File created: 25-Jan-93 16:56:13 **{PELE:MV:ENVOS}<LISPCORE>LIBRARY>DLAP.;1**

changes to:
```
(PROPS (FLOAT DOPVAL)
       (\FLOAT.BOX DOPVAL))
```

previous date: 17-Nov-92 01:01:02 {PELE:MV:ENVOS}<LISPCORE>SOURCES>DLAP.;11

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ DLAPCOMS
       (;; Assembler for the Byte Compiler.
        (FNS C.FLOATBOX C.FLOATUNBOX DASSEM.DASSEM DASSEM.DWRITEFN DASSEM.SAVELOCALVARS DASSEM.DSTOREFNDEF
             DASSEM.DPRINTLAP DASSEM.EQCONSTANTP DASSEM.MATCHVARS DASSEM.COUNTVARS DASSEM.CANSHAREBINDING)
        (CONSTANTS NARGMAX NLOCALMAX NFREEMAX)
        (FNS DASSEM.DASMBIND DASSEM.DSTOREFN DASSEM.ASMAJ)
        (VARS (EMFLAG)
              (COMPILEMODE 'D))
        (PROP (MOPVAL AJSIZES)
              JUMP FJUMP TJUMP NTJUMP NFJUMP)
        (PROP DOPVAL * DOPVALS)
        (VARS CONSTOPS (COMPILE.ARG.FAST.FLG)
              (IPLUSNFLG))
        (ADDVARS (8BITEXTS DCOM))
        (ADDVARS (MACROPROPS DMACRO ALTOMACRO BYTEMACRO MACRO))
        (ADDVARS (COMPILERMACROPROPS DMACRO ALTOMACRO BYTEMACRO MACRO))
        (VARS (BYTEASSEMFN 'DASSEM.DASSEM)
              (MAXBVALS 15)
              (BYTECOMPFLG T)
              (SELECTQFMEMB NIL)
              (LAMBDANOBIND T)
              (SELECTVARTYPES '(AVAR HVAR))
              [CONST.FNS '((NIL (1 CAR (CONST))
                                (1 CDR (CONST))
                                (1 NULL (CONST . T))
                                (2 EQ (FN 1 . NULL)))
                           (0 (2 ITIMES2 (POP)
                                 (CONST . 0))
                              (2 LOGAND2 (POP)
                                 (CONST . 0))
                              (2 IPLUS (FN 1 . FIX))
                              (2 LOGOR2 (FN 1 . FIX))
                              (2 \ADDBASE))
                           (1 (2 ITIMES2 (FN 1 . FIX]
              (MERGEFRAMEFLG T)
              (MERGEFRAMEMAX 2)
              (CLEANFNLIST '(NTYPX EQ AND OR CONS LIST FMEMB MEMB GETP SUB1 ADD1 ZEROP ELT ILESSP LLSH LRSH
                                   IPLUS IDIFFERENCE \ARG0 GETHASH \ADDBASE \GETBASEPTR \GETBASEBYTE \GETBASE
                                   \GETBASEFIXP \GETBASESTRING \VAG2 \ADDBASE))
              (OPCODEPROP 'DOPVAL)
              (VCONDITIONALS '(ARRAYP FIXP FLOATP LISTP SMALLP STACKP NUMBERP))
              (CONDITIONALS '(EQ IGREATERP NULL GREATERP LESSP ILESSP))
              (CONSTFNS '(IPLUS SUB1 ADD1 ZEROP LLSH LRSH IDIFFERENCE))
              (MAXARGS 80)
              (XVARFLG NIL)
              (NOFREEVARFNS '(RPLACA RPLACD PUTHASH SETA))
              (CLEANFNTEST 'DASSEM.CLEANFNTEST)
              (EQCONSTFN 'DASSEM.EQCONSTANTP))
        (ADDVARS (NUMBERFNS LLSH1 LRSH1 LLSH8 LRSH8))
        (CONSTANTS (SHALLOWFLG NIL)
                (SPAGHETTIFLG T))
        (FNS DASSEM.CLEANFNTEST)
        (OPTIMIZERS ATOM EVALV FRPLACA GETATOMVAL LIST LITATOM MINUSP IEQP FASSOC SETATOMVAL SYSTEMTYPE
               SPREADAPPLY*)
        (PROP DMACRO FGETD FGREATERP FLESSP FMEMB FRPLACD GETD GREATERP IGREATERP ILESSP LESSP LLSH LRSH
              PRINTNUM RPLACD \FLOATBOX \FLOATUNBOX)
        (FNS COMP.RPLACD COMP.SHIFT COMP.COMPARENUM COMP.GETD COMP.FMEMB)
        (PROP PROPTYPE DMACRO)
        (COMS                                                           ; COMP.GETBASE
              (OPTIMIZERS \GETBASEBYTE \PUTBASEBYTE \HILOC \LOLOC \VAG2)
              (PROP DMACRO \GETBASE \GETBASEPTR \PUTBASE \PUTBASEPTR \RPLPTR \GETBITS \PUTBITS)
              (FNS COMP.GETBASE COMP.GETBASEBITS))
        (COMS (FNS COMP.SPREADFN)
              (OPTIMIZERS NCONC APPEND))
        (COMS                                                           ; CAPPLYFN
              (PROP DMACRO NILAPPLY .PUSHNILS. SPREADAPPLY .SPREAD. .EVALFORM. .CALLAFTERPUSHINGNILS. APPLY*)
              (PROP DOPVAL .SPREADCONS. .SWAPNIL.)
              (FNS COMP.PUSHNILS COMP.SPREAD COMP.EVALFORM COMP.PUSHCALL COMP.APPLY*))
        (COMS                                                             ; for ARG and SETARG
              (PROP DMACRO ARG SETARG NAMEDLET)
              (FNS COMP.ARG COMP.SETARG COMP.NAMEDLET))
        (COMS (PROP DMACRO LOADTIMECONSTANT)
              (VARS LOADTIMECONSTANTMARKER))
        (PROP FILETYPE DLAP)
        (DECLARE%: EVAL@COMPILE DONTCOPY (RECORDS DASM)
               (GLOBALVARS FVINDEXHARRAY)
               (MACROS PARENTP AST OPCOUNT)
               (MACROS CHECKRANGE)
               DONTEVAL@LOAD
               (FILES (LOADCOMP)
                      BYTECOMPILER LLCODE))))
```

;; Assembler for the Byte Compiler.

```
(DEFINEQ

(C.FLOATBOX
  [LAMBDA (A)                                                       (* lmm "29-Dec-84 11:39")
    (SELECTQ COMPILE.CONTEXT
        (EFFECT (COMP.VALN A 'EFFECT))
        (PROGN (COMP.VAL1 A)
               (COMP.FLOATBOX])

(C.FLOATUNBOX
  [LAMBDA (A)                                                       (* lmm "29-Dec-84 11:44")
    (SELECTQ COMPILE.CONTEXT
        (EFFECT (COMP.VALN A 'EFFECT))
        (COMP.VAL1 A '(UNBOXED . FLOAT])

(DASSEM.DASSEM
  [LAMBDA (FN CC)                                                   ; Edited 17-Nov-92 00:58 by sybalsky:mv:envos
    (PROG ((ARGTYPE (fetch (COMINFO COMTYPE) of CC))
           (ARGS (fetch (COMINFO ARGS) of CC))
           (CODE (fetch (COMINFO CODE) of CC))
           NARGS NLOCALS FREEVARS NFREEVARS ORG CD (VARCOUNT 0)
           LOCALS
           (FRAMENAME FN))
          (DECLARE (SPECVARS VARCOUNT FRAMES CD CODELOC))
          (fetch (DASM CLEAR) of T)
          (COND
             ((AND (EQ ARGTYPE 2)
                   ARGS)
              [push CODE (create OP
                                OPNAME _ 'FN
                                OPARG _ '(0 . \MYARGCOUNT))
                    (create OP
                           OPNAME _ 'BIND
                           OPARG _ (CONS NIL (SETQ ARGS (fetch (COMINFO TOPFRAME) of CC]
              (replace NVALS of ARGS with 1)
              (replace NNILS of ARGS with 0)
              (SETQ ARGS NIL)
              (SETQ NARGS 0))
             (T (DASSEM.COUNTVARS ARGS)
                (SETQ NARGS VARCOUNT)))
          (PROGN (PROG ((LL CODE)
                        X A D FREELST FRAMES)
                       (DECLARE (SPECVARS FRAMES))
                   LP  (COND
                          ((NULL LL)
                           (GO OUT)))
                       [SETQ A (fetch OPARG of (SETQ X (CAR LL]
                   PR  (SELECTQ (fetch OPNAME of X)
                           (CONST (COND
                                     ((EQ (fetch OPNAME of (SETQ D (CADR LL)))
                                          'FN)
                                      (SELECTQ (CDR (fetch OPARG of D))
                                          ((IDIFFERENCE IPLUS2)
                                               (COND
                                                  ((AND (NOT OPTIMIZATIONSOFF)
                                                        IPLUSNFLG
                                                        (EQ (CAR (fetch OPARG of D))
                                                            2)
                                                        (IGEQ A 0)
                                                        (ILEQ A 255))
                                                   (RPLACA LL (SELECTQ (CDR (fetch OPARG of D))
                                                                  (IDIFFERENCE 'IDIFFERENCE.N)
                                                                  'IPLUS.N))
                                                   (RPLACA (CDR LL)
                                                          A)
                                                   (SETQ LL (CDR LL))
```

```
                                         (GO LP))))
                                 (\CALLME (COND
                                              ((EQ (CAR (fetch OPARG of D))
                                                   1)
                                               (SETQ FRAMENAME A)
                                               (RPLNODE2 LL (CDDR LL))
                                               (GO LP))))
                                 NIL)))
                        (COND
                           ((FASSOC A CONSTOPS)         ; HAS OPCODE

