL IN VALUE TO THE ADDRESS FIELD EXPRESSION. THE 'EQU' OPERATION CONSISTS OF...

1. A SYMBOL IN THE LABEL FIELD,
2. BLANKS IN THE VARIANT FIELD,
3. 'EQU' IN THE OP-CODE FIELD,
4. AN EXPRESSION (CONTAINING SYMBOLS DEFINED PHYSICALLY EARLIER IN THE SOURCE PROGRAM) IN THE ADDRESS FIELD.

A SYMBOL DEFINED BY AN 'EQU' IS ASSUMED TO HAVE A UNIQUE VALUE, AND ANY ATTEMPT AT REDEFINITION WILL BE DIAGNOSED AS AN ERROR.

2) 'SET'

THE 'SET' PSEUDO-OPERATION DEFINES A 'SET-SYMBOL' TO HAVE THE VALUE OF THE ADDRESS FIELD EXPRESSION. THE 'SET' OPERATION CONSISTS OF...

1. A SYMBOL IN THE LABEL FIELD,
2. BLANKS IN THE VARIANT FIELD,
3. 'SET' IN THE OP-CODE FIELD,
4. AN EXPRESSION (CONTAINING SYMBOLS DEFINED PHYSICALLY EARLIER IN THE SOURCE PROGRAM) IN THE ADDRESS FIELD.

A SYMBOL ORIGINALLY DEFINED BY A 'SET' OPERATION IS REFERRED TO AS A 'SET-SYMBOL' AND HAS THE SPECIAL PROPERTY THAT IT MAY BE REDEFINED AS OFTEN AS DESIRED BY SUBSEQUENT 'SET' OPERATIONS. THE 'SET' OPERATION MAY NOT BE USED TO REDEFINE ANY SYMBOL WHICH IS ALREADY DEFINED AS AN ORDINARY SYMBOL, NOR MAY ANY OTHER OPERATION BE USED TO REDEFINE A SET-SYMBOL. ANY ATTEMPTED VIOLATION OF THESE RULES WILL BE DIAGNOSED AS AN ERROR AND IGNORED. ALL 'SET' STATEMENTS ARE EVALUATED IN BOTH PASSES OF THE ASSEMBLER, AND MAY THEREFORE BE USED FOR SETTING ASSEMBLY PARAMETERS, KEEPING TALLIES, SAVING VALUES OF THE LOCATION COUNTER, ETC. SET-SYMBOLS ARE NOT FLAGGED AS MULTIPLY-DEFINED IN THE SYMBOL DICTIONARY. A SET-SYMBOL IS LISTED IN THE UNREFERENCED SYMBOL LIST IF IT IS NOT REFERENCED AFTER ITS FINAL DEFINITION.

C. OPERATION CODE DEFINING PSEUDO-OPERATIONS

1) 'OPSYN'

THE 'OPSYN' PSEUDO-OPERATION PROVIDES A MEANS OF RENAMING PREVIOUSLY DEFINED OP-CODES. THE 'OPSYN' OPERATION CONSISTS OF...

1. AN OP-CODE MNEMONIC IN THE LABEL FIELD,
2. BLANKS IN THE VARIANT FIELD,
3. 'OPSYN' IN THE OP-CODE FIELD,
4. A PREVIOUSLY DEFINED OP-CODE MNEMONIC IN THE ADDRESS FIELD. (COMMENTS MUST NOT OCCUR BEFORE COLUMN 31.)

THE MNEMONIC APPEARING IN THE LABEL FIELD (WHICH MAY ALSO BE THE BLANK MNEMONIC, ORDINARILY SYNONYMOUS WITH 'HLT') IS DEFINED TO BE SYNONYMOUS WITH THE MNEMONIC APPEARING IN THE ADDRESS FIELD. A MNEMONIC MAY CONSIST OF FROM ONE TO FIVE ALPHAMERIC CHARACTERS, EXCLUDING ONLY THE DOLLAR SIGN ($), APOSTROPHE ('), AND COMMA (,).

IT SHOULD BE OBSERVED THAT ONCE A 'STANDARD' MNEMONIC HAS BEEN REDEFINED, THE OPERATION WHICH IT ORIGINALLY REPRESENTED CANNOT BE REFERENCED BY THE REMAINDER OF THE SOURCE PROGRAM UNLESS IT MAS BEEN 'SAVED' UNDER SOME OTHER NAME. FOR EXAMPLE, SUPPOSE 'ORG' IS TO BE DEFINED TO BE SYNONYMOUS WITH 'LOAD'. THIS, IN EFFECT, DESTROYS THE 'ORG' OPERATION UNLESS 'ORG' IS SAVED...

```
.ORG.     OPSYN     ORG
ORG       OPSYN     LOAD
```

SUBSEQUENT TO THESE STATEMENTS, THE OPERATION OF 'ORG-ING' CAN BE OBTAINED UNDER THE NEW NAME '.ORG.'

2) 'OPDEF'

THE 'OPDEF' PSEUDO-OPERATION PROVIDES A MEANS OF DEFINING NEW OP-CODES FOR USE WITHIN AN ASSEMBLY. THE 'OPDEF' OPERATION CONSISTS OF...

1. AN OP-CODE MNEMONIC IN THE LABEL FIELD,
2. BLANKS IN THE VARIANT FIELD,
3. 'OPDEF' IN THE OP-CODE FIELD,
4. A NUMBER OF UP TO TEN DIGITS IN THE ADDRESS FIELD.

THE 'OPDEF' OPERATION DEFINES THE MNEMONIC (INCLUDING THE BLANK MNEMONIC) IN THE LABEL FIELD TO BE THE OPERATION SPECIFIED IN THE ADDRESS FIELD. FOR THE FORMAT OF THIS SPECIFICATION, SEE APPENDIX A.

NOTE...

'OPDEF' AND 'OPSYN' BOTH USE THE OPDEF TABLE FOR ENTRY OF NEW OP-CODE MNEMONICS. SINCE THE CAPACITY OF THIS TABLE IS 100 UNIQUE MNEMONICS, THE MAXIMUM NUMBER OF DIFFERENT MNEMONICS ADDED IS LIMITED TO THIS FIGURE. THE OPDEF TABLE IS RESET ONLY WHEN 'SNAP' IS RELOADED, BUT NOT BETWEEN PROGRAMS IN A BATCH ASSEMBLY. THIS ALLOWS A STANDARD 'PACKAGE' OF OP-CODES TO BE ADDED WHICH WILL APPLY TO ALL THE PROGRAMS OF THE BATCH. (THE 'STANDARD' OP-CODES ARE RESET BETWEEN PROGRAMS AND BETWEEN PASSES UNLESS 'SNAP' IS IN THE PAPER-TAPE-ONLY MODE.)

3) 'MACRO'

THE 'MACRO' PSEUDO-OPERATION IS NOT OPERATIVE INDEPENDENT OF THE MACRO-EXPANDER.

