

```

*
*
* MACRO PACKAGE FOR META-APL ASSEMBLIES.
*
*
* THE SYNTAX FOR A META-APL ASSEMBLY PROGRAM IS AS FOLLOWS:
*   (TERMINALS ARE QUOTED,
*   PARENTHESES ARE METACHARACTERS,
*   A SUFFIX QUESTION MARK MEANS 0 OR 1 OCCURRENCES, AND
*   A SUFFIX ASTERISK MEANS 0 OR MORE OCCURRENCES.)
*
*   "PROGRAM"  GLØBSIZE "," LØCSIZE  CR
* (GLØBNAME "GLVAR"  CR)*
* (FNAME "FCN"  CR)*
* (
*FNAME "FUNCTION"  RESULT "," NSTMTS "," NARGS "," NLØCALS  CR
*   ("LØCAL"  LØCNAME ("," LØCNAME)*  CR)*
*   ("LØCAL"  ARGNAME ("," ARGNAME)*  CR)*
*   (
*   "STATEMENT"  STNØ  CR
*   (INSTRUCTION  CR)*
*   )*
*   "ENDFUNCTION"  CR
* )*
*   "ENDPROGRAM"  CR
*
*
* THIS MACRO OUTPUTS (TO THE LISTING FILE) THE VALUE OF A SYMBOL OR
* LIST OF SYMBOLS.
OUTSYM      MACRO      D
NNN          NARG
              RPT      NNN,(III=1,1)
              REM      [D(III)]=(SD(III))
              ENDR
              ENDM

```

*
*

* THIS MACRO BEGINS A META-APL ASSEMBLY PROGRAM.

PROGRAM MACRO D

* D(1) IS THE SIZE FOR THE GLOBAL SEGMENT;

* D(2) IS THE SIZE FOR THE LOCAL SEGMENT.

RELORG

DATA 0,0

DATA 0,0

DATA DFIRST, DLEN

DATA POFIRST, POLEN

DATA 0,0

EQU */2

EQU */2

DATA LFIRST, 1

DATA 0,0

DATA 0,0

BSS 2*3

DATA FREEBLOCK, 0

DATA 0,0

* COMMAND WORD

* STATUS WORD

* DATA SEGMENT DESCRIPTOR

* PROC SEGMENT DESCRIPTOR

* ALT PROC SEG DESCRIPTOR

DFIRST

GFIRST

* SBASE, SPTR

* FLAGS, LBASE

* TRAP INFORMATION

* SAVE NOABR

* ROVER

* BLOCKPTR

*

GLEN

EQU D(1)

LLEN

EQU D(2)

DLEN

EQU D(1)+D(2)

*

FCNO

EQU 0

*

ENDM

```

*
*
* THIS MACRO DECLARES A GLOBAL VARIABLE.
GLVAR      LMACRO      D
D(0).&:G   EQU         [* / 2 - GFIRST] - GLOBAL OFFSET
          OUTSYM      D(0).&:G
          DATA      UNDFTYPEB; DATA 0
          ENDM

```

```

*
*
*
*
* THIS MACRO DEFINES A FUNCTION NAME.
FCN        LMACRO      D
          IF          FCN0=0
          FREESPACE
          LOCSEG
POFIRST    EQU         * / 2
          ENDF
FCN0       EQU         FCN0+1
D(0).&:N    EQU         FCN0
          OUTSYM      D(0).&:N
          DATA      D(0).&:B; DATA 0
          DATA      0; DATA 0
          ENDM