                            )
                           ((AND (FIXP A)
                                 (IGEQ A -256)
                                 (ILEQ A 65535))
                            [SETQ LL (PROG1 (CDR LL)
                                            [RPLACA LL (COND
                                                          ((ILESSP A 0)
                                                           (push (CDR LL)
                                                                 (IPLUS 256 A))
                                                           'SNIC)
                                                          ((IGREATERP A 255)
                                                           (push (CDR LL)
                                                                 (LRSH A 8)
                                                                 (LOGAND A 255))
                                                           'SICX)
                                                          (T (push (CDR LL)
                                                                   A)
                                                             'SIC])]
                            (GO LP))))
                  (BIND (PROG [(FRAME (CDR A))
                               (VARS (fetch (FRAME VARS) of (CDR A]
                              (DECLARE (SPECVARS FRAME))
                                                        ; frame is used free below DASSEM.MATCHVARS
                              [COND
                                 ((NEQ FRAME TOPFRAME)
                                  (for VAR in VARS when (EQ (CAR VAR)
                                                            'HVAR)
                                     do                 ; eliminate name of LOCALVAR variable
                                        (RPLACD VAR NIL]
                              (COND
                                 [(NULL LOCALS)         ; no local variables seen yet.  Assign var numbers sequentially
                                  (DASSEM.COUNTVARS (SETQ LOCALS (APPEND VARS]
                                 (T                     ; try to share binding pointers with some previously seen local
                                                        ; variables
                                    (DASSEM.MATCHVARS VARS LOCALS)))
                                                        ; remember this frame as having been seen
                              (push FRAMES (CDR A))))
                  (GVAR [SETQ LL (PROG1 (CDR LL)
                                        [RPLNODE LL (COND
                                                       ((EQ X (CAR LL))
                                                        'GVAR)
                                                       (T 'GVAR_))
                                               (COND
                                                  [(FMEMB :4-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE
                                                                   *BC-MACRO-ENVIRONMENT*))
                                                   (CONS 0 (CONS 0 (CONS 0 (CONS (CONS 'ATOM A)
                                                                                 (CDR LL]
                                                  [(FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE
                                                                   *BC-MACRO-ENVIRONMENT*))
                                                   (CONS 0 (CONS 0 (CONS (CONS 'ATOM A)
                                                                         (CDR LL]
                                                  (T (CONS 0 (CONS (CONS 'ATOM A)
                                                                   (CDR LL])]
                        (GO LP))
                  (FVAR [COND
                           [(SETQ D (FASSOC A FREELST))
                                                        ; count how often each var occurs
                            (FRPLACD (CDR D)
                                   (ADD1 (CDDR D]
                           (T (SETQ FREELST (CONS (CONS A (CONS (CAR X)
                                                                0))
                                                  FREELST])
                  (SETQ [SETQ A (fetch OPARG of (SETQ X (fetch OPARG of X]
                        (GO PR))
                  NIL)
              (SETQ LL (CDR LL))
              (GO LP)
         OUT  (SETQ A 0)
              [MAPC [SORT FREELST (FUNCTION (LAMBDA (X Y)
                                              (IGREATERP (CDDR X)
                                                     (CDDR Y]
                    (FUNCTION (LAMBDA (X)
                                (replace FREEVARINDEX of (CAR X) with A)
                                (ADD1VAR A]             ; Assign numbers to the free variables (most frequent first)
              [MAPC FREELST (FUNCTION (LAMBDA (X)
                                         (FRPLACD X (PROG1 (CAR X)
                                                                  (FRPLACA X (CADR X)))]
                        (SETQ FREEVARS FREELST)))
       ;; SCAN CODE
            (SETQ NLOCALS (IDIFFERENCE VARCOUNT NARGS))
            (CHECKRANGE NARGS NARGMAX 'ARGS)
            (CHECKRANGE NLOCALS NLOCALMAX 'LOCALS)
            (SETQ NFREEVARS (LENGTH FREEVARS))
            (CHECKRANGE NFREEVARS NFREEMAX 'FREEVARS)
            (PROGN                                                ; TURN INTO REAL CODE
                   (PROG ((CODELOC 0)
                          (LL CODE)
                          OP X D A JL N)
                     LP  (COND
                            ((NULL LL)
                             (SETQ CD (OPT.DREV CD))
                             (OPT.RESOLVEJUMPS (OPT.DREV JL)
                                    'AJSIZES
                                    (FUNCTION DASSEM.ASMAJ))
                             (RETURN)))
                         (SETQ X (CAR LL))
                         (COND
                            ((NLISTP X)
                             (AST X)
                             (GO NEXT)))
                         (SETQ A (fetch OPARG of X))
                     ;; Dispatch on the main opcode type:
                         (SELECTQ (SETQ OP (fetch OPNAME of X))
                             ((AVAR HVAR)                         ; Variable references
                                  [SETQ OP (COND
                                              ((ILESSP (SETQ A (fetch VARINDEX of X))
                                                      NARGS)
                                               '(IVAR . IVARX))
                                              (T (SETQ A (IDIFFERENCE A NARGS))
                                                 '(PVAR . PVARX]
                                  [COND
                                     ((ILESSP A (OPCOUNT (CAR OP)))
                                      (AST (LIST (CAR OP)
                                                 A)))
                                     (T (AST (CDR OP))
                                        (AST (LLSH A 1])
                             (FN                                  ; Function calls (includes many primitives like IPLUS2)
                                 [COND
                                    ((LISTP (SETQ D (CDR A)))
                                     (OR (EQ (CAR D)
                                             'OPCODES)
                                         (OPT.COMPILERERROR))
                                     (for X in (CDR D) do (AST X)))
                                    [(SETQ D (GETP D 'DOPVAL))    ; A fn has DOPVAL
                                     (PROG ((N (CAR A))
                                            (F (CDR A)))
                                       OPLP
                                           (COND
                                              ((NLISTP D))
                                              [(OR (EQ [CAR (SETQ A (COND
                                                                       ((FIXP (CAR D))
                                                                        (PROG1 D (SETQ D)))
                                                                       (T (CAR D]
                                                       N)
                                                   (NULL (CAR A)))
                                               ;; Arg count matches the DOPVAL's needs, so emit it:
                                               (COND
                                                  ((LISTP (SETQ D (CDR A)))
                                                   (RETURN (MAPC D (FUNCTION (LAMBDA (X)
                                                                                 (AST X]
                                              ((ILESSP N (CAR A)) ; A fn with DOPVAL supplied too few args
                                               (SETQ LL (CONS (create OP
                                                                     OPNAME _ 'FN
                                                                     OPARG _ (CONS (CAR A)
                                                                                   F))
                                                              (CDR LL)))
                                                                  ; put out NIL's and change # args.
                                               (FRPTQ (IDIFFERENCE (CAR A)
                                                              N)
                                                      (SETQ LL (CONS OPNIL LL)))
                                               (GO LP))
                                              ((NULL (CDR D))     ; A fn with DOPVAL supplied too many args
                                               (SETQ LL (CONS (create OP
                                                                     OPNAME _ 'FN
                                                                     OPARG _ (CONS (CAR A)
                                                                                   F))
                                                              (CDR LL)))
                                               (FRPTQ (IDIFFERENCE N (CAR A))
                                                      (SETQ LL (CONS OPPOP LL)))
                                               (GO LP))
```

```
                    (T (SETQ D (CDR D))
                       (GO OPLP)))
             APPLY
                 (SETQ LL (APPLY* D (fetch OPARG of X)
                                 LL]
          (T ;; Function is neither an opcode nor a DOPVAL, so emit the function call:
             (SELECTQ (CAR A)
                 (0 (AST 'FN0)
                    (DASSEM.DSTOREFN (CDR A)))
                 (1 (AST 'FN1)
                    (DASSEM.DSTOREFN (CDR A)))
                 (2 (AST 'FN2)
                    (DASSEM.DSTOREFN (CDR A)))
                 (3 (AST 'FN3)
                    (DASSEM.DSTOREFN (CDR A)))
                 (4 (AST 'FN4)
                    (DASSEM.DSTOREFN (CDR A)))
                 (PROGN (AST 'FNX)
                        (AST (CAR A))
                        (DASSEM.DSTOREFN (CDR A])
((JUMP FJUMP TJUMP NTJUMP NFJUMP)
     (push JL (create JD
                      JPT _ (push CD X)
                      JMIN _ CODELOC))
     [add CODELOC (CAAR (GETP OP 'AJSIZES])
(TAG (replace (TAG JD) of X with (SETQ D (create JD
                                                 JMIN _ CODELOC)))
      (SETQ JL (CONS D JL)))
(CONST [COND
           ((SETQ D (FASSOC A CONSTOPS))
            (AST (CDR D)))
           ((LITATOM A)
            (AST 'ACONST)
            (AST 0)
            (COND
               ((FMEMB :4-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE
                                      *BC-MACRO-ENVIRONMENT*))
                (AST 0)
                (AST 0))
               ((FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE
                                      *BC-MACRO-ENVIRONMENT*))
                (AST 0)))
            (AST (CONS 'ATOM A)))
           (T (AST 'GCONST)
              (AST 0)
              (COND
                 ((FMEMB :4-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE
                                        *BC-MACRO-ENVIRONMENT*))
                  (AST 0)
                  (AST 0))
                 ((FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE
                                        *BC-MACRO-ENVIRONMENT*))
                  (AST 0)))
              (AST (CONS 'PTR A])
(SETQ (SELECTQ (fetch OPNAME of A)
           ((AVAR HVAR)
                [COND
                   ((ILESSP (SETQ D (fetch VARINDEX of A))
                           NARGS)
                    (AST 'IVARX_)
                    (AST (LLSH D 1)))
                   (T (SETQ D (IDIFFERENCE D NARGS))
                      (COND
                         ([AND (EQ (fetch OPNAME of (CADR LL))
                                   'POP)
                               (ILESSP D (OPCOUNT 'PVAR_^]
                          (SETQ LL (CDR LL))
                          (AST (LIST 'PVAR_^
                                     D)))
                         ((ILESSP D (OPCOUNT 'PVAR_))
                          (AST (LIST 'PVAR_ D)))
                         (T (AST 'PVARX_)
                            (AST (LLSH D 1])
           (FVAR (AST 'FVARX_)
                 (AST (LLSH (IPLUS NLOCALS (fetch FREEVARINDEX
                                                  of (fetch OPARG of A)))
                            1)))
           (OPT.COMPILERERROR)))
(FVAR [COND
          ((ILESSP (SETQ A (IPLUS NLOCALS (fetch FREEVARINDEX of A)))
                   (OPCOUNT 'FVAR))
           (AST (LIST 'FVAR A)))
          (T (AST 'FVARX)
             (AST (LLSH A 1])
(BIND (SETQ A (CDR A))
      (DASSEM.DASMBIND (fetch NVALS of A)
                                                   (fetch NNILS of A)
                                                   (COND
                                                      ((SETQ D (fetch VARS of A))
                                                       (IDIFFERENCE (fetch VARINDEX of (CAR D))
                                                              NARGS))
                                                      (T 1))))
                                     ((UNBIND DUNBIND)
                                          (SETQ A (CDR A))
                                          (COND
                                             ((IGREATERP (fetch NVALS of A)
                                                     15)
                                              (OPT.COMPILERERROR)))           ; if did extra BINDs because of #NILs bound, do extra UNBINDs
                                          (FRPTQ (ADD1 (LRSH (fetch NNILS of A)
                                                             4))
                                                 (AST OP)))
                                     (ATOM (AST X))
                                     (STORE (AST 'STORE.N)
                                            (OR (GREATERP A 0)
                                                (SHOULDNT))
                                            (AST (LLSH (SUB1 A)
                                                       1)))
                                     (*STORE [MAPC A (FUNCTION (LAMBDA (X)
                                                                 (AST X])
                                     (COPY (COND
                                              (A (if (EQ A 0)
                                                     then (HELP))
                                                 (AST 'COPY.N)
                                                 (AST (LLSH A 1)))
                                              (T (AST OP))))
                                     (AST OP))
                            NEXT
                                (SETQ LL (CDR LL))
                                (GO LP)))
                  (DASSEM.DWRITEFN FN FRAMENAME ARGTYPE ARGS LOCALS FREEVARS CD])
```

(DASSEM.DWRITEFN