E. DATA DEFINING PSEUDO-OPERATIONS

1) 'ALPH'

THE 'ALPH' PSEUDO-OPERATION GENERATES FROM ONE TO SIX ALPHAMERIC WORDS. THE 'ALPH' OPERATION CONSISTS OF...

1. A SYMBOL OR BLANKS IN THE LABEL FIELD,
2. A SINGLE DIGIT BETWEEN 1 AND 6 OR A BLANK IN THE LEAST SIGNIFICANT POSITION OF THE VARIANT FIELD,
3. 'ALPH' IN THE OP-CODE FIELD,
4. ANY ALPHAMERIC CHARACTERS IN THE VARIABLE FIELD (INCLUDING CARRIAGE CONTROL CHARACTERS). COMMENTS NEED NOT BE PRECEDED BY A BLANK.

IF THE VARIANT FIELD OF AN 'ALPH' STATEMENT IS LEFT BLANK, A LENGTH OF ONE WILL BE ASSUMED. IF A LENGTH GREATER THAN SIX WORDS IS SPECIFIED, SIX WILL BE ASSUMED AND AN ERROR DETECTED. ANY CHARACTER IS LEGAL IN THE VARIABLE FIELD OF AN 'ALPH' EXCEPT DOLLAR SIGN ($), APOSTROPHE ('), AND COMMA (,). THESE CHARACTERS MAY BE GENERATED BY A 'DEC' WITH A SIGN OF 2 IF NEEDED.)

TO CLARIFY THE LISTING, ONLY THE FIRST WORD GENERATED BY EACH 'ALPH' STATEMENT IS SHOWN IN THE 'INSTRUCTION' COLUMN, AND CARRIAGE CONTROL CHARACTERS ARE TRANSLATED INTO ORDINARY PRINTABLE CHARACTERS FOR VERIFICATION OF THE CONTENTS OF THE VARIABLE FIELD. THE TRANSLATION IS AS FOLLOWS...

CHARACTER...	MULTIPLE PUNCH...	LISTED AS...
CARRIAGE RETURN	11-8-6	PHI
TABULATE	0-8-6	THETA
EJECT CODE	11-8-5	COMMA

2) 'DEC'

THE 'DEC' PSEUDO-OPERATION DEFINES A SINGLE FIXED OR FLOATING POINT DECIMAL CONSTANT HAVING ANY SPECIFIED SIGN. THE 'DEC' OPERATION CONSISTS OF...

1. A SYMBOL OR BLANKS IN THE LABEL FIELD,
2. A SIGN OR BLANKS IN THE VARIANT FIELD,
3. 'DEC' IN THE OP-CODE FIELD,
4. A FIXED OR FLOATING POINT NUMBER (SEE BELOW) IN THE ADDRESS FIELD.

THE NUMBER IN THE ADDRESS FIELD HUST HAVE ITS LITERAL VALUE AND MAY NOT BE AN EXPRESSION. IF THE NUMBER CONTAINS A DECIMAL POINT IT IS CCNSIDERED TO BE A FLOATING POINT NUMBER, OTHERWISE IT IS ASSUMED TO BE FIXED POINT. THE FLOATING POINT CONVERSION ROUTINE STRIPS THE EIGHT MOST SIGNIFICANT DIGITS (PRECEDING THE FIRST BLANK) FROM THE ADDRESS FIELD, IGNORING NON-NUMERIC CHARACTERS. THE EXPONENT IS THEN CALCULATED FRCM THE RELATIVE POSITIONS OF THE DECIMAL POINT AND THE FIRST SIGNIFICANT DIGIT. THE FIXED POINT CONVERSION ROUTINE STRIPS THE TEN MOST SIGNIFICANT DIGITS AND JUSTIFIES THE NUMBER SO THAT THE LEAST SIGNIFICANT DIGIT STRIPPED BECOMES THE (0,1) DIGIT OF THE GENERATED WORD. A PRECEDING PLUS OR MINUS SIGN IS INTERPRETED BY EITHER CONVERTER AS A SIGN OF ZERO OR ONE, RESPECTIVELY. IF A SIGN DIGIT OTHER THAN ZERO OR ONE IS DESIRED, IT MAY BE SPECIFIED IN THE VARIANT FIELD SIGN POSITION.

NOTE...

A FLOATING POINT 'ZERO' DOES NOT RESULT IN THE GENERATION OF A TRUE ZERO WORD, SINCE A NON-ZERO EXPONENT IS GENERATED.

SOME EXAMPLES OF THE USE GF 'DEC' ARE GIVEN BELOW...

```
     THE STATEMENT...                  ASSEMBLES AS...

     DEC       3.14159                 0 5131415900
     DEC       -3.14159                1 5131415900
 3   DEC       3.14159                 3 5131415900
 -   DEC       50                      1 0000000050
     DEC       -3331                   1 0000003331
     DEC                               0 0000000000
     DEC       -                       1 0000000000
 5   DEC                               5 0000000000
     DEC       0.0                     0 5200000000
     DEC       0000000.                0 5800000000
     DEC       .000057689              0 4657689000
 2   DEC       0202020216              2 0202020216 (ILLEGAL)
```

WARNING-- IT IS POSSIBLE TO GENERATE WORDS WITH A '2' IN THE SIGN POSITION AND ARBITRARY DIGITS. WHEN ALPHAMERIC TRANSLATION OF SUCH A WORD IS ATTEMPTED DURING PUNCHING, IT MAY RESULT IN THE PRODUCTION OF ILLEGAL ALPHAMERIC CHARACTERS OR IN NON-PUNCHABLE CHARACTERS (SUCH AS IGNORE CODES (02)) WHICH WILL CAUSE INCORRECT READ-IN OF THE WORDS INVOLVED DURING OBJECT PROGRAM LOADING.

3) 'ZZZ'

THE 'ZZZ' PSEUDO-OPERATION GENERATES THE 'ZERO CREATOR' WORD IN NUMERIC ASSEMBLER LANGUAGE. THE 'ZZZ' OPERATION CONSISTS OF...

1. BLANKS IN THE LABEL FIELD,
2. THE LENGTH OF THE NULL SECTION RIGHT-ADJUSTED IN THE VARIANT FIELD,
3. 'ZZZ' IN THE OP-CODE FIELD,
4. AN EXPRESSION (CONTAINING SYMBOLS DEFINED PHYSICALLY EARLIER IN THE SOURCE PROGRAM) IN THE ADDRESS FIELD.

THE ADDRESS FIELD EXPRESSION IS EVALUATED MODULO 10000 AND THE LEAST SIGNIFICANT TWO DIGITS (AA) ARE PLACED IN THE (0,2) FIELD OF THE GENERATED WORD...

```
9 LLLL 90 90AA
```

F. SUBROUTINE RELATED PSEUDO-OPERATIONS

1) 'CALL'

THE 'CALL' PSEUDO-OPERATION IS A SINGLE-STATEMENT GENERATOR OF THE STANDARD TWO-INSTRUCTION LINKAGE TO A SPAC SUBROUTINE. THE 'CALL' OPERATION CONSISTS OF...