```

```

*
*
* THIS MACRO ASSEMBLES THE FREE SPACE AREA.
FREESPACE MACRO
FREEBLOCK EQU      */2-CFIRST
FREESIZE EQU      GLEN-FREEBLOCK
* FREE BLOCK.
    IF      FREESIZE<5
    REM      NO ROOM FOR FREE SPACE
    ENDF
    DATA    THISFREEB; DATA FREESIZE-2
    DATA    FREEBLOCK+FREESIZE-2; DATA FREEBLOCK+FREESIZE-2
    BSS      2*[FREESIZE-2-3]
    DATA    0; DATA FREESIZE-2
* DUMMY BLOCK.
    DATA    0; DATA 0
    DATA    FREEBLOCK; DATA FREEBLOCK
*
    ENDM
*
THISFREEB EQU      100000B
*
*
* THIS MACRO ASSEMBLES THE LOCAL SEGMENT.
LOCSEG MACRO
LFIRST EQU      */2
    DATA    177777B,0
    DATA    P.N, CODEOFFSET*4
    BSS      2*[LLEN-2]
    ENDM
*
CODEOFFSET EQU      2
    
```

```

* "LBASE"
* FDESCR,PCTR
    
```

```

*
*
* THIS MACRO BEGINS A FUNCTION BODY.
FUNCTION   LMACRO      D
* D(0) IS THE FUNCTION NAME;
* D(1) IS THE RESULT FLAG;
* D(2) IS THE NUMBER OF STATEMENTS;
* D(3) IS THE NUMBER OF ARGUMENTS;
* D(4) IS THE NUMBER OF OTHER LOCALS.
*
* "PATCH UP" THE FUNCTION DIRECTORY ENTRY BY DEFINING BASE ADDRESS.
D(0).&:B   EQU          */2-PFIRST
          OUTSYM      D(0).&:B
*
* ASSEMBLE THE FUNCTION BODY HEADER.
          DATA [D(1)]*2+12+[D(2)];  DATA [D(3)]*2+12+[D(4)]
          DATA D(0).&:L;              DATA 0
*
* INITIALIZE THE BYTE POSITION COUNTER.
GENPOS     EQU          16
*
* DEFINE THE MACRO FOR DECLARING LOCAL VARIABLES (ARGS & OTHERS).
LOCAL      MACRO        E
TEMP       NARG
          RPT            TEMP,(I=1,1)
D(0).&E(1) EQU          LOCNO&7777B
LOCNO      EQU          LOCNO-1
          ENDR
          ENDM
*
* INITIALIZE THE LOCAL VARIABLE CURSOR.
LOCNO      EQU          -1
*
* DEFINE THE MACROS FOR REFERENCING LOCAL VARIABLES.
SHORTLOCAL MACRO        E
          GEN8            000B+[D(0).&E(1)&77B]
          ENDM
*
LONGLOCAL  MACRO        E
          GEN16           040000B+D(0).&E(1)
          ENDM
*
* DEFINE THE MACRO FOR INDICATING STATEMENT BOUNDARIES.
STATEMENT  MACRO        E
          IF              STNO+1=E(1)
          IF              E(1)<=D(2)
STNO       EQU          STNO+1
D(0).&($STNO) EQU      2*[*-2*PFIRST]+[GENPOS=8]-4*D(0).&:B
          OUTSYM         D(0).&($STNO)
          ELSE
          REM             TOO MANY STATEMENTS
          ENDF
          ELSE

```

```

      REM          STATEMENT NUMBER OUT OF SEQUENCE
      ENDF
      ENDM

*
* INITIALIZE THE STATEMENT COUNTER.
STN0      EQU          0
*
* DEFINE THE MACRO WHICH TERMINATES A FUNCTION BODY.
ENDFUNCTION MACRO
      IF          STN0=D(2)
      FLUSH
D(0).&:L    EQU          [*/2-PFIRST1-[D(0).&:B+2]
      OUTSYM      D(0).&:L
* GENERATE THE LINE TABLE.
      RPT          D(2),(I=1,1)
      DATA        D(0).&($I)
      ENDR
      IF          D(2)/2*2#D(2)
      DATA        -1
      ENDF
      ELSE
      REM          TOO FEW STATEMENTS
      ENDF
      ENDM

*
* END OF "FUNCTION" MACRO.
      ENDM

```

```

*
*
* THIS MACRO OUTPUTS ONE BYTE OF CODE.

```

```

GEN8      MACRO      D
           IF          GENPOS=16
GENWORD    EQU        [D(1)]*2+8
GENPOS     EQU        8
           ELSE
           DATA       GENWORD! [D(1)]
GENPOS     EQU        16
           ENDF
           ENDM

```

```

*
*
* THIS MACRO OUTPUTS TWO BYTES OF CODE.