```
  [LAMBDA (FN FRAMENAME ARGTYPE ARGS LOCALS FREEVARS CD)          (* bvm%: " 2-Oct-86 22:01")
    (PROG ((NARGS (LENGTH ARGS))
           (NLOCALS (LENGTH LOCALS))
           (NFREEVARS (LENGTH FREEVARS))
           [LC (FLENGTH (NCONC1 CD '-X-]
           NAMETABLE LOCALVARINFO)                                  ; had set radix to 8, but not sure why that matters.  Nowadays
                                                                    ; want to write in current reader environment anyway
          [PROGN
                 ;; Construct the name table.  Is a flattened list of entries <code, index, varname>, where code is one of P, I, F.  First come
                 ;; PVAR's, in reverse order of binding, then IVAR's, then FVAR's.  Thus free variable lookup can search the table in order.  We
                 ;; build NAMETABLE backwards, consing onto front

                 [COND
                    (FREEVARS (for X in FREEVARS as I from NLOCALS do (push NAMETABLE (CDR X)
                                                                             I
                                                                             'F))
                                                                    ; Fine, but backwards: the FVARS need to be in order, while the
                                                                    ; PVARS want to be in reverse order
                           (SETQ NAMETABLE (DREVERSE NAMETABLE]
                 [for X in ARGS as I from 0 do (COND
                                                  ((NEQ (CAR X)
                                                        'HVAR)
                                                   (push NAMETABLE 'I I (CDR X)))
                                                  (T                ; Need to save localvar args for ARGLIST
                                                     (push LOCALVARINFO I (CDR X]
                 [for X in LOCALS as I from 0 do (COND
                                                    ((NEQ (CAR X)
                                                          'HVAR)
                                                     (push NAMETABLE 'P I (CDR X)))
                                                    ((AND (EQ ARGTYPE 2)
                                                          (EQ I 0))
                                                     (push LOCALVARINFO I (CDR X]
                 (COND
                    ((AND LOCALVARINFO (DASSEM.SAVELOCALVARS FN))
                                                                    ; Keep this separate, so for now DCODERD can easily discard it
                     (push NAMETABLE 'L LOCALVARINFO]
          (COND
             ((NEQ FRAMENAME FN)
              (push NAMETABLE 'NAME FRAMENAME)))
          (SELECTQ LAPFLG
              ((2 T)
                   (DASSEM.DPRINTLAP FN NAMETABLE ARGTYPE CD))
              NIL)
          [COND
             (LCFIL (LET [(OUTSTREAM (GETSTREAM LCFIL 'OUTPUT]

                         ;; First dump function name and codeindicator to say that what follows is compiled code.  This is in FILERDTBL in the
                         ;; old days, or the current environment nowadays

                         (PRIN4 FN OUTSTREAM)
                         (PRIN3 " " OUTSTREAM)
                         (PRIN4 CODEINDICATOR OUTSTREAM)
```

```
                             (TERPRI OUTSTREAM)
                             (LET ((*READTABLE* (if (EQ *READTABLE* FILERDTBL)
                                                    then                 ; old style file, print code with different read table!
                                                         CODERDTBL
                                                  else                   ; print code in same readtable
                                                       *READTABLE*))
                                   FNFIX ATOMFIX PTRFIX)

                                  ;; Now comes the code in several parts.  Read table is now CODERDTBL old style, or current environment
                                  ;; nowadays.

                                  (PRIN4 NAMETABLE OUTSTREAM)
                                  (PRIN3 " " OUTSTREAM)
                                  (\BOUT OUTSTREAM (LRSH LC 8))
                                  (\BOUT OUTSTREAM (LOGAND LC 255))
                                  (\BOUT OUTSTREAM NLOCALS)
                                  (\BOUT OUTSTREAM NFREEVARS)
                                  (\BOUT OUTSTREAM ARGTYPE)
                                  (\BOUT OUTSTREAM NARGS)

                                  ;; Now the actual code

                                  [for X in CD as LOC from 0
                                     do (\BOUT OUTSTREAM (COND
                                                            [(NLISTP X)
                                                             (COND
                                                                ((AND (FIXP X)
                                                                      (IGEQ X 0)
                                                                      (ILEQ X 255))
                                                                 X)
                                                                (T (fetch OP# of (\FINDOP X T]
                                                            (T         ; something to be fixed up at load time
                                                               (SELECTQ (CAR X)
                                                                   (FN (push FNFIX LOC (CDR X))
                                                                       0)
                                                                   (ATOM (push ATOMFIX LOC (CDR X))
                                                                         0)
                                                                   (PTR (push PTRFIX LOC (CDR X))
                                                                        0)
                                                                   (IPLUS (CAR (fetch OP# of (\FINDOP (CAR X)
                                                                                                       T)))
                                                                          (CADR X]

                                  ;; Now print 3 lists of code fixups.

                                  (PRIN4 FNFIX OUTSTREAM)
                                  (TERPRI OUTSTREAM)
                                  (PRIN4 ATOMFIX OUTSTREAM)
                                  (TERPRI OUTSTREAM)
                                  (PRIN3 "(" OUTSTREAM)
                                  [for X in PTRFIX do (SPACES 1 OUTSTREAM)
                                                      (COND
                                                         ((EQ (CAR X)
                                                              LOADTIMECONSTANTMARKER)
                                                          (if (fetch (READTABLEP COMMONLISP) of *READTABLE*)
                                                              then (PRIN3 "#." OUTSTREAM)
                                                            else (BOUT OUTSTREAM (CHARCODE ^Y)))
                                                          (PRIN4 (CDR X)
                                                                 OUTSTREAM))
                                                         (T (PRIN4 X OUTSTREAM]
                                  (PRIN3 ")" OUTSTREAM)
                                  (TERPRI OUTSTREAM]
               (COND
                  (STRF (DASSEM.DSTOREFNDEF FN CD LC ARGTYPE NARGS NLOCALS NFREEVARS NAMETABLE)))
               (RETURN FN])
```

(DASSEM.SAVELOCALVARS

```
  [LAMBDA (FN)
    T])
```

(DASSEM.DSTOREFNDEF

```
  [LAMBDA (FN CD LC ARGTYPE NARGS NLOCALS NFREEVARS NAMETABLE)     ; Edited  6-Feb-91 17:26 by jds

    ;; Really store the definition of a [byte-compiled] function into a code block.  Builds the name table, and fills in local-function, symbol, and constant
    ;; corrections as well.

    ;; DO NOT RUN THIS CODE INTERPRETED.  It depends for its proper operation on being compiled with XCLC::*TARGET-ARCHITECTURE*  set
    ;; correctly w.r.t. :3-BYTE atoms.

    ;; Much of this code is duplicated in DCODERD (in file LLCODE).  Any changes to the codeblock format or this function's behavior should be
    ;; mirrored there.

    (PROG ((NTSIZE 0)
           (FRAMENAME FN)
           REALSIZE STARTPC NTWORDS CA FVAROFFSET LOCALARGS STARTLOCALS LOCALSIZE)
          [COND
             ((EQ (CAR NAMETABLE)
                  'NAME)
              (SETQ FRAMENAME (CADR NAMETABLE))
              (SETQ NAMETABLE (CDDR NAMETABLE]
          [COND
             ((EQ (CAR NAMETABLE)
                  'L)
              (SETQ LOCALARGS (CADR NAMETABLE))
              (SETQ NAMETABLE (CDDR NAMETABLE]
          [COND
             (NAMETABLE                                          ; NAMETABLE now is a sequence of flat triples, one per name to
                                                                 ; be stored in nametable
                     (on NAMETABLE by CDDDR do (add NTSIZE 1))
                     (SETQ NTSIZE (CEIL (UNFOLD (ADD1 NTSIZE)
                                                (CONSTANT (WORDSPERNAMEENTRY)))
                                        WORDSPERQUAD]
          [SETQ NTWORDS (COND
                           (NAMETABLE (IPLUS NTSIZE NTSIZE))
                           (T (CONSTANT WORDSPERQUAD]

     ;; NameTable must end in quadword which ends in 0 --- thus, round down and add a quad --- NTWORDS is the number of words allocated for
     ;; nametable