1. A SYMBOL OR BLANKS IN THE LABEL FIELD,
2. A SIGN AND VARIANT OR BLANKS IN THE VARIANT FIELD,
3. 'CALL' IN THE OP-CODE FIELD,
4. AN EXPRESSION IN THE ADDRESS FIELD.

THE 'CALL' STATEMENT...

```
LABEL-1234CALL      SUBR
```

IS EXACTLY EQUIVALENT TO THE 'ABSMOD' SEQUENCE...

```
LABEL-1234STP       SUBR
     -I234BUN       SUBR+1
```

WHICH IS THE STANDARD LINKAGE FOR MOST SPAC SUBROUTINES. AS THE EXAMPLE IMPLIES, THE SIGNS AND VARIANTS OF THE GENERATED INSTRUCTIONS ARE IDENTICAL AND SPECIFIED BY THE VARIANT FIELD OF THE 'CALL'. THE SYMBOL IN THE LABEL FIELD, IF ANY, REFERS TO THE FIRST ('STP') INSTRUCTION OF THE CALLING SEQUENCE. IF DIFFERENT VARIANTS ARE REQUIRED IN THE 'STP' AND 'BUN' INSTRUCTIONS, 'CALL' CANNOT BE USED, BUT THE INDIVIDUAL INSTRUCTIONS MUST BE WRITTEN OUT. OTHER CALLING PARAMETERS REQUIRED BY THE SUBROUTINE MAY BE WRITTEN AS 'DEC' OR 'HLT' STATEMENTS AFTER THE 'CALL' STATEMENT.

THE 'CALL' PSEUDO-OPERATION MAY BE USED IN A 'RELMOD' PROGRAM, BUT IT WILL REFER TO ABSOLUTE ADDRESSES. THIS IS CONSISTENT WITH THE NECESSITY OF ASSIGNING ABSOLUTE LOCATIONS TO SPAC SUBROUTINES. IF IT IS NECESSARY TO LINK TO A SUBROUTINE WRITTEN INTO THE 'RELMOD' PROGRAM, 'CALL' SHOULD NOT BE USED, BUT INSTEAD THE 'STP'-'BUN' SEQUENCE SHOULD BE WRITTEN OUT.

NOTE...

WHEN THE 'SELF' SYMBOL, ASTERISK (*), IS USED IN THE ADDRESS FIELD OF A 'CALL' STATEMENT, IT REFERS TO THE FIRST INSTRUCTION ('STP') GENERATED BY THE 'CALL'.

2) 'SPAC'

THE 'SPAC' PSEUDO-OPERATION CAUSES A SPAC SUBROUTINE TO BE CALLED INTO STORAGE DURING LOADING OF THE OBJECT TAPE, AND ALSO ASSIGNS AN INTERNAL NAME TO THE SUBROUTINE SO THAT IT MAY BE REFERENCED SYMBOLICALLY. THE 'SPAC' OPERATION CONSISTS OF...

1. A SYMBOL OR BLANKS IN THE LABEL FIELD,
2. THE NUMBER OF LOCATIONS TO BE RESERVED FOR THE SUBROUTINE RIGHT-ADJUSTED IN THE VARIANT FIELD,
3. 'SPAC' IN THE OP-CODE FIELD,
4. AN EXPRESSION (CONTAINING SYMBOLS DEFINED PHYSICALLY EARLIER IN THE SOURCE PROGRAM) IN THE ADDRESS FIELD.

THE 'SPAC' OPERATION ENTERS THE SPAC SUBROUTINE NUMBER (EQUAL TO THE VALUE OF THE ADDRESS FIELD EXPRESSION MODULO 100) IN THE INTERNAL SPAC TABLE, ALONG WITH THE PRESENT VALUE OF THE LOCATION COUNTER. THESE TABLE ENTRIES ARE PUNCHED OUT AT THE BEGINNING OF THE OBJECT TAPE FOLLOWING THE SPAC CALL-IN SEQUENCE DURING THE INTERLUDE PASS. THE SYMBOL IN THE LABEL FIELD, IF ANY, IS DEFINED TO HAVE THE PRESENT VALUE OF THE LOCATION COUNTER, AND THE LOCATION COUNTER IS THEN INCREMENTED BY THE LENGTH SPECIFIED IN THE VARIANT FIELD. FINALLY, A 'LOAD' INSTRUCTION INTO THE ADDRESS SPECIFIED BY THE INCREMENTED LOCATION COUNTER IS GENERATED TO RESERVE THE SPACE SPECIFIED.

THE 'SPAC' PSEUDO-OP IS NOT ALLOWED IN A 'RELMOD' ASSEMBLY. THAT IS, A RELOCATABLE SUBROUTINE IS NOT ALLOWED TO CALL-IN RELOCATABLE SUBROUTINES. OF COURSE, A DUMMY 'ABSMOD' SECTION MAY PRECEDE A 'RELMOD' PROGRAM IF IT IS NECESSARY TO GENERATE AN ABSOLUTE SPAC CALL-IN SEQUENCE.

NOTE...

SINCE THE CALL-IN SEQUENCE PUTS THE SPAC CONTROL PROGRAM INTO LOCATIONS 0000 TO 0400, SUBROUTINES MAY NOT BE CALLED INTO THIS REGION. A GOOD PROCEDURE IN SHORT PROGRAMS IS TO USE THE 'SPAC' OPERATION UNDER A REMOTE LOCATION COUNTER ORIGIN. FOR EXAMPLE...

```
.SAVE     SET       *       SAVE LOCATION COUNTER
          ORG       2000    REMOTE ORIGIN..
SQRT    25SPAC      1       SQRT = SPAC 01
PLOT   230SPAC      28      PLOT = SPAC 28
          LOAD      .SAVE   RESTORE LOAD ADDRESS.
```

THE SPAC SUBROUTINE CALL-IN GENERATED BY 'SNAP' DOES NOT REQUIRE ANY CONTROL SWITCH SETTINGS.

G. LISTING CONTROL PSEUDO-OPERATIONS

1) 'REM'

THIS OPERATION IS USED TO INSERT REMARKS IN THE LISTING. THE 'REM' PSEUDO-OPERATION CONSISTS OF...

1. ANY ALPHAMERIC CHARACTERS IN THE LABEL FIELD AND THE VARIANT FIELD,
2. 'REM' IN THE OP-CODE FIELD,
3. ANY ALPHAMERIC CHARACTERS IN COL. 21-25 AND THE VARIABLE FIELD.

THE CARD IMAGE, WITH THE EXCEPTION OF THE OP-CODE FIELD, WILL BE LISTED.

NOTE...

REMARKS MAY ALSO BE INSERTED BY PUTTING AN ASTERISK (*) IN COLUMN 6. THE REST OF THE CARD (COL. 7-55) IS THEN AVAILABLE FOR COMMENTS.

2) 'SPACE'

THE 'SPACE' PSEUDO-OPERATION CAUSES BLANK LINES TO APPEAR IN THE LISTING. THE 'SPACE' OPERATION CONSISTS OF...