```

```

GEN16     MACRO      D
           IF          GENPOS=16
           DATA       D(1)
           ELSE
           DATA       GENWORD! [[D(1)]&177400B]/2+8
GENWORD    EQU        [[D(1)]&000377B]*2+8
           ENDF
           ENDM

```

```

*
*
* THIS MACRO "FLUSHES OUT" THE LAST WORD OF CODE.

```

```

FLUSH     MACRO
           IF          GENPOS=8
           DATA       GENWORD! 0
GENPOS     EQU        16
           ENDF
           IF          */2*2#*
           DATA       0
           ENDF
           ENDM

```

```

*
*
LONGGLOBAL MACRO      D
                GEN16   050000B+D(1).&:G
                ENDM

*
*
* MEMREF <SUBOPCODE> MUST BE FOLLOWED BY A LOCAL OR GLOBAL REFERENCE.
MEMREF        MACRO      D
                GEN8      140B+D(1)
                ENDM

*
* ASINDEXED IS 00 THROUGH 17B.
ASSIGN        EQU        20B
ASNØRESULT    EQU        21B
REFERENCE     EQU        22B
TESTDEF       EQU        23B
* 24B THROUGH 37B ARE UNUSED.
*
*
GENSCALAR     MACRO      D
                GEN8      200B+D(1)
                ENDM

*
IDENTITY      EQU        00B
NEGATIVE      EQU        01B
FLOOR         EQU        02B
CEILING       EQU        03B
MAGNITUDE     EQU        04B
NOT           EQU        05B
TESTNUM       EQU        06B
CONVERT       EQU        07B
SUM           EQU        10B
DIFFERENCE    EQU        11B
PRODUCT       EQU        12B
QUOTIENT      EQU        13B
AND           EQU        14B
ØR            EQU        15B
LESS          EQU        16B
EQUAL         EQU        17B
*
*
INDEX         MACRO      D
                GEN8      220B+D(1)
                ENDM

*
*
SHORTCONST    MACRO      D
                IF        D(1)<0
                GEN16      120000B+[010000B-D(1)]
                ELSE
                GEN16      120000B+D(1)
                ENDF

```



```

      ENDM
*
*
ETC      MACRO      D
      GEN8      300B+D(1)
      IF      CONSCALAR=D(1)
      GCONSCALAR D(2)
      ELSF      CONVEC=D(1)
TEMP      NARG
      IF      TEMP=2
      GCONVEC      D(2)
      ELSF      TEMP=3
      GCONVEC      D(2),D(3)
      ELSE
      REM      BAD OPERANDS FOR CONVEC
      ENDF
      ELSF      CALLFCN=D(1)
      GEN16      D(2)*2+12+D(3).&:N      NARGS,FCNO
      ENDF
      ENDM

```

```

*
UNDEFINED EQU      00B
EAT1      EQU      01B
INTERCHANGE EQU      02B
SETORIGIN EQU      03B
GETORIGIN EQU      04B
SHAPE      EQU      05B
RESHAPE      EQU      06B
RAVEL      EQU      07B
CATENATE      EQU      10B
INDEXGEN      EQU      11B
CONSCALAR      EQU      12B
CONVEC      EQU      13B
BRANCH      EQU      14B
GO      EQU      15B
GOTRUE      EQU      16B
GOFALSE      EQU      17B
RETURNF      EQU      20B
CALLFCN      EQU      21B
BKPTTRAP      EQU      22B
ATTNTRAP      EQU      23B

```

NOOP EQU 22B

```

*
*
GCONSCALAR MACRO      D
      GEN16      D(1); GEN16 D(2)
      ENDM

```

```

*
*
GCONVEC      MACRO      D
* D(1) IS THE NUMBER OF ELEMENTS;
* D(2) THROUGH D(NARG) ARE ELEMENTS.
* IF NECESSARY, GCONSCALAR CALLS CAN BE USED FOR FURTHER ELEMENTS.
      GEN8      D(1)
* MOVE TO NEXT FULLWORD.