          (SETQ STARTPC (UNFOLD (IPLUS (fetch (CODEARRAY OVERHEADWORDS) of T)
                                       NTWORDS)
                               BYTESPERWORD))                    ; initial pc for the function: after fixed header and double
                                                                 ; nametable
          [COND
             (LOCALARGS (SETQ STARTLOCALS STARTPC)               ; Insert an extra nametable between the real one and the start pc
                                                                 ; where we store localvar args
                     (SETQ LOCALSIZE (CEIL [ADD1 (UNFOLD (FOLDLO (FLENGTH LOCALARGS)
                                                                 2)
                                                         (CONSTANT (WORDSPERNAMEENTRY]
                                           (IQUOTIENT WORDSPERQUAD 2)))
                                                                 ; Number of words in half this nametable: must end in zero, when
                                                                 ; doubled is quad-aligned
                     (SETQ LOCALSIZE (UNFOLD LOCALSIZE BYTESPERWORD))
                                                                 ; size in bytes now
                     (add STARTPC (UNFOLD LOCALSIZE 2]
          (SETQ REALSIZE (CEIL (IPLUS STARTPC LC)
                               BYTESPERQUAD))
          (SETQ CA (\CODEARRAY REALSIZE (CEIL (ADD1 (FOLDHI STARTPC BYTESPERCELL))
                                              CELLSPERQUAD)))
          [for X in CD as LOC from STARTPC
             do (COND
                   [(NLISTP X)
                    (CODESETA CA LOC (COND
                                        ((AND (FIXP X)
                                              (IGEQ X 0)
                                              (ILEQ X 255))
                                         X)
                                        (T                       ; assume that this is an opcode which isn't a 'range'
                                           (fetch OP# of (\FINDOP X T]
                   (T (SELECTQ (CAR X)
                          (FN (\FIXCODESYM CA LOC (\ATOMDEFINDEX (CDR X))))
                          (ATOM (\FIXCODESYM CA LOC (\ATOMPNAMEINDEX (CDR X))))
                          (PTR [\FIXCODEPTR CA LOC (COND
                                                      ((EQ (CADR X)
                                                           LOADTIMECONSTANTMARKER)
                                                       (EVAL (CDDR X)))
                                                      (T (CDR X])
                          (PROGN                                 ; assume that this is a 'range' type opcode
                                 (CODESETA CA LOC (IPLUS (CAR (fetch OP# of (\FINDOP (CAR X)
                                                                                    T)))
                                                         (CADR X]

     ;; Now build the name table, which has two parallel parts: the names, and where to find them on the stack

          (for X on NAMETABLE by (CDDDR X) as NT1 from (IPLUS (SUB1 (BYTESPERNAMEENTRY))
                                                            (UNFOLD (fetch (CODEARRAY OVERHEADWORDS)
                                                                           of T)
                                                                   BYTESPERWORD))
             by (BYTESPERNAMEENTRY) bind (NTBYTESIZE _ (UNFOLD NTSIZE BYTESPERWORD))
             do (\FIXCODESYM CA NT1 (CADDR X)
                       -1)                                       ; Insert the name into first half of table
                (SETSTKNTOFFSET CA (IPLUS NT1 NTBYTESIZE)
                       (SELECTQ (CAR X)
                           (P (CONSTANT PVARCODE))
                           (F [OR FVAROFFSET (SETQ FVAROFFSET (UNFOLD (FOLDLO NT1 (CONSTANT (BYTESPERNAMEENTRY)
                                                                                        ))
                                                                     (CONSTANT (WORDSPERNAMEENTRY]
                                                                 ; Save word offset of first FVAR in nametable, so ucode can
                                                                 ; easily access FVAR n
                              (CONSTANT FVARCODE))
                           (I (CONSTANT IVARCODE))
                           (SHOULDNT))
                       (CADR X))                                 ; Code type and index into second half

                )
          [COND
             (LOCALARGS                                          ; Build invisible name table for locals
                     (for X on LOCALARGS by (CDDR X) as NT from (IPLUS (SUB1 (BYTESPERNAMEENTRY))
                                                                       STARTLOCALS)
                     by (CONSTANT (BYTESPERNAMEENTRY)) do (\FIXCODESYM CA NT (\ATOMVALINDEX (CADR X))
                                                                  -1)
                                                              ; Name in first half
                                                          (SETSTKNTOFFSET CA (IPLUS NT LOCALSIZE)
                                                                 (CONSTANT IVARCODE)
                                                                 (CAR X))
                                                              ; index in second half

                                                          ]
          (PROGN                                              ; Fill in function header
                 (replace (CODEARRAY NA) of CA with (COND
                                                     ((EQ ARGTYPE 2)
                                                      -1)
                                                     (T NARGS)))
                 (replace (CODEARRAY PV) of CA with (SUB1 (FOLDHI (IPLUS NLOCALS NFREEVARS)
                                                               CELLSPERQUAD)))
                 (replace (CODEARRAY STARTPC) of CA with STARTPC)
                 (replace (CODEARRAY ARGTYPE) of CA with ARGTYPE)
                 (replace (CODEARRAY FRAMENAME) of CA with FRAMENAME)
                 (replace (CODEARRAY NTSIZE) of CA with NTSIZE)
                 (replace (CODEARRAY NLOCALS) of CA with NLOCALS)
                 (replace (CODEARRAY FVAROFFSET) of CA with (OR FVAROFFSET 0))
                 (replace (CODEARRAY FIXED) of CA with T))
          (RESETVARS [(DFNFLG (COND
                                 (SVFLG NIL)
                                 (T T]
                 (DPUTCODE FN CA (IPLUS STARTPC LC])
```

(DASSEM.DPRINTLAP

```
  [LAMBDA (FN NAMETABLE ARGTYPE CD)                          (* bvm%: " 2-Oct-86 21:57")
    (LET ((OUTSTREAM (GETSTREAM LSTFIL 'OUTPUT))
          (*PRINT-BASE* 8))
         (printout OUTSTREAM .P2 FN T "name table: " T .P2 NAMETABLE T "code length: " " argtype: " ARGTYPE T)
         (MAPRINT CD OUTSTREAM NIL NIL NIL (FUNCTION PRIN2))
         (printout OUTSTREAM T T])
```

(DASSEM.EQCONSTANTP

```
  [LAMBDA (ARG FLG)                                          (* lmm "26-DEC-81 15:52")
    (OR (LITATOM ARG)
        (AND (FIXP ARG)
             (IGEQ ARG -65536)
             (ILEQ ARG 65535])
```

(DASSEM.MATCHVARS

```
  [LAMBDA (VARS TAIL)                                        (* lmm "29-JUL-81 07:03")

    ;; find a match for VARS in TAIL (a tail of LOCALS) --- tack VARS onto end if not possible

    (COND
       [(AND (for VAR in VARS as X in TAIL always (EQUAL VAR X))
             (for VAR in VARS as X in TAIL always (DASSEM.CANSHAREBINDING VAR X)))
                                                             ; variables in VARS can share binding pointers with variables in
                                                             ; TAIL
        (PROG NIL
          LP  (replace VARINDEX of (CAR VARS) with (fetch VARINDEX of (CAR TAIL)))
              (COND
                 ((SETQ VARS (CDR VARS))
                  (COND
                     ((CDR TAIL)
                      (SETQ TAIL (CDR TAIL))
                      (GO LP))
                     (T                                      ; some variables left;  tack onto end
                        (DASSEM.COUNTVARS VARS)
                        (RPLACD TAIL VARS]
       ((CDR TAIL)
        (DASSEM.MATCHVARS VARS (CDR TAIL)))
       (T (DASSEM.COUNTVARS VARS)
          (RPLACD TAIL VARS])
```

(DASSEM.COUNTVARS

```
  [LAMBDA (VARS)                                             (* lmm "26-JAN-80 21:23")

    ;; assign sequential variable numbers to VARS

    (for VAR in VARS do (replace VARINDEX of VAR with (PROG1 VARCOUNT (ADD1VAR VARCOUNT])
```

(DASSEM.CANSHAREBINDING

```
  [LAMBDA (V1 V2)                                            (* lmm "22-DEC-81 22:58")

    ;; can the two variables V1 and V2 share binding pointers?  --- yes, if they are both either (HVAR) or else both (AVAR  . atom) with same atom
    ;; name, and V2's frame (and the frame of any variable which shares a binding pointer with V2) is mutually exclusive from V1's frame (i.e., both
    ;; binds cannot happen at the same time)

    (AND (EQUAL V1 V2)
         (for FR in FRAMES when (AND (find V3 in (fetch (FRAME VARS) of FR)
                                                suchthat (EQ (fetch VARINDEX of V3)
                                                             (fetch VARINDEX of V2)))
                                             (OR (PARENTP FR FRAME)
                                                 (PARENTP FRAME FR)))
            do                                                              ; KILROY wuz here
               (RETURN NIL)
            finally (RETURN T])

)

(DECLARE%: EVAL@COMPILE

(RPAQQ NARGMAX 127)

(RPAQQ NLOCALMAX 127)

(RPAQQ NFREEMAX 127)

(CONSTANTS NARGMAX NLOCALMAX NFREEMAX)
)

(DEFINEQ

(DASSEM.DASMBIND
  [LAMBDA (NV NN K)                                                    (* lmm "13-Jul-84 21:18")
    (COND
       [(IGREATERP NV 15)
        (COMPERROR (CONS NV '(- too many values bound]
       ((IGREATERP NN 15)                                              ; BIND of more than 15 NIL s
        (DASSEM.DASMBIND NV 15 K)
        (DASSEM.DASMBIND 0 (IDIFFERENCE NN 15)
               (IPLUS K NV 15)))
       (T                                                              ; BIND opcode
          (AST 'BIND)
          (AST (IPLUS (LLSH NN 4)
                      NV))
          (AST (SUB1 (IPLUS K NV NN])

(DASSEM.DSTOREFN
  [LAMBDA (X)                                                          ; Edited 17-Nov-92 00:59 by sybalsky:mv:envos

    ;; Write out the extra bytes that go with a function call:  The 2 (or on sun 3) bytes of symbol of the function to be called.

    (AST 0)

    ;; For suns, it's a 4-byte add-on.