1. BLANKS IN THE LABEL AND VARIANT FIELDS,
2. 'SPACE' IN THE OP-CODE FIELD,
3. AN EXPRESSION HAVING VALUE LESS THAN 50 IN THE ADDRESS FIELD.

THE NUMBER OF BLANK LINES INSERTED IS EQUAL TO THE VALUE OF THE EXPRESSION, WITH THE FOLLOWING EXCEPTIONS...

1) IF THE NUMBER OF BLANK LINES SPECIFIED IS GREATER THAN THE NUMBER OF LINES REMAINING CN THE PAGE, 'EJECT' IS SIMULATED.
2) IF THE VALUE OF THE ADDRESS FIELD EXPRESSION IS ZERO, ONE BLANK LINE IS INSERTED.

3) 'EJECT'

THE 'EJECT' PSEUDO-OPERATION CAUSES THE PRESENT PAGE OF LISTING TO BE FILLED OUT WITH BLANK LINES SO THAT THE NEXT STATEMENT APPEARS AT THE TOP OF THE NEXT PAGE. THE 'EJECT' OPERATION CONSISTS OF...

1. BLANKS IN THE LABEL AND VARIANT FIELDS,
2. 'EJECT' IN THE OP-CODE FIELD,
3. BLANKS IN THE VARIABLE FIELD.

THE 'EJECT' STATEMENT DOES NOT APPEAR ON THE LISTING.

IV. DIAGNOSTIC MESSAGES

GENERAL--

WHEN AN ERROR IN THE SOURCE PROGRAM IS DETECTED DURING ASSEMBLY, THE TYPE OF ERROR WILL BE PRINTED IN CODED FORM ON THE TELETYPE PRINTER (E.G. 'ER S1') FOLLOWED BY THE CARD IMAGE CONTAINING THE ERROR. IN PASS II THE LISTING IS ALSO FLAGGED WITH THE ERROR CODE, IMMEDIATELY TO THE LEFT OF THE LABEL FIELD. ERROR PRINTING ON THE TELETYPE PRINTER MAY BE SUPPRESSED BY TURNING SWITCH 8 ON. (THIS IS NOT RECOMMENDED UNLESS A CATASTROPHIC SNOWBALLING OF ERROR TYPEOUT OCCURS.)

A. SYMBOL RELATED ERRORS

1) S1 --UNDEFINED SYMBOL

THIS ERROR IS DETECTED WHENEVER A REFERENCE IS MADE TO A SYMBOL WHICH IS NOT DEFINED AT THE TIME OF THE REFERENCE.

DURING PASS I IT IS CAUSED BY PSEUDO-OPS WHICH MUST CALCULATE THE VALUE OF AN EXPRESSION CONTAINING SYMBOLS WHICH HAVE NOT AS YET ENTERED THE SYMBOL TABLE. SUCH PSEUDO-DPS ARE...

```
BES                 BSS                 EQU
LOAD                OPDEF               ORG
SET                 SEX                 SPAC
```

SYMBOLS USED IN THE ADDRESS FIELD EXPRESSIONS OF THESE PSEUDO-OPS MUST HAVE BEEN DEFINED PHYSICALLY EARLIER IN THE SOURCE PROGRAM. AN S1 ERROR IN PASS I CAUSES PASS II TO BE ABANDONED (UNLESS A VALUE FOR THE EXPRESSION IS MANUALLY ASSIGNED AS DESCRIBED BELOW).

ALL SYMBOL TABLE ENTRIES HAVE BEEN MADE BEFORE THE START OF PASS II, THEREFORE, AN S1 ERROR IN THE SECOND PASS INDICATES THAT A REFERENCE HAS BEEN MADE TO A SYMBOL NOT DEFINED ANYWHERE IN THE SOURCE PROGRAM.

THE VALUE OF AN EXPRESSION IN WHICH AN UNDEFINED SYMBOL APPEARS WILL NORMALLY BE SET EQUAL TO ZERO. HOWEVER, A SENSE SWITCH OPTION (SWITCH 7) IS PROVIDED WHICH WILL CAUSE THE COMPUTER TO COME TO A HALT WITH THE C-REGISTER = 7777 00 7777 WHEN AN S1 ERROR OCCURS. AT THIS TIME THE OPERATOR MAY CLEAR THE ACCUMULATOR (WHICH CONTAINS THE NUMERIC REPRESENTATION OF THE UNDEFINED SYMBOL) AND INSERT IN THE 0,4 FIELD THE PROPER VALUE FOR THE ENTIRE EXPRESSION, AS DETERMINED FROM THE DICTIONARY LISTING AND THE PRESENT VALUE OF THE LOCATION COUNTER.

2) S2 --MULTIPLY-DEFINED SYMBOL

IF THE PROGPAMMER ATTEMPTS TO DEFINE THE SAME SYMBOL MORE THAN ONCE IN ANY OPERATION OTHER THAN 'SET', ERROR CODE S2 IS PRINTED. THE SYMBOL RETAINS ITS ORIGINAL VALUE BUT IS FLAGGED WITH '+M+' IN THE SYMBOL DICTIONARY. ONLY THE 'SET' PSEUDO-OPERATION IS ABLE TO ASSIGN A NEW VALUE TO A DEFINED SYMBOL, AND THEN ONLY IF THE SYMBOL WAS ORIGINALLY DEFINED BY A 'SET' OPERATION.

3) S3 --SYMBOL TABLE FULL

THE CAPACITY OF THE 'SNAP' SYMBOL TABLE IS 1535 SYMBOLS. IF THE PROGRAMMER ATTEMPTS TO USE MORE SYMBOLS, 'ER S3' WILL BE PRINTED, ALONG WITH THE CARD IMAGE CONTAINING THE 1536TH LABEL, AND THE ASSEMBLY WILL BE TERMINATED IMMEDIATELY. THE ONLY REMEDY IS TO RE-WRITE THE PROGRAM USING FEWER SYMBOLS.

4) S4 --SYMBOL MISSING FROM LABEL FIELD OF 'EQU'

SINCE THE 'EQU' PSEUDO-OP EXISTS FOR THE PURPOSE OF DEFINING SYMBOLS, AN 'EQU' WITH A BLANK LABEL FIELD IS DIAGNOSED AS AN ERRDR. SUCH A STATEMENT WILL BE TREATED AS A REMARK.

5) S5 --NON-'SET' SYMBOL IN LABEL FIELD OF 'SET'

IF AN ATTEMPT IS MADE TO REDEFINE A SYMBOL WHICH FIRST APPEARED IN THE LABEL FIELD OF ANY LABEL-DEFINING OPERATION EXCEPT 'SET' THE USE OF A 'SET' OPERATION, ERROR S5 WILL BE DETECTED AND THE VALUE OF THE SYMBOL IS LEFT UNCHANGED. THE SYMBOL DICTIONARY ENTRY IS NOT FLAGGED. A BLANK LABEL FIELD IN A 'SET' STATEMENT WILL ALSO RESULT IN ERROR S5.