```

8 Jan 73

TEMP FLUSH
 NARG
 RPT [TEMP-1],(I=2,1)
 GCONSCALAR D(I)
 ENDR
 ENDM

*

* DESCRIPTOR TYPES

INTTYPEB EQU 020000B
CHTYPEB EQU 010000B

*

*

ENDPROGRAM MACRO

P0LEN EQU */2-PFIRST
 OUTSYM GFIRST,FREEBLOCK,GLEN,LFIRST,LLEN,PFIRST,P0LEN
 END
 ENDM

*

*

FREEZE
END

*
 *
 ENDPROGRAM MACRO
 POLEN EQU */2-PFIRST
 OUTSYM GFIRST,FREEBLOCK,CLEN,LFIRST,LLEN,PFIRST,PLEN
 ENDM

-Y

Y

FREEZ
 : 10

E.&N.&L

```

* META-APL ASSEMBLY TEST PROGRAM
*
* DEL R ← F N
* [1] BRANCH (2 4)[IORG + N>0]
* [2] R ← 1
* [3] BRANCH 0
* [4] R ← N TIMES F N-1
* [5] RETURN
* DEL

```

```

* PROGRAM 40,40
*
* P FCN
* F FCN
* P
* FUNCTION 0,1,0,0
* STATEMENT 1
* SHORTCONST 2
* ETC CALLFCN,1,F
* ETC ATTNTRAP
* ENDFUNCTION
*
* F FUNCTION 1,5,1,1
* LOCAL R
* LOCAL N
*
* STATEMENT 1
* SHORTCONST 0
* SHORTLOCAL N
* ETC INTERCHANGE
* GENSCALAR LESS
* ETC GETORIGIN
* GENSCALAR SUM
* ETC CONVEC,2,((INTTYPEB,2),(INTTYPEB,4))
* INDEX 1
* ETC BRANCH
*
* STATEMENT 2
* SHORTCONST 1
* MEMREF ASNORESULT; SHORTLOCAL R
*
* STATEMENT 3
* SHORTCONST 0
* ETC BRANCH
*
* STATEMENT 4
* SHORTCONST 1
* SHORTLOCAL N
* GENSCALAR DIFFERENCE
* ETC CALLFCN,1,F
* SHORTLOCAL N
* GENSCALAR PRODUCT

```

MEMREF ASNØRESULT; SHØRTLØCAL R
*
STATEMENT 5
ETC RETURNF
*
ENDFUNCTION
*
ENDPRØGRAM

file: TP10X

```
PROGRAM      100,40

* A      GLVAR
* B      GLVAR
* C      GLVAR
* I      GLVAR
* J      GLVAR
* K      GLVAR
*
* P      FCN
*
* P      FUNCTION      0,14,0,0
*
* SETORIGIN 1
      STATEMENT 1
      SHORTCONST 1
      ETC      SETORIGIN
      ETC      EAT1
* A←(2 2 2) RESHAPE INDEXGEN 8
      STATEMENT 2
      SHORTCONST 8
      ETC      INDEXGEN
      ETC      CONVEC,3,((INTTYPEB,2),(INTTYPEB,2),(INTTYPEB,2))
      ETC      RESHAPE
      MEMREF    ASNORESULT; LONGGLOCAL A
* B←C←(SHAPE A) RESHAPE 0
      STATEMENT 3
      SHORTCONST 0
      LONGGLOCAL A
      ETC      SHAPE
      ETC      RESHAPE
      MEMREF    ASSIGN; LONGGLOCAL C
      MEMREF    ASNORESULT; LONGGLOCAL B
* I←1
      STATEMENT 4
      SHORTCONST 1
      MEMREF    ASNORESULT; LONGGLOCAL I
* J←1
      STATEMENT 5
      SHORTCONST 1
      MEMREF    ASNORESULT; LONGGLOCAL J
* K←1
      STATEMENT 6
      SHORTCONST 1
      MEMREF    ASNORESULT; LONGGLOCAL K
* B[I.,J.,K]←A[I.,J.,K]
      STATEMENT 7
      LONGGLOCAL K; LONGGLOCAL J; LONGGLOCAL I
      LONGGLOCAL A
      INDEX      3
      LONGGLOCAL K; LONGGLOCAL J; LONGGLOCAL I
      MEMREF    3; LONGGLOCAL B
      ETC      EAT1
* K←K+1
```

```

STATEMENT 8
SHORTCONST 1
LONGGLOBAL K
GENSCALAR SUM
MEMREF ASNORESULT; LONGGLOBAL K
* BRANCH 7 PRODUCT INDEXGEN K LESS 3
STATEMENT 9
SHORTCONST 3
LONGGLOBAL K
GENSCALAR LESS
ETC INDEXGEN
SHORTCONST 7
GENSCALAR PRODUCT
ETC BRANCH

* J-J+1
STATEMENT 10
SHORTCONST 1
LONGGLOBAL J
GENSCALAR SUM
MEMREF ASNORESULT; LONGGLOBAL J
* BRANCH 6 PRODUCT INDEXGEN J LESS 3
STATEMENT 11
SHORTCONST 3
LONGGLOBAL J
GENSCALAR LESS
ETC INDEXGEN
SHORTCONST 6
GENSCALAR PRODUCT
ETC BRANCH

* I-I+1
STATEMENT 12
SHORTCONST 1
LONGGLOBAL I
GENSCALAR SUM
MEMREF ASNORESULT; LONGGLOBAL I
* BRANCH 5 PRODUCT INDEXGEN I LESS 3
STATEMENT 13
SHORTCONST 3
LONGGLOBAL I
GENSCALAR LESS
ETC INDEXGEN
SHORTCONST 5
GENSCALAR PRODUCT
ETC $ BRANCH

* ATTNTRAP
STATEMENT 14
ETC ATTNTRAP

*
ENDFUNCTION

*
ENDPROGRAM