    (COND
       ((FMEMB :4-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE *BC-MACRO-ENVIRONMENT*))
        (AST 0)
        (AST 0))
       ((FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE *BC-MACRO-ENVIRONMENT*))
        (AST 0)))
    (AST (CONS 'FN X])

(DASSEM.ASMAJ
  [LAMBDA (P D)                                                        (* lmm " 8-Nov-84 18:46")
    (PROG ((OP (CAAR P))
           Y S)
          (SETQ Y (GETP OP 'MOPVAL))
          (SELECTQ (SETQ S (OPT.JSIZE (CAR P)
                                  D
                                  'AJSIZES))
              (1                                                       ; 1 byte jump --- JUMP FJUMP TJUMP
                 [FRPLACA P (if (ILESSP D 2)
                                then (OR (AND (EQ D 1)
                                              (SELECTQ (CAAR P)
                                                  (JUMP 'NOP)
                                                  ((TJUMP FJUMP)
                                                       'POP)
                                                  NIL))
                                         (COMPILER.ERROR))
                              else (LIST (CAR Y)
                                         (IPLUS D -2])
              (2                                                       ; 2 byte jump -- JUMPX TJUMPX FJUMPX NTJUMPX
                                                                       ; NFJUMPX
                 (FRPLNODE P (CADR Y)
                        (CONS (COND
                                 ((ILESSP D 0)
                                  (COND
                                     ((ILESSP D -128)
                                      (OPT.COMPILERERROR)))
                                  (IPLUS 256 D))
                                 (T (COND
                                       ((IGREATERP D 127)
                                        (OPT.COMPILERERROR)))
                                    D))
```

```
                          (CDR P))))
            ((3 4)                                              ; 3 byte jump is JUMPXX.  4 byte jump is FJUMP.+4 JUMPXX to
                                                                ; implement TJUMPXX
               [COND
                  ((EQ S 3)
                   (OR (EQ (CADDR Y)
                           'JUMPXX)
                       (OPT.COMPILERERROR))
                   (FRPLACA P (CADDR Y)))
                  (T                                            ; long t/f jump implemented by short jump followed by JUMPXX
                     (add D -1)
                     (FRPLNODE P (CADDR Y)
                            (SETQ P (CONS 'JUMPXX (CDR P]
               [FRPLACD P (CONS (LOGAND (RSH D 8)
                                        255)
                                (CONS (LOGAND D 255)
                                      (CDR P])
            (6                                                  ; long NXJUMP implemented by NXJUMP.+2 JUMP.+4
                                                                ; JUMPXX.place IN 6 BYTES
              [FRPLNODE P (CADR Y)
                     (CONS 3 (CONS '(JUMP 2)
                                   (CONS 'JUMPXX (CONS (LRSH (SETQ D (LOGAND (IPLUS D -3)
                                                                             65535))
                                                             8)
                                                       (CONS (LOGAND D 255)
                                                             (CDR P])
            (OPT.COMPILERERROR])
)
```

(RPAQQ **EMFLAG** NIL)

(RPAQQ **COMPILEMODE** D)

(PUTPROPS **JUMP MOPVAL** (JUMP JUMPX JUMPXX))

(PUTPROPS **FJUMP MOPVAL** (FJUMP FJUMPX (TJUMP 2)))

(PUTPROPS **TJUMP MOPVAL** (TJUMP TJUMPX (FJUMP 2)))

(PUTPROPS **NTJUMP MOPVAL** (NIL NTJUMPX))

(PUTPROPS **NFJUMP MOPVAL** (NIL NFJUMPX))

```
(PUTPROPS JUMP AJSIZES ((1 . 3)
                        18
                        (1 (-127 3 . 2) . 1)
                        127 2 . 3))

(PUTPROPS FJUMP AJSIZES ((1 . 4)
                         18
                         (1 (-127 4 . 2) . 1)
                         127 2 . 4))

(PUTPROPS TJUMP AJSIZES ((1 . 4)
                         18
                         (1 (-127 4 . 2) . 1)
                         127 2 . 4))

(PUTPROPS NTJUMP AJSIZES ((2 . 6)
                          18
                          (2 (-127 6 . 2) . 2)
                          127 2 . 6))

(PUTPROPS NFJUMP AJSIZES ((2 . 6)
                          18
                          (2 (-127 6 . 2) . 2)
                          127 2 . 6))

(RPAQQ DOPVALS
       (.APPLYFN. ARRAYP ASSOC BIN CAR CDR CONS CREATECELL DIFFERENCE EQ EQL EQUAL FDIFFERENCE FGREATERP FIX
              FIXP FLESSP FLOAT FLOATP FMEMB FPLUS FQUOTIENT FTIMES GREATERP IDIFFERENCE IGREATERP ILESSP IPLUS
              IQUOTIENT IREMAINDER ITIMES LESSP LISTGET LISTP LLSH1 LLSH8 LOGAND LOGOR LOGXOR LRSH1 LRSH8 LSH
              NTYPX NULL NUMBERP PLUS QUOTIENT READPRINTERPORT RPLACA RPLACD SMALLP STACKP TIMES
              WRITEPRINTERPORT \ADDBASE \ARG0 \BIN \BLKEXPONENT \BLKFDIFF \BLKFLOATP2COMP \BLKFPLUS \BLKFTIMES
              \BLKMAG \BLKPERM \BLKSEP \BLKSMALLP2FLOAT \BLT \BOXIDIFFERENCE \BOXIPLUS \CONTEXTSWITCH
              \DRAWLINE.UFN \EVAL \FLOAT.BOX \FLOATBOX \FLOATUNBOX \GCRECLAIMCELL \GCSCAN1 \GCSCAN2 \IBLT1
              \IBLT2 \MAKENUMBER \MTIMES3 \MTIMES4 \MYALINK \MYARGCOUNT \PILOTBITBLT \PIXELBLT \RCLK \READFLAGS
              \READRP \RPLCONS \STKSCAN \WRITEMAP \\ADDBASE))
```

(PUTPROPS **.APPLYFN. DOPVAL** ((NIL APPLYFN)))

(PUTPROPS **ARRAYP DOPVAL** (1 TYPEP 6))

(PUTPROPS **ASSOC DOPVAL** (2 ASSOC))

(PUTPROPS **BIN DOPVAL** (1 BIN))

```
(PUTPROPS CAR DOPVAL (1 CAR))

(PUTPROPS CDR DOPVAL (1 CDR))

(PUTPROPS CONS DOPVAL (2 CONS))

(PUTPROPS CREATECELL DOPVAL (1 CREATECELL))

(PUTPROPS DIFFERENCE DOPVAL (2 DIFFERENCE))

(PUTPROPS EQ DOPVAL (2 EQ))

(PUTPROPS EQL DOPVAL (2 EQL))

(PUTPROPS EQUAL DOPVAL (2 EQUAL))

(PUTPROPS FDIFFERENCE DOPVAL (2 FDIFFERENCE))

(PUTPROPS FGREATERP DOPVAL (2 FGREATERP))

(PUTPROPS FIX DOPVAL (1 %'0 IPLUS2))

(PUTPROPS FIXP DOPVAL (1 TYPEMASK.N 32))

(PUTPROPS FLESSP DOPVAL (2 SWAP FGREATERP))

(PUTPROPS FLOAT DOPVAL ((1 DTEST 0 0 0 (ATOM . FLOATP))))

(PUTPROPS FLOATP DOPVAL (1 TYPEP 3))

(PUTPROPS FMEMB DOPVAL (2 FMEMB))

(PUTPROPS FPLUS DOPVAL ((2 FPLUS2)))

(PUTPROPS FQUOTIENT DOPVAL (2 FQUOTIENT))

(PUTPROPS FTIMES DOPVAL ((2 FTIMES2)))

(PUTPROPS GREATERP DOPVAL (2 GREATERP))

(PUTPROPS IDIFFERENCE DOPVAL (2 IDIFFERENCE))

(PUTPROPS IGREATERP DOPVAL (2 IGREATERP))

(PUTPROPS ILESSP DOPVAL (2 SWAP IGREATERP))

(PUTPROPS IPLUS DOPVAL ((0 . OPT.COMPILERERROR)
                        (1 %'0 IPLUS2)
                        (2 IPLUS2) . OPT.COMPILERERROR))

(PUTPROPS IQUOTIENT DOPVAL (2 IQUOTIENT))

(PUTPROPS IREMAINDER DOPVAL (2 IREMAINDER))

(PUTPROPS ITIMES DOPVAL ((0 . OPT.COMPILERERROR)
                         (1 0 IPLUS2)
                         (2 ITIMES2) . OPT.COMPILERERROR))

(PUTPROPS LESSP DOPVAL (2 SWAP GREATERP))

(PUTPROPS LISTGET DOPVAL (2 LISTGET))

(PUTPROPS LISTP DOPVAL (1 LISTP))

(PUTPROPS LLSH1 DOPVAL (1 LLSH1))

(PUTPROPS LLSH8 DOPVAL (1 LLSH8))

(PUTPROPS LOGAND DOPVAL ((2 LOGAND2)))

(PUTPROPS LOGOR DOPVAL ((2 LOGOR2)))

(PUTPROPS LOGXOR DOPVAL ((2 LOGXOR2)))

(PUTPROPS LRSH1 DOPVAL (1 LRSH1))

(PUTPROPS LRSH8 DOPVAL (1 LRSH8))

(PUTPROPS LSH DOPVAL (2 LSH))

(PUTPROPS NTYPX DOPVAL (1 NTYPX))

(PUTPROPS NULL DOPVAL (1 %'NIL EQ))

(PUTPROPS NUMBERP DOPVAL (1 TYPEMASK.N 16))
```

```
(PUTPROPS PLUS DOPVAL ((1 %'0 PLUS2)
                       (2 PLUS2) . OPT.COMPILERERROR))

(PUTPROPS QUOTIENT DOPVAL (2 QUOTIENT))

(PUTPROPS READPRINTERPORT DOPVAL (0 READPRINTERPORT))

(PUTPROPS RPLACA DOPVAL (2 RPLACA))

(PUTPROPS RPLACD DOPVAL (2 RPLACD))

(PUTPROPS SMALLP DOPVAL (1 TYPEP 1))

(PUTPROPS STACKP DOPVAL (1 TYPEP 8))

(PUTPROPS TIMES DOPVAL ((2 TIMES2)))

(PUTPROPS WRITEPRINTERPORT DOPVAL (1 WRITEPRINTERPORT))

(PUTPROPS \ADDBASE DOPVAL (2 ADDBASE))

(PUTPROPS \ARG0 DOPVAL (1 ARG0))

(PUTPROPS \BIN DOPVAL (1 BIN))

(PUTPROPS \BLKEXPONENT DOPVAL (3 MISC3 0))

(PUTPROPS \BLKFDIFF DOPVAL (4 MISC4 3))