B. OPERATION CODE RELATED ERRORS

1) O1 --UNDEFINED OPERATION CODE

IN EITHER PASS, AN UNDEFINED OPERATION CODE RESULTS IN THE ASSUMPTION OF A 'HLT' COMMAND. OPERATION CODES DEFINED BY 'OPDEF' OR 'OPSYN' (OR BY 'MACRO' DEFINITIONS) MUST BE DEFINED PHYSICALLY PRIOR TO THEIR APPEARANCE IN THE OP-CODE FIELD. A FREQUENT CAUSE OF O1 ERRORS IS MISSPELLING OF OP-CODES.

2) O2 --OPDEF TABLE FLLL

THE PSEUDO-OPERATIONS 'OPDEF', 'OPSYN', AND 'MACRO' WILL INSERT A NEW ENTRY INTO THE OPDEF TABLE IF THE MNEMONIC BEING DEFINED IS NOT ALREADY DEFINED TO BE AN OP-CODE MNEMONIC. SINCE THE MAXIMUM CAPACITY OF THE OPDEF TABLE IN 'SNAP' IS 100 ENTRIES, ANY ATTEMPT TO ENTER A 101ST OP-CODE NAME WILL RESULT IN AN O2 ERROR. THIS ERROR WILL NOT TERMINATE ASSEMBLY, BUT IT MAY, OF COURSE, RESULT IN SUBSEQUENT O1 ERRORS.

3) O3 --OPERATION CODE ILLEGAL IN 'RELMOD' ASSEMBLY

THE STATEMENT CAUSING THE ERROR IS ASSEMBLED AS IF IT WERE IN AN 'ABSMOD' ASSEHBLY. ASSEMBLY IS NCT TERMINATED.

CERTAIN OPERATION CODES MAY NOT BE USED AT ALL IN 'RELMOD' ASSEMBLIES. NUMERIC ASSEMBLER PSEUDO-OPS ('AUX','PSR','SEX', ETC.) ARE EXAMPLES OF THIS TYPE.

TWO OTHER OPERATION CODES, SPECIFYING MACHINE OPERATIONS, ARE NOT USEABLE IN THE NORMAL WAY IN 'RELMOD' ASSEMBLIES. THESE CODES ARE 'MFC' AND 'MFS', WHICH REQUIRE THE 4-BIT OF THE SIGN TO BE ON AS A PART OF THE OP-CODE. SINCE 'FLOATING ADDRESS' LOADING CAN CNLY MODIFY THE ADDRESSES OF INSTRUCTIONS WITH 8-BITS ON, AND SUBSEQUENT TO MODIFICATION REDUCES THE SIGN TO 0 OR 1, THE INCOMPATIBILITY OF 'MFC' AND 'MFS' IS CLEARLY SEEN. THE PROGRAMMER MUST OBTAIN THE RESULT HE SEEKS BY PROGRAM MODIFICATION AT EXECUTION TIME. OF COURSE THERE IS NO SUCH INCOMPATIBILITY IN AN 'ABSMOD' ASSEMBLY.

4) O4 --'SPAC' USED IN 'RELMOD' ASSEMBLY

THIS ERROR CAN ONLY OCCUR IN PASS I. SPAC SUBROUTINES CANNOT BE CALLED INTO MEMORY BY A RELOCATABLE PROGRAM. THE 'SPAC' STATEMENT CAUSING THE ERROR WILL BE TREATED AS A REMARK AND PASS II WILL BE ABANDONED. THE ERROR SCAN IN PASS I WILL CONTINUE. IF NECESSARY, A DUMMY 'ABSMOD' SECTION MAY BE INTRODUCED AHEAD OF A 'RELMOD' PROGRAM FOR THE PURPOSE OF GENERATING AN (ABSOLUTE) SPAC CALL-IN SEQUENCE.

C. MISCELLANEOUS ERRORS

1) V1 --ILLEGAL SIGN OR VARIANT

THIS ERROR CAN OCCUR ONLY IN PASS II. IT IS DETECTED WHEN THE ADDITION OF THE SIGN AND THE RETAINED PORTION OF THE VARIANT (AS SPECIFIED BY THE OP-CODE) TO THE OP-CODE CAUSES OVERFLOW. FOR EXAMPLE, SUPPOSE 'STO' HAS BEEN DEFINED TO BE 'STORE THE OP-CODE PART OF THE A-REGISTER'...

```
          STO       OPDEF     0621040100
```

NOW SUPPOSE IT IS USED IN THE FOLLOWING WAY...

```
     96210STO       ALPHA+3
```

WHEN THE SIGN-VARIANT COMBINATION IS NOW ADDED TO THE OP-CODE PROTOTYPE, OVERFLOW RESULTS...

```
    0 0621040100 + 0 9621000000 = 0 0242040100   (OVERFLOW)
```

THE INSTRUCTION WOULD THEN BE FLAGGED 'V1' AND ASSEMBLED AS...

```
         0 242040AAAA   (WHERE AAAA = ALPHA+3)
```

IT SHOULD BE NOTICED THAT THE RESULTANT SIGN AFTER SUCH AN ERROR CAN BE 2. WHEN SUCH AN 'INSTRUCTION' IS PUNCHED OUT, ALPHAMERIC TRANSLATION WILL BE ATTEMPTED, RESULTING IN SOME UNIQUE AND PROBABLY ILLEGAL ALPHAMERIC CHARACTERS.

2) V2 --ILLEGAL SIGN

THIS ERROR, LIKE V1, OCCURS ONLY DURING PASS II, WHEN OVERFLOW IS SENSED AS A RESULT OF ADDING IN THE SIGN OF 8 ASSOCIATED WITH 'RELMOD' ASSEMBLY. OBVIOUSLY, AN INSTRUCTION TO BE ASSEMBLED IN RELOCATABLE FORM CANNOT HAVE A SOURCE LANGUAGE SIGN OTHER THAN 0 OR 1.

3) V3 --'ALPH' LENGTH GREATER THAN SIX SPECIFIED

A SINGLE 'ALPH' CAN DEFINE FROM ONE TO SIX ALPHAMERIC WORDS. IF A VARIANT DIGIT (LENGTH) GREATER THAN 6 IS GIVEN, 6 WILL BE ASSUMED AND ERROR V3 DETECTED. THIS ERROR WILL BE DETECTED IN BOTH PASS I AND PASS II. IT IS A WARNING ONLY AND DOES NOT TERMINATE ASSEMBLY.

4) T1 --INSUFFICIENT MAGNETIC TAPE

THIS ERROR CAN CCCUR CNLY DURING PASS I WHEN MAGNETIC TAPE IS BEING WRITTEN. ONLY AN EXTREMELY LONG PROGRAM (GREATER THAN 5000 CARDS) CAN EVER CAUSE THIS ERROR UNDER NORMAL CIRCUMSTANCES. SINCE THE AMOUNT OF MAG-TAPE ON THE REEL IS USUALLY BEYOND THE CONTROL OF THE PROGRAMMER, THE ONLY SOLUTION IS TO RE-ASSEMBLE USING THE PAPER-TAPE-ONLY OPTION (SWITCH 0). (BEAR IN MIND THAT IN THIS MODE THE OPERATION CODE TABLE IS NOT RESET, EITHER BETWEEN PASSES OR BETWEEN BATCH ASSEMBLIES.)