```

PROGRAM 40,40

CLVAR

FCN

FCN

FCN

FUNCTION 0,2,0,0

STATEMENT 1

MEMREF REFERENCE; LONGGLOBAL A

ETC CALLFCN,1,F

STATEMENT 2

ETC /TTINTRAP

ENDFUNCTION

FUNCTION 0,3,1,1

LOCAL L,X

STATEMENT 1

MEMREF REFERENCE; LONGLOCAL L

MEMREF REFERENCE; LONGLOCAL X

ETC CALLFCN,2,G

STATEMENT 2

ETC ATTINTRAP

STATEMENT 3

ETC RETURNF

ENDFUNCTION

FUNCTION 0,3,2,0

LOCAL X,Y

STATEMENT 1

SHORTCONST 1

MEMREF ASNORESULT; LONGLOCAL X

STATEMENT 2

SHORTCONST 2

MEMREF ASNORESULT; LONGLOCAL Y

STATEMENT 3

ETC RETURNF

ENDFUNCTION

ENDPROGRAM

8 Jan 73

file: TPAKTH

```

PROGRAM      200,50

*
A            GLVAR
B            GLVAR
I            GLVAR
R            GLVAR
*
P            FCN
D            FCN
*
P            FUNCTION      0,16,0,0
* SETORIGIN 1
      STATEMENT 1
      SHORTCONST 1
      ETC        SETORIGIN
      ETC        EAT1
* A←1.0 PRODUCT INDEXGEN 25
      STATEMENT 2
      SHORTCONST 25
      ETC        INDEXGEN
      ETC        CONSCALAR,(040000B,000200B)
      GENSCALAR  PRODUCT
      MEMREF     ASNORESULT; LONGGLOBAL A
* B←A+A
      STATEMENT 3
      LONGGLOBAL A
      LONGGLOBAL A
      GENSCALAR  SUM
      MEMREF     ASNORESULT; LONGGLOBAL B
* I←1.0
      STATEMENT 4
      ETC        CONSCALAR,(040000B,000200B)
      MEMREF     ASNORESULT; LONGGLOBAL I
* LOOP: R←D[A.,B.,I]
      STATEMENT 5
      LONGGLOBAL I; LONGGLOBAL B; LONGGLOBAL A
      ETC        CALLFCN,3,D
      MEMREF     ASNORESULT; LONGGLOBAL R
* ATTNTRAP
      STATEMENT 6
      ETC        ATTNTRAP
* BRANCH LOOP PRODUCT INDEXGEN (I←I+1)<5
      STATEMENT 7
      SHORTCONST 5
      SHORTCONST 1
      LONGGLOBAL I
      GENSCALAR  SUM
      MEMREF     ASSIGN; LONGGLOBAL I
      GENSCALAR  LESS
      ETC        INDEXGEN
      SHORTCONST 5
      GENSCALAR  PRODUCT
      ETC        BRANCH
* R←(0 0 1 1) AND (0 1 0 1)