(PUTPROPS \BLKFLOATP2COMP DOPVAL (3 MISC3 3))

(PUTPROPS \BLKFPLUS DOPVAL (4 MISC4 2))

(PUTPROPS \BLKFTIMES DOPVAL (4 MISC4 0))

(PUTPROPS \BLKMAG DOPVAL (3 MISC3 1))

(PUTPROPS \BLKPERM DOPVAL (4 MISC4 1))

(PUTPROPS \BLKSEP DOPVAL (4 MISC4 4))

(PUTPROPS \BLKSMALLP2FLOAT DOPVAL (3 MISC3 2))

(PUTPROPS \BLT DOPVAL (3 BLT))

(PUTPROPS \BOXIDIFFERENCE DOPVAL (2 BOXIDIFFERENCE))

(PUTPROPS \BOXIPLUS DOPVAL (2 BOXIPLUS))

(PUTPROPS \CONTEXTSWITCH DOPVAL (1 CONTEXTSWITCH))

(PUTPROPS \DRAWLINE.UFN DOPVAL (9 DRAWLINE))

(PUTPROPS \EVAL DOPVAL (1 EVAL))

(PUTPROPS \FLOAT.BOX DOPVAL (1 GCONST 0 0 0 (PTR . 0.0)
                                FPLUS2))

(PUTPROPS \FLOATBOX DOPVAL (1 UBFLOAT1 0))

(PUTPROPS \FLOATUNBOX DOPVAL (1 UBFLOAT1 1))

(PUTPROPS \GCRECLAIMCELL DOPVAL (1 RECLAIMCELL))

(PUTPROPS \GCSCAN1 DOPVAL (1 GCSCAN1))

(PUTPROPS \GCSCAN2 DOPVAL (1 GCSCAN2))

(PUTPROPS \IBLT1 DOPVAL (8 MISC8 0))

(PUTPROPS \IBLT2 DOPVAL (8 MISC8 1))

(PUTPROPS \MAKENUMBER DOPVAL (2 MAKENUMBER))

(PUTPROPS \MTIMES3 DOPVAL (3 UBFLOAT3 1))

(PUTPROPS \MTIMES4 DOPVAL (3 UBFLOAT3 2))

(PUTPROPS \MYALINK DOPVAL (1 MYALINK))

(PUTPROPS \MYARGCOUNT DOPVAL (0 MYARGCOUNT))

(PUTPROPS \PILOTBITBLT DOPVAL (2 PILOTBITBLT))

(PUTPROPS \PIXELBLT DOPVAL (10 MISC10 0))

(PUTPROPS \RCLK DOPVAL (1 RCLK))
```

```
(PUTPROPS \READFLAGS DOPVAL (1 READFLAGS))

(PUTPROPS \READRP DOPVAL (1 READRP))

(PUTPROPS \RPLCONS DOPVAL (2 RPLCONS))

(PUTPROPS \STKSCAN DOPVAL (1 STKSCAN))

(PUTPROPS \WRITEMAP DOPVAL (3 WRITEMAP))

(PUTPROPS \\ADDBASE DOPVAL (2 ADDBASE))

(RPAQQ CONSTOPS ((NIL . %'NIL)
                 (T . %'T)
                 (0 . %'0)
                 (1 . %'1)))

(RPAQQ COMPILE.ARG.FAST.FLG NIL)

(RPAQQ IPLUSNFLG NIL)

(ADDTOVAR 8BITEXTS DCOM)

(ADDTOVAR MACROPROPS DMACRO ALTOMACRO BYTEMACRO MACRO)

(ADDTOVAR COMPILERMACROPROPS DMACRO ALTOMACRO BYTEMACRO MACRO)

(RPAQQ BYTEASSEMFN DASSEM.DASSEM)

(RPAQQ MAXBVALS 15)

(RPAQQ BYTECOMPFLG T)

(RPAQQ SELECTQFMEMB NIL)

(RPAQQ LAMBDANOBIND T)

(RPAQQ SELECTVARTYPES (AVAR HVAR))

(RPAQQ CONST.FNS
       [(NIL (1 CAR (CONST))
             (1 CDR (CONST))
             (1 NULL (CONST . T))
             (2 EQ (FN 1 . NULL)))
        (0 (2 ITIMES2 (POP)
              (CONST . 0))
           (2 LOGAND2 (POP)
              (CONST . 0))
           (2 IPLUS (FN 1 . FIX))
           (2 LOGOR2 (FN 1 . FIX))
           (2 \ADDBASE))
        (1 (2 ITIMES2 (FN 1 . FIX])

(RPAQQ MERGEFRAMEFLG T)

(RPAQQ MERGEFRAMEMAX 2)

(RPAQQ CLEANFNLIST (NTYPX EQ AND OR CONS LIST FMEMB MEMB GETP SUB1 ADD1 ZEROP ELT ILESSP LLSH LRSH IPLUS
                          IDIFFERENCE \ARG0 GETHASH \ADDBASE \GETBASEPTR \GETBASEBYTE \GETBASE \GETBASEFIXP
                          \GETBASESTRING \VAG2 \ADDBASE))

(RPAQQ OPCODEPROP DOPVAL)

(RPAQQ VCONDITIONALS (ARRAYP FIXP FLOATP LISTP SMALLP STACKP NUMBERP))

(RPAQQ CONDITIONALS (EQ IGREATERP NULL GREATERP LESSP ILESSP))

(RPAQQ CONSTFNS (IPLUS SUB1 ADD1 ZEROP LLSH LRSH IDIFFERENCE))

(RPAQQ MAXARGS 80)

(RPAQQ XVARFLG NIL)

(RPAQQ NOFREEVARFNS (RPLACA RPLACD PUTHASH SETA))

(RPAQQ CLEANFNTEST DASSEM.CLEANFNTEST)

(RPAQQ EQCONSTFN DASSEM.EQCONSTANTP)

(ADDTOVAR NUMBERFNS LLSH1 LRSH1 LLSH8 LRSH8)

(DECLARE%: EVAL@COMPILE

(RPAQQ SHALLOWFLG NIL)

(RPAQQ SPAGHETTIFLG T)
```

```
(CONSTANTS (SHALLOWFLG NIL)
       (SPAGHETTIFLG T))
)

(DEFINEQ

(DASSEM.CLEANFNTEST
  [LAMBDA (FN TYPE)                                                  (* lmm "23-May-86 16:27")
    (DECLARE (GLOBALVARS CONDITIONALS VCONDITIONALS NUMBERFNS CLEANFNLIST NOFREEVARFNS NOSIDEFNS))
    (COND
       ((LITATOM FN)
        (OR (GETPROP FN 'CROPS)
            (FMEMB FN CONDITIONALS)
            (FMEMB FN VCONDITIONALS)
            (FMEMB FN NUMBERFNS)
            (FMEMB FN CLEANFNLIST)
            (SELECTQ TYPE
                 (FREEVARS (FMEMB FN NOFREEVARFNS))
                 (NOSIDE (FMEMB FN NOSIDEFNS))
                 NIL)))
       ((EQ (CAR FN)
            'OPCODES)
        (while (SETQ FN (CDR FN)) do [SELECTQ (CAR FN)
                                         ((GETBASEPTR.N GETBASE.N)
                                              (SETQ FN (CDR FN)))
                                         (GETBITS.N.FD (SETQ FN (CDDR FN)))
                                         (ARG0)
                                         (GCONST (SETQ FN (CDDDR FN)))
                                         (COND
                                            ((LISTP (CAR FN))
                                             (SELECTQ (CAAR FN)
                                                  (IVAR)
                                                  (RETURN)))
                                            (T (RETURN]
           finally (RETURN T])

)


(DEFOPTIMIZER ATOM (&REST ARGS)
                   (CONS '(OPENLAMBDA (X)
                            (OR (NULL X)
                                (AND (\TYPEMASK.UFN X 8)
                                     T)))
                         ARGS))


(DEFOPTIMIZER EVALV (&REST X)
                    (COND
                       ((CADR X)
                        'IGNOREMACRO)
                       (T (CONS '\EVALV1 X))))


(DEFOPTIMIZER FRPLACA (&REST ARGS)
                      (CONS 'RPLACA ARGS))


(DEFOPTIMIZER GETATOMVAL (ATM)
                         `(GETTOPVAL ,ATM))


(DEFOPTIMIZER LIST (&REST X)
                   [AND X (LIST 'CONS (CAR X)
                                (CONS 'LIST (CDR X])


(DEFOPTIMIZER LITATOM (X &ENVIRONMENT ENV)
                      ;; Optimizer for LITATOM predicate.  For 3-byte atom world, needs to include a check for 3-byte-atoms.  For old atoms,
                      ;; it's just a type check.  For new ones, use the type mask.
                      [COND
                         [(FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
                          `((OPCODES COPY TYPEMASK.N ,(CONSTANT (LRSH \TT.SYMBOLP 8))
                                   EQ)
                            ,X]
                         (T `(EQ (NTYPX ,X)
                                 (CONSTANT \LITATOM])


(DEFOPTIMIZER MINUSP (X)
                     `(GREATERP 0 ,X))


(DEFOPTIMIZER IEQP (X Y)
                   `(EQ 0 (IDIFFERENCE ,X ,Y)))
```

```
(DEFOPTIMIZER FASSOC (&REST ARGS)
                     (CONS 'ASSOC ARGS))

(DEFOPTIMIZER SETATOMVAL (ATM VAL)
                         `(SETTOPVAL ,ATM ,VAL))

(DEFOPTIMIZER SYSTEMTYPE (&CTXT IGNORE)
                         ''D)

(DEFOPTIMIZER SPREADAPPLY* (FN &REST ARGS)
                           `(CL:FUNCALL ,FN ,@ARGS))

(PUTPROPS FGETD DMACRO COMP.GETD)

(PUTPROPS FGREATERP DMACRO (APPLY* COMP.COMPARENUM FLOAT FGREATERP NIL (OPCODES UBFLOAT2 5)))

(PUTPROPS FLESSP DMACRO (APPLY* COMP.COMPARENUM FLOAT FLESSP FGREATERP (OPCODES SWAP UBFLOAT2 5)))

(PUTPROPS FMEMB DMACRO COMP.FMEMB)

(PUTPROPS FRPLACD DMACRO COMP.RPLACD)