5) (NO CODE) --OVERFLOW HALT

A COMPUTER OVERFLOW HALT CAN CCCUR IF THE LOCATION COUNTER IS INCREMENTED PAST 9999. ASSEMBLY MAY BE CONTINUED (IF SUCH A CONDITION IS ALLOWABLE) BY RESETTING THE OVERFLOW INDICATOR AND PRESSING START. THE LOCATION COUNTER WILL CONTINUE TO COUNT, MODULO 10000.

6) L1 --SPAC SUBROUTINE CALLED INTO LOCATION BELOW 0400

SINCE THE SPAC CONTROL PROGRAM OCCUPIES 0000-0399 AT LOAD TIME, SPAC SUBROUTINES MAY NOT OCCUPY THIS REGICN. PASS I WILL CONTINUE, BUT PASS II WILL BE ABANDONED. SEE SECTION III, F, PART 2, FOR A SUGGESTION AND EXAMPLE REGARDING SPAC SUBROUTINES IN SHORT PROGRAMS.

V. PROGRAM EXAMPLE

AN EXAMPLE OF A 'SNAP' LANGUAGE PROGRAM IS GIVEN BELOW. THE LISTING IS IN EXACTLY THE SAME FORM AS THE LISTING PRODUCED BY 'SNAP'. THE EXAMPLE IS A RELOCATABLE SUBROUTINE SUITABLE FOR WRITING TEN-WORD RECORDS ON MAGNETIC-TAPE. (THIS SUBROUTINE IS OF PARTICULAR INTEREST SINCE IT IS VERY SIMILAR TO THE 'WRITE' SUBROUTINE IN 'SNAP' ITSELF.)

NOTE THAT THE VALUE OF THE LOCATION COUNTER IS PRINTED IN THE 'ADRES' COLUMN AND THE ASSEMBLED TEXT IN THE 'INSTRUCTION' COLUMN. THE SPACES BETWEEN THE 'INSTRUCTION' COLUMN AND THE 'LABEL' COLUMN ARE USED FOR ERROR FLAGS. THE COMPLETE CARD IMAGE (COL. 6-55) APPEARS TO THE RIGHT OF THE ERROR FLAG FIELD.

```
   S N A P   VER. 1E    5/10/64



   SWITCH OPTIONS MUST BE SET DURING O-O HALT.

   SW 5 ON IF INPUT IS ON MAG-TAPE
   SW 6 ON SUPPRESSES PUNCHING
   SW 7 ON TO INSERT VALUE IN AC FOR UNDEFINED ADRES
   SW 8 ON SUPPRESSES ERROR TYPING
   SW 9 ON SUPPRESSES DICTIONARY PRINTING
   SW 0 ON INDICATES PAPER-TAPE-ONLY MODE




                                   DICTIONARY

                LABEL VALUE    LABEL VALUE    LABEL VALUE    LABEL VALUE

                .WRIT 0009     BTOGO 0017     BTOTL 0022     CLOSE 0015
                EOT   0030     EOTMS 0033     BUF   0036     OPEN  0003
                WRITE 0000     WRTRC 0026
```

```
ADRES INSTRUCTION    LABEL VAR  OP       VARIABLE FIELD

                     *   WRITE --MAGNETIC TAPE WRITE SUBROUTINE
                     *           WRITES 10 WRD RECORDS IN 100 WRD BLKS
                     *           BLOCKS TAPE AS NEEDED
                     *           (TAPE MUST BE EDITED UPON LOADING.)
                     *
                     *  ENTRY AT WRITE+1 FOR NORMAL WRITE.
                     *  ENTRY AT WRITE+2 TO CLOSE OUT THE FILE.
                     *  INITIAL ENTRY AT WRITE+3 TO OPEN FILE.
                     *
                     *  ENTER WITH ADDRESS OF FIRST WORD OF RECORD IN
                     *  B-REGISTER.  UPON EXIT NEXT AVAILABLE ADDRESS
                     *  IN BUFFER IS LEFT IN B-REG.



0000  0 0000000000             REL                     SET RELMOD
0000  7 0008040000             LOAD      0
0000  8 0009300000   WRITE    9BUN       **            RETURN
0001  8 0000300009             BUN       .WRIT         WRITE A RECORD
0002  8 0000300015             BUN       CLOSE         CLOSE FILE...
0003  8 1008500036   OPEN  1000MRW       BUF           PREPARE TAPE 1
0004  8 0000420022             LDB       BTOTL         SET BTOGO
0005  8 0412400017         0410STB       BTOGO         = BTOTL
0006  8 0000100000             CAD       WRITE         GET =9
0007  8 4410400014         4410STA       CLOSE-1       SET TALY=9...
0008  8 0000300027             BUN       WRTRC+1       AND GO ON....
0009  8 0412400011   .WRIT 0410STB       *+2           SET RECRD ADDR.
0010  8 0000420003             LDB       OPEN          LOAD BUF ADDR..
0011  8 0100290000          100RTF       **            MOVE 10 WORDS
0012  8 0412400003         0410STB       OPEN          HOLD BUF ADDR..
0013  8 4101270014         4101DFL       *+1           BUMP TALLY.
0014  8 0009320000            9BRP       WRITE         IS IT TENTH....
0015  8 0401270017   CLOSE 0401DFL       BTOGO         ANY BLOCKS LEFT
0016  8 0000320026             BRP       WRTRC         YES. WRITE...
0000  0 0000000000             ABS           (WANT ABSOLUTE ADDRESS)
0017  0 1002580000   BTOGO 1000MPE                     NO. SET AND
0000  0 0000000000             REL
0018  8 1000540036         1000MIW       BUF           WRITE 10 BLOCKS
0019  8 1000590021         1000MIB       *+2           DELAY UNTIL
0020  8 0000300019             BUN       *-1           TAPE STOPS...
0021  8 1001590030         1000MIE       EOT           END OF TAPE ERR
0000  0 0000000000             ABS           (WANT ABSOLUTE ADDRESS)
0022  8 1001580000   BTOTL 1000MPB       0             BACKSPACE 10...
0000  0 0000000000             REL
0023  8 0410260017         0410IFL       BTOGO         BUMP BTOGO 10..
0024  8 0000310025             BOF       *+1           RESET OVFLOW...
0025  8 0410260022         0410IFL       BTOTL         BUMP BTOTL 10.
```

```
   ADRES INSTRUCTION    LABEL VAR  OP       VARIABLE FIELD

   0026  8 1100560036   WRTRC 1100MOW       BUF           WRITE BUFFER.
   0027  8 0000420026             LDB       WRTRC         RESET BUFFER
   0028  8 0412400003         0410STB       OPEN             ADDRESS....
   0029  8 0000300000             BUN       WRITE         AND GO HOME..
   0030  8 0030090033   EOT     30SPO       EOTMS         WRITE MESSAGE
   0000  0 0000000000             ABS           (WANT HALT ADDR ABSOLUTE)
   0031  0 9669009669         9669HLT       9669          AND HALT....
   0000  0 0000000000             REL
   0032  8 0000300000             BUN       WRITE         RETURN....
   0033  2 1645554400   EOTMS    3ALPH      $END OF TAPE      (ER MESS)
   0036  7 0008040136   BUF       BSS       100           OUTPUT BUFFER
   0000  0 0000000000             ABS           (WANT TO PUT 6 IN SIGN)
   0136  6 1111001111        61111HLT       1111          HALT LOADING...
                                  END
```