```

8 Jan 73

```

STATEMENT 8
ETC CONVEC,4
GCONSCALAR INTTYPEB,0
GCONSCALAR INTTYPEB,1
GCONSCALAR INTTYPEB,0
GCONSCALAR 040000B,000200B
ETC CONVEC,4
GCONSCALAR INTTYPEB,0
GCONSCALAR INTTYPEB,0
GCONSCALAR INTTYPEB,1
GCONSCALAR 040000B,000200B
GENSCALAR AND
MEMREF ASNORESULT; LONGGLOBAL R

* ATTNTRAP
STATEMENT 9
ETC ATTNTRAP
* R+(0 0 1 1) OR (0 1 0 1)
STATEMENT 10
ETC CONVEC,4
GCONSCALAR INTTYPEB,0
GCONSCALAR INTTYPEB,1
GCONSCALAR INTTYPEB,0
GCONSCALAR INTTYPEB,1
ETC CONVEC,4
GCONSCALAR INTTYPEB,0
GCONSCALAR INTTYPEB,0
GCONSCALAR INTTYPEB,1
GCONSCALAR INTTYPEB,1
GENSCALAR OR
MEMREF ASNORESULT; LONGGLOBAL R

* ATTNTRAP
STATEMENT 11
ETC ATTNTRAP
* R-A<B
STATEMENT 12
LONGGLOBAL B; LONGGLOBAL A
GENSCALAR LESS
MEMREF ASNORESULT; LONGGLOBAL R

* ATTNTRAP
STATEMENT 13
ETC ATTNTRAP
* R-A=B
STATEMENT 14
LONGGLOBAL B; LONGGLOBAL A
GENSCALAR EQUAL
MEMREF ASNORESULT; LONGGLOBAL R

* ATTNTRAP
STATEMENT 15
ETC ATTNTRAP
* RETURN
STATEMENT 16
ETC RETURNF

* DEL
ENDFUNCTION

```

8 Jan 73

```

* DEL R-D[A.,B.,N]
D      FUNCTION 1,9,3,1
      LOCAL R
      LOCAL A,B,N
* BRANCH (2 3 4 5 6 7 8 9)[N]
      STATEMENT 1
      SHORTLOCAL N
      ETC CONVEC,8
      GCONSCALAR INTTYPEB,2
      GCONSCALAR INTTYPEB,3
      GCONSCALAR INTTYPEB,4
      GCONSCALAR INTTYPEB,5
      GCONSCALAR INTTYPEB,6
      GCONSCALAR INTTYPEB,7
      GCONSCALAR INTTYPEB,8
      GCONSCALAR INTTYPEB,9
      INDEX 1
      ETC BRANCH
*
S      MACRO D
* D(1) IS THE STATEMENT NUMBER,
* D(2) IS THE (SCALAR) OPERATOR.
      STATEMENT D(1)
      SHORTLOCAL B; SHORTLOCAL A
      GENSCALAR D(2)
      MEMREF ASSIGN; SHORTLOCAL R
      SHORTCONST 0
      ETC CATENATE
      ETC BRANCH
      ENDM
*
* BRANCH 0,R+A+B
      S 2,SUM
* BRANCH 0,R+A-B
      S 3,DIFFERENCE
* BRANCH 0,R+A PRODUCT B
      S 4,PRODUCT
* BRANCH 0,R+A QUOTIENT B
      S 5,QUOTIENT
* BRANCH 0,R+A AND B
      S 6,AND
* BRANCH 0,R+A OR B
      S 7,OR
* BRANCH 0,R+A LESS B
      S 8,LESS
* BRANCH 0,R+A EQUAL B
      S 9,EQUAL
* DEL
      ENDFUNCTION
*
      ENDPROGRAM

```