(PUTPROPS GETD DMACRO COMP.GETD)

(PUTPROPS GREATERP DMACRO (APPLY* COMP.COMPARENUM PLUS GREATERP))

(PUTPROPS IGREATERP DMACRO (APPLY* COMP.COMPARENUM FIX IGREATERP))

(PUTPROPS ILESSP DMACRO (APPLY* COMP.COMPARENUM FIX ILESSP IGREATERP))

(PUTPROPS LESSP DMACRO (APPLY* COMP.COMPARENUM PLUS LESSP GREATERP))

(PUTPROPS LLSH DMACRO COMP.SHIFT)

(PUTPROPS LRSH DMACRO COMP.SHIFT)

(PUTPROPS PRINTNUM DMACRO T)

(PUTPROPS RPLACD DMACRO COMP.RPLACD)

(PUTPROPS \FLOATBOX DMACRO C.FLOATBOX)

(PUTPROPS \FLOATUNBOX DMACRO C.FLOATUNBOX)

(DEFINEQ

(COMP.RPLACD
  [LAMBDA (A)                                                ; Edited  2-Mar-88 14:08 by amd
    (PROG (NEED-POP)
          (COMP.EXPR (CAR A))
          [COND
             ((OPT.CALLP (CAR CODE)
                     'CONS)                                  ; (RPLACD (CONS --) --) -> (CONS & &)
              (FRPTQ (PROG1 (CAR (fetch OPARG of (CAR CODE)))
                            (COMP.DELFN)
                            (COMP.STCONST))
                     (SELECTQ (fetch OPNAME of (CAR CODE))
                          ((CONST AVAR FVAR HVAR GVAR)
                               (COMP.DELPUSH))
                          (COMP.STPOP)))
              (COMP.VAL1 (CDR A))
              (RETURN (COMP.STFN 'CONS (COND
                                          ((EQ (CAR CODE)
                                               OPNIL)
                                           (COMP.DELPUSH)
                                           1)
                                          (T 2]
      DOIT
          (OR (EQ COMPILE.CONTEXT 'EFFECT)
              (COMP.STCOPY))
          (COMP.VAL1 (CDR A))
          (COND
             ([AND (EQ (fetch OPNAME of (CAR CODE))
                       'SETQ)
                   (OR (OPT.CALLP (CADR CODE)
                              'CONS 1)
                       (SETQ NEED-POP (AND (OPT.CALLP (CADR CODE)
                                                  'CONS 2)
                                           (EQ (CADDR CODE)
                                               OPNIL]
              (COMP.ST (PROG1 (pop CODE)
                              (COMP.DELFN)
                           (AND NEED-POP (COMP.DELPUSH))
                           (COMP.STFN '\RPLCONS 2))
                     0))
             ([OR (OPT.CALLP (CAR CODE)
                         'CONS 1)
                  (SETQ NEED-POP (AND (OPT.CALLP (CAR CODE)
                                             'CONS 2)
                                      (EQ (CADR CODE)
                                          OPNIL]
              (COMP.DELFN)
              (AND NEED-POP (COMP.DELPUSH))
              (COMP.STFN '\RPLCONS 2))
             (T (COMP.STFN 'RPLACD 2)))
          (COMP.STPOP)
          (RETURN 'NOVALUE])
```

(**COMP.SHIFT**

```
  [LAMBDA (A)                                                     (* Pavel " 3-Nov-86 18:13")
    (COMP.VAL (CAR A))
    (COMP.DELFIX)
    (COMP.VAL (CADR A))
    (COMP.DELFIX)
    (COND
       [(EQ (fetch OPNAME of (CAR CODE))
            'CONST)                                               ; A compile shift open
        (PROG ((N (fetch OPARG of (CAR CODE)))
               FNS)
              (OR (FIXP N)
                  (COMPERROR (LIST N "non-numeric arg to shift")))
              (COMP.DELPUSH)
              [COND
                 ((EQ (fetch OPNAME of (CAR CODE))
                      'CONST)
                  (RETURN (COMP.STCONST (PROG1 (APPLY* (CAR EXP)
                                                      (fetch OPARG of (CAR CODE))
                                                      N)
                                               (COMP.DELPUSH]
              [SETQ FNS (SELECTQ [COND
                                    ((EQ 0 N)
                                     (RETURN))
                                    ((IGREATERP N 0)
                                     (CAR EXP))
                                    (T (SETQ N (IMINUS N))
                                       (SELECTQ (CAR EXP)
                                            (LLSH 'LRSH)
                                            'LLSH]
                            (LLSH '(LLSH8 . LLSH1))
                            '(LRSH8 . LRSH1]
         LP8 (COND
                ((IGREATERP N 7)
                 (COMP.STFN (CAR FNS)
                        1)
                 (SETQ N (IDIFFERENCE N 8))
                 (GO LP8)))
         LP1 (COND
                ((IGREATERP N 0)
                 (COMP.STFN (CDR FNS)
                        1)
                 (SETQ N (SUB1 N))
                 (GO LP1]
       (T                                                         ; A can't compile shift open
          (COMP.STFN (CAR EXP)
                 2])
```

(**COMP.COMPARENUM**

```
  [LAMBDA (A TYPE FN OFN)                                         (* lmm "24-Jan-85 19:20")
    (PROG (V1)
          (if (EQ COMPILE.CONTEXT 'EFFECT)
              then (RETURN (COMP.PROGN A)))
          [COND
             (OFN (COND
                     ((SETQ V1 (CONSTANTEXPRESSIONP (CADR A)))
                      (RETURN (COMP.COMPARENUM (LIST (CAR V1)
                                                     (CAR A))
                                     TYPE OFN]
          (COMP.EXPR (CAR A)
                 TYPE)
          (COMP.DELFIX TYPE)
          [COND
             ((AND OFN (SELECTQ (fetch OPNAME of (CAR CODE))
                           (CONST [SETQ V1 (KWOTE (fetch OPARG of (CAR CODE])
                           ((AVAR HVAR GVAR FVAR)
                                (SETQ V1 CODE)
                                NIL)
                           NIL))
              (RETURN (PROGN (COMP.DELPUSH)
                             (COMP.VAL1 (CDR A))
                             (COMP.DELFIX TYPE)
                             (COMP.VAL V1)
                             (COMP.STFN OFN 2]
          (COMP.VAL1 (CDR A)
                 TYPE)
          (COMP.DELFIX TYPE)
          (COND
             ((AND OFN V1 (FMEMB (fetch OPNAME of (CAR CODE))
                                 '(CONST AVAR HVAR FVAR GVAR))
                   (EQ (CDR CODE)
                       V1))
              (swap (CAR CODE)
                    (CAR V1))
              (COMP.STFN OFN 2))
             (T (COMP.STFN FN 2])

(COMP.GETD
  [LAMBDA (A)                                                (* Pavel " 3-Nov-86 18:13")
    (COMP.VAL1 A)
    (COND
       ((EQ COMPILE.CONTEXT 'EFFECT)
        (COMP.STPOP)
        'NOVALUE)
       ((COMP.PREDP COMPILE.CONTEXT)
        (COMP.STFN (SELECTQ (CAR COMPILE.CONTEXT)
                       ((TJUMP FJUMP NFJUMP)                 ; \DEFINEDP is the same as GETD when the value is used only
                                                             ; for NIL or T
                            '\DEFINEDP)
                       (CAR EXP))
              1))
       (T (COMP.STFN (CAR EXP)
                 1])

(COMP.FMEMB
  [LAMBDA (A)                                                (* Pavel " 3-Nov-86 18:13")
    (PROG NIL
          [COND
             ((EQ COMPILE.CONTEXT 'EFFECT)
              (RETURN (COMP.VALN A COMPILE.CONTEXT]
          (COMP.EXPR (pop A))
          (COMP.VAL1 A)
          [COND
             ([AND (COMP.PREDP COMPILE.CONTEXT)
                   (EQ (fetch OPNAME of (CAR CODE))
                       'CONST)
                   (FMEMB (fetch OPNAME of COMPILE.CONTEXT)
                          '(FJUMP TJUMP NFJUMP]
              (RETURN (COMP.SELECTQ (LIST DONOTHING (LIST (PROG1 (fetch OPARG of (CAR CODE))
                                                                 (COMP.DELPUSH))
                                                          T)
                                          NIL]
          (RETURN (COMP.STFN (CAR EXP)
                         2])
)

(PUTPROPS DMACRO PROPTYPE MACROS)
```

;; COMP.GETBASE

```
(DEFOPTIMIZER \GETBASEBYTE (X N)
                           `((OPCODES GETBASEBYTE)
                             ,X
                             ,N))

(DEFOPTIMIZER \PUTBASEBYTE (X N V)
                           `((OPCODES PUTBASEBYTE)
                             ,X
                             ,N
                             ,V))

(DEFOPTIMIZER \HILOC (X)
                     `((OPCODES HILOC)
                       ,X))

(DEFOPTIMIZER \LOLOC (X)
                     `((OPCODES LOLOC)
                       ,X))
```

```
(DEFOPTIMIZER \VAG2 (X Y)
                    `((OPCODES VAG2)
                      ,X
                      ,Y))

(PUTPROPS \GETBASE DMACRO (APPLY* COMP.GETBASE NIL GETBASE.N))

(PUTPROPS \GETBASEPTR DMACRO (APPLY* COMP.GETBASE NIL GETBASEPTR.N))

(PUTPROPS \PUTBASE DMACRO (APPLY* COMP.GETBASE T PUTBASE.N))

(PUTPROPS \PUTBASEPTR DMACRO (APPLY* COMP.GETBASE T PUTBASEPTR.N))

(PUTPROPS \RPLPTR DMACRO (APPLY* COMP.GETBASE T RPLPTR.N))

(PUTPROPS \GETBITS DMACRO (APPLY* COMP.GETBASEBITS))

(PUTPROPS \PUTBITS DMACRO (APPLY* COMP.GETBASEBITS T))