SINCE THE COMMENTS MAKE THE PROGRAM MORE OR LESS SELF EXPLANATORY, WE WILL LOOK INTO THE DETAILS OF ASSEMBLER CONTROL WITHIN THE PROGRAM. IN THE FIRST PLACE, NOTE THAT THE ASSEMBLY IS 'RELMOD', SINCE A RELOCATABLE OBJECT PROGRAM IS DESIRED. THIS MEANS THAT 'SNAP', AFTER FINDING A PARTICULAR OP-CODE PROTOTYPE IN THE OP-TABLE AND ASCERTAINING THAT IT IS INDEED LEGAL IN 'RELMOD', WILL PROCEED TO ADD ON THE SIGN AND VARIANT SPECIFIED IN THE SOURCE PROGRAM. THEN, IF THE ADDRESS OF THE INSTRUCTION IS NOT SPECIFIED BY THE PROTOTYPE TO BE UNCONDITIONALLY ABSOLUTE (AS IN A SHIFT COMMAND, FOR EXAMPLE), 'SNAP' WILL ADD 8 TO THE SIGN DIGIT. THUS IT IS NECESSARY TO MOMENTARILY ENTER 'ABSMOD' FOUR TIMES DURING ASSEMBLY, THREE TIMES TO PRODUCE ABSOLUTE ADDRESSES AND ONCE TO ALLOW SPECIFICATION OF A LITERAL SIGN NOT EQUAL TO 8 OR 9. (OF COURSE, AN ALTERNATE METHOD WHICH THE PROGRAMMER MAY FIND USEFUL IN SOME CASES IS TO REDEFINE, SAY, THE 'HLT' OPERATION TO HAVE AN ABSOLUTE ADDRESS, AND DEFINING ANOTHER OP-CODE TO MEAN 'ZERO OP-CODE WITH RELATIVE ADDRESS'.)

NOTE ALSO THE USE OF ** (LITERALLY 'SELF' TIMES ZERO) IN THE CONVENTIONAL WAY TO INDICATE AN ADDRESS TO BE FILLED IN AT EXECUTION TIME. OF COURSE A ZERO ADDRESS MAY BE OBTAINED IN MANY WAYS, THE MOST COMMON BEING AN EXPLICIT ZERO IN THE ADDRESS FIELD OR SIMPLY A 'NULL' (BLANK) ADDRESS FIELD, OBTAINED BY LEAVING (AT LEAST) COLUMN 26 BLANK. EXAMPLES OF BOTH METHODS ARE SHOWN.

THE INITIAL CHARACTER IN THE ADDRESS FIELD OF THE 'ALPH' STATEMENT AT LOCATION 0033 WAS AN 11-8-6 PUNCH (CARRIAGE RETURN) WHICH WAS TRANSLATED INTO 'PHI' (REPRESENTED BY A $, SINCE 'PHI' CANNOT BE LISTED IN THIS MANUAL).

APPENDIX A

NOTE ON FORMAT OF ENTRIES IN OP-TABLE

THERE ARE TWO FORMATS, THE MACHINE-OPERATION FORMAT (INDICATED BY A SIGN OF 0) AND THE PSEUDO-OPERATION FORMAT (INDICATED BY A SIGN OF 1).

MACHINE-OPERATION FORMAT...

```
                    0 SVVVVXXABC

        S = SIGN OF UNMODIFIED INSTRUCTION
     VVVV = VARIANT OF UNMODIFIED INSTRUCTION
       XX = NUMERIC OP-CODE OF INSTRUCTION
        A = '(4,1) DIGIT FIXED' FLAG
            A=1 IMPLIES (4,1) DIGIT MAY NOT BE ALTERED
            A=0 IMPLIES (4,1) DIGIT MAY BE ALTERED
        B = 'ILLEGAL IN RELMOD' FLAG
            B=1 IMPLIES ILLEGAL IN 'RELMOD'
            B=0 IMPLIES O.K. IN 'RELMOD'
        C = 'ABSOLUTE ADDRESS' FLAG
            C=1 IMPLIES ADDRESS IS ABSOLUTE EVEN IN 'RELMOD'
            C=0 IMPLIES ADDRESS IS RELATIVE IN 'RELMOD'
```

PSEUDO-OPERATION FORMAT...

```
                    1 00AA00BBBB

     BBBB = PASS II ENTRY POINT OF PSEUDO-OP SUBROUTINE
       AA = PASS I ENTRY POINT (RELATIVE TO BBBB) OF PSEUDO-OP SUBROUTINE
```

MNEMONIC	OPERATION	TYPE	ENTRY
(BLANK)	HALT	M	0 0000000000
ABS	SET 'ABSMOD'	P	1 0002000655
ADA	ADD ABSOLUTE	M	0 0000112100
ADD	ADD	M	0 0000012100
ADL	ADD TO LOCATION	M	0 0000019000
ALPH	ALPHAMERIC INFORMATION	P	1 0024000659
AUX	AUXILIARY STORAGE	N	0 9000081010
BCE	BRANCH, COMPARE EQUAL	M	0 0000035100
BCH	BRANCH, COMPARE HIGH	M	0 0000034100
BCL	BRANCH, COMPARE LOW	M	0 0000134100
BCS	BRANCH, CONTROL SWITCH	M	0 0000038000
BCU	BRANCH, COMPARE UNEQUAL	M	0 0000135100
BES	BLOCK ENDED BY SYMBOL	P	1 0004000703
BFA	BRANCH, FIELD A	M	0 0000036000
BFR	BRANCH, FIELD R	M	0 0000037000
BOF	BRANCH, OVERFLOW	M	0 0000031000
BRO	BRUTE-FORCE BY OPERATOR	N	0 9000096010

BRP	BRANCH, REPEAT	M	0	0000032000
BRU	BRUTE-FORCE IN ASSEMBLY	N	0	9000095010
BRZ	BRUTE-FORCE IN EXECUTION	N	0	9000097010
BSA	BRANCH, SIGN A	M	0	0000033000
BSS	BLOCK STARTED BY SYMBOL	P	1	0013000713
BUN	BRANCH, UNCONDITIONALLY	M	0	0000030000
CAA	CLEAR, ADD ABSOLUTE	M	0	0000110100
CAB	CLEAR A, B	M	0	0000545100
CAD	CLEAR, ADD	M	0	0000010100
CALL	CALL (SUBROUTINE)	P	1	0018000732
CAR	CLEAR A, R	M	0	0000345100
CFA	COMPARE FIELD A	M	0	0000018100
CFR	COMPARE FIELD R	M	0	0000118100
CHA	CHANGE	N	0	9000093010
CLA	CLEAR A	M	0	0000145100
CLB	CLEAR B	M	0	0000445100
CLL	CLEAR LOCATION	M	0	0000046000
CLR	CLEAR R	M	0	0000245100
CLT	CLEAR A, R, B (TOTAL)	M	0	0000745100
CRB	CLEAR R, B	M	0	0000645100
CSA	CLEAR, SUBTRACT ABSOLUTE	M	0	0000111100
CSU	CLEAR, SUBTRACT	M	0	0000011100
DBB	DECREASE B, BRANCH	M	0	0000021000
DEC	DECIMAL NUMBER	P	1	0042000754
DEL	DELETE	N	0	9000092010
DFL	DECREASE FIELD LOCATION	M	0	0000027000
DIV	DIVIDE	M	0	0000015000
DLB	DECREASE FIELD LOCATION, LOAD B	M	0	0000028000
DUP	DUPLICATE	P	1	0001000812
EJECT	EJECT PAGE	P	1	0006000816
END	END (OF PROGRAM)	P	1	0003000824
EQU	EQUALS	P	1	0001000828
EXT	EXTRACT	M	0	0000017000
FAA	FLOATING ADD ABSOLUTE	M	0	0000122100
FAD	FLOATING ADD	M	0	0000022100
FDV	FLOATING DIVIDE	M	0	0000025000
FMU	FLOATING MULTIPLY	M	0	0000024000
FSA	FLOATING SUBTRACT ABSOLUTE	M	0	0000123100
FSU	FLOATING SUBTRACT	M	0	0000023100
HLT	HALT	M	0	0000000000
IBB	INCREASE B, BRANCH	M	0	0000020000
IFL	INCREASE FIELD LOCATION	M	0	0000026000
INS	INSERT	N	0	9000091010
IOM	INTERROGATE OVERFLOW MODE	M	0	0000239100
KAD	KEYBOARD ADD	M	0	0000008000
LBC	LOAD B COMPLEMENT	M	0	0000142100
LDB	LOAD B	M	0	0000042100
LDR	LOAD R	M	0	0000041000
LOAD	LOAD (OBJECT TAPE)	P	1	0012000843
LSA	LOAD SIGN A	M	0	0000043000

MACRO	DEFINE MACRO OPERATION	P	1	0000000858
MDA	MAGNETIC-TAPE REWIND, DE-ACTIVATE	M	0	0000950100
MFC	MAGNETIC-TAPE FIELD SCAN	M	0	4000051010
MFS	MAGNETIC-TAPE FIELD SEARCH	M	0	4000050110
MIB	MAGNETIC-TAPE INTERROGATE, BRANCH	M	0	0000059100
MIE	MAGNETIC-TAPE INTERROGATE, END OF TAPE	M	0	0000159100
MIR	MAGNETIC-TAPE INITIAL WRITE, RECORD	M	0	0000055000
MIW	MAGNETIC-TAPE INITIAL WRITE	M	0	0000054000
MLS	MAGNETIC-TAPE LANE SELECT	M	0	0000450100
MOR	MAGNETIC-TAPE OVERWRITE, RECORD	M	0	0000057000
MOW	MAGNETIC-TAPE OVERWRITE	M	0	0000056000
MPB	MAGNETIC-TAPE POSITION BACKWARD	M	0	0000158100
MPE	MAG-TAPE POSITION END OF INFORMATION	M	0	0000258100
MPF	MAGNETIC-TAPE POSITION FORWARD	M	0	0000058100
MRD	MAGNETIC-TAPE READ	M	0	0000052000
MRR	MAGNETIC-TAPE READ, RECORD	M	0	0000053000
MRW	MAGNETIC-TAPE REWIND	M	0	0000850100
MSR	MAGNETIC-TAPE SUBROUTINE	N	0	9000080010
MTC	MAGNETIC-TAPE SCAN	M	0	0000051000
MTS	MAGNETIC-TAPE SEARCH	M	0	0000050100
MUL	MULTIPLY	M	0	0000014000
NOP	NO OPERATION	M	0	0000001000
OPDEF	DEFINE OPERATION-CODE	P	1	0000000862
OPSYN	DEFINE OPERATION-CODE SYNONYM	P	1	0000000889
ORG	ORIGIN LOCATION COUNTER	P	1	0005000900
PAUSE	PAUSE (TO CHANGE PAPER TAPE)	P	1	0002000911
PRB	PAPER-TAPE READ, BRANCH	M	0	0000004000
PRD	PAPER-TAPE READ	M	0	0000003000
PRI	PAPER-TAPE READ, INVERSE FORMAT	M	0	0000005000
PSR	PAPER-TAPE SUBROUTINE	N	0	9000082010
PWI	PAPER-TAPE WRITE INTERROGATE, BRANCH	M	0	0000007000
PWR	PAPER-TAPE WRITE	M	0	0000006000
REF	REFERENCE CHANGE	N	0	9000098010
REL	SET 'RELMOD'	P	1	0002000915
REM	REMARK	P	1	0007000921
RND	ROUND	M	0	0000016000
RTF	RECORD TRANSFER	M	0	0000029000
SEC	SECTION CHANGE	N	0	9000094010
SET	SET VALUE OF SYMBOL	P	1	0000000931
SEX	SECTION DEFINITION	N	1	0002000953
SLA	SHIFT LEFT A	M	0	0000049101
SLS	SHIFT LEFT A WITH SIGN	M	0	0000249101
SLT	SHIFT LEFT A AND R (TOTAL)	M	0	0000149101
SOH	SET OVERFLOW HALT MODE	M	0	0000139100
SOR	SET OVERFLOW REMEMBER MODE	M	0	0000039100
SPAC	REQUEST AND NAME SPAC SUBROUTINE	P	1	0006000962
SPACE	SPACE (ON LISTING)	P	1	0017001012
SPO	SUPERVISORY PRINT-OUT	M	0	0000009000

SRA	SHIFT RIGHT A	M	0	0000048101
SRS	SHIFT RIGHT A WITH SIGN	M	0	0000248101
SRT	SHIFT RIGHT A AND R (TOTAL)	M	0	0000148101
STA	STORE A	M	0	0000040100
STB	STORE B	M	0	0000240100
STP	STORE P	M	0	0000044000
STR	STORE R	M	0	0000140100
SUA	SUBTRACT ABSOLUTE	M	0	0000113100
SUB	SUBTRACT	M	0	0000013100
TCD	'TRANSFER CARD'	P	1	0011001030
ZZZ	ZERO CREATOR	N	1	0009001044