(DEFINEQ

(COMP.GETBASE
  [LAMBDA (A STFLG OPCODE)                                   (* Pavel " 3-Nov-86 18:13")
    (COND
       ([AND STFLG (NOT (EQ COMPILE.CONTEXT 'EFFECT]
        (COMP.VAL (CONS (LIST 'OPENLAMBDA '(X N V)
                              (CONS (CAR EXP)
                                    '(X N V))
                              'V)
                        A)))
       ((AND (NOT STFLG)
             (EQ COMPILE.CONTEXT 'EFFECT))
        (COMP.VALN A 'EFFECT))
       (T (PROG ((OFF 0))
                (COMP.VAL (pop A))
                (COND
                   ((AND (OPT.CALLP (CAR CODE)
                                    '\ADDBASE 2)
                         (EQ (fetch OPNAME of (CADR CODE))
                             'CONST))
                    (COMP.DELFN)
                    (add OFF (fetch OPARG of (CAR CODE)))
                    (COMP.DELPUSH)))
                (COMP.EXPR (pop A)
                       '(TYPE . FIX))
                (COND
                   ([AND (EQ (fetch OPNAME of (CAR CODE))
                             'CONST)
                         (FIXP (fetch OPARG of (CAR CODE]
                    (add OFF (fetch OPARG of (CAR CODE)))
                    (COMP.DELPUSH))
                   (T (COMP.STFN '\ADDBASE 2)))
                (COND
                   ((OR (ILESSP OFF 0)
                        (IGREATERP OFF 255))
                    (COMP.STCONST OFF)
                    (COMP.STFN '\ADDBASE 2)
                    (SETQ OFF 0)))
                [COND
                   (STFLG (COMP.EXPR (pop A)
                                 '(TYPE . FIX]
                (MAPC A (FUNCTION COMP.EFFECT))
                (RETURN (if (EQ OPCODE 'GETBASE.32)
                            then (if STFLG
                                     then (HELP)
                                   else (COMP.STFN `(OPCODES COPY GETBASE.N %, OFF SWAP GETBASE.N %, (ADD1 OFF)
                                                           VAG2)
                                               1))
                          else (COMP.STFN (LIST 'OPCODES OPCODE OFF)
                                      (COND
                                         (STFLG 2)
                                         (T 1])

(COMP.GETBASEBITS
  [LAMBDA (A STFLG)                                          (* Pavel " 3-Nov-86 18:13")
    (COND
       [[AND STFLG (NOT (EQ COMPILE.CONTEXT 'EFFECT]
        (COMP.VAL (LIST (LIST 'OPENLAMBDA '(X V)
                              (LIST (CAR EXP)
                                    'X
                                    (CADR A)
                                    (CADDR A)
                                    'V)
                              'V)
                        (CAR A)
                        (CADDDR A]
        (T (PROG ((OFF (CADR A)))
                 (COMP.VAL (CAR A))
                 (COND
                    ((AND (OPT.CALLP (CAR CODE)
                                '\ADDBASE 2)
                          (EQ (fetch OPNAME of (CADR CODE))
                              'CONST))
                     (COMP.DELFN)
                     (add OFF (fetch OPARG of (CAR CODE)))
                     (COMP.DELPUSH)))
                 (COND
                    ((OR (ILESSP OFF 0)
                         (IGREATERP OFF 255))
                     (COMP.STCONST OFF)
                     (COMP.STFN '\ADDBASE 2)
                     (SETQ OFF 0)))
                 [COND
                    (STFLG (COMP.VAL (CADDDR A]
                 (RETURN (COMP.STFN [CONS 'OPCODES (COND
                                                      [(EQ (CADDR A)
                                                           15)
                                                       (COND
                                                          (STFLG (LIST 'PUTBASE.N OFF))
                                                          (T (LIST 'GETBASE.N OFF]
                                                      (T (COND
                                                            (STFLG (LIST 'PUTBITS.N.FD OFF (CADDR A)))
                                                            (T (LIST 'GETBITS.N.FD OFF (CADDR A]
                                   (COND
                                      (STFLG 2)
                                      (T 1])

)

(DEFINEQ

(COMP.SPREADFN
  [LAMBDA (2FN ARGS)                                          (* lmm "15-APR-82 22:26")
    (COND
       ((NULL (CDR ARGS))
        (CAR ARGS))
       ((NULL (CDDR ARGS))
        (CONS 2FN ARGS))
       (T (LIST 2FN (CAR ARGS)
                (COMP.SPREADFN 2FN (CDR ARGS])

)


(DEFOPTIMIZER NCONC (&REST ARGS)
                        [COND
                           ((NULL (CDR ARGS))
                            (CAR ARGS))
                           ((NULL (CDDR ARGS))
                            (CONS '\NCONC2 ARGS))
                           (T (LIST '\NCONC2 (CAR ARGS)
                                    (CONS 'NCONC (CDR ARGS])


(DEFOPTIMIZER APPEND (&REST ARGS)
                         [COND
                            ((NULL (CDR ARGS))
                             (LIST '\APPEND2 (CAR ARGS)
                                   NIL))
                            ((NULL (CDDR ARGS))
                             (CONS '\APPEND2 ARGS))
                            (T (LIST '\APPEND2 (CAR ARGS)
                                     (CONS 'APPEND (CDR ARGS])

;; CAPPLYFN

(PUTPROPS NILAPPLY DMACRO (OPENLAMBDA (FN N)
                            (.PUSHNILS. N FN)))

(PUTPROPS .PUSHNILS. DMACRO (APPLY COMP.PUSHNILS))

(PUTPROPS SPREADAPPLY DMACRO [OPENLAMBDA (FN ARGLIST)
                               (PROG ((CNT 0))
                                     (DECLARE (LOCALVARS . T))
                                     (RETURN (.SPREAD. ARGLIST CNT FN])

(PUTPROPS .SPREAD. DMACRO (APPLY COMP.SPREAD))

(PUTPROPS .EVALFORM. DMACRO COMP.EVALFORM)

(PUTPROPS .CALLAFTERPUSHINGNILS. DMACRO (APPLY COMP.PUSHCALL))
```

```
(PUTPROPS APPLY* DMACRO COMP.APPLY*)

(PUTPROPS .SPREADCONS. DOPVAL (1 COPY CAR SWAP CDR))

(PUTPROPS .SWAPNIL. DOPVAL (2 SWAP))

(DEFINEQ

(COMP.PUSHNILS
  [LAMBDA (N FN)                                                   (* lmm "16-APR-82 00:39")
    (COMP.EXPR N)
    (PROG ((CHK (create TAG))
           (LP (create TAG))
           (LEV LEVEL)
           (FR FRAME))
          (COMP.STJUMP 'JUMP CHK)
          (SETQ LEVEL LEV)
          (SETQ FRAME FR)
          (COMP.STTAG LP)
          (COMP.STCONST)
          (COMP.STFN '.SWAPNIL. 2)
          (COMP.STCONST 1)
          (COMP.STFN 'IDIFFERENCE 2)
          (COMP.STTAG CHK)
          (COMP.STCOPY)
          (COMP.STCONST 0)
          (COMP.STFN 'IGREATERP 2)
          (COMP.STJUMP 'TJUMP LP)
          (COMP.STPOP)
          (COMP.VAL N)
          (COMP.VAL FN)
          (COMP.STFN '.APPLYFN. 2])

(COMP.SPREAD
  [LAMBDA (L VAR FN APPLYTOEACH)                                   ; Edited 18-Dec-86 14:53 by lmm
    [while [AND (EQ (CL:FIRST L)
                    'MAPCAR)
                (LISTP (CL:THIRD L))
                (NULL (CDDDR L))
                (FMEMB (CAR (CL:THIRD L))
                       '(FUNCTION CL:FUNCTION))
                (LITATOM (CL:SECOND (CL:THIRD L]
       do (push APPLYTOEACH (CADR (CADDR L]
    (COMP.EXPR L)
    (PROG ((LSTCHECK (create TAG))
           (LP (create TAG))
           (LEV LEVEL)
           (FR FRAME))
          (COMP.STJUMP 'JUMP LSTCHECK)
          (SETQ LEVEL LEV)
          (SETQ FRAME FR)
          (COMP.STTAG LP)
          (COMP.STFN '(OPCODES COPY CAR)
                 1)
          [for X in APPLYTOEACH do (if (CL:SYMBOLP X)
                                       then (COMP.STFN X 1)
                                     else (LET ((N 1))
                                               (for ARG in (CDR X) do (COMP.EXPR ARG)
                                                                      (add N 1))
                                               (COMP.STFN (CAR X)
                                                      N]
          (COMP.STFN '(OPCODES SWAP CDR)
                 1)
          [COMP.EFFECT (LIST 'AND (LIST 'IGEQ (LIST 'ADD1VAR VAR)
                                        CL:CALL-ARGUMENTS-LIMIT)
                             '(LISPERROR "TOO MANY ARGUMENTS"]
          (COMP.STTAG LSTCHECK)
          (COMP.STJUMP 'NTJUMP LP)
          (COMP.VAL VAR)
          (COMP.VAL FN)
          (RETURN (COMP.STFN '.APPLYFN. 2])

(COMP.EVALFORM
  [LAMBDA NIL                                                      (* lmm "29-Jun-84 08:25")

    ;; Special code for compiling interpreter (see function \EVALFORM on LLNINTERP).  Assume *ARGVAL* bound to 0, *FN* bound, *TAIL* bound

    (OR (EQ COMPILE.CONTEXT 'RETURN)
        (SHOULDNT))                                                ; Must be in return context, since otherwise would have to pop off
                                                                   ; *ARGVAL* value
    (PROG ((DONE (create TAG))
           (LP (create TAG)))
          (COMP.STCONST '*ARGVAL*)                                 ; for BLIPVAL to find
          (COMP.STTAG LP)
          [COMP.VAL '(LISTP (SETQ *TAIL* (CDR *TAIL*]              ; *TAIL* initially bound to entire form.
          (COMP.STJUMP 'FJUMP DONE)
          [COMP.VAL '(\EVAL (CAR *TAIL*]                                  ; evaluate this argument
          [COMP.EFFECT (LIST 'AND (LIST 'IGREATERP '(SETQ *ARGVAL* (ADD1 *ARGVAL*))
                                        MAXARGS)
                             (LIST 'LISPERROR '"TOO MANY ARGUMENTS" '*TAIL*]
                                                                          ; increment counter of number of values
          (SETQ LEVEL (SUB1 LEVEL))                                       ; fool level check;  the value of the \EVAL is left on the stack,
                                                                          ; even though the compiler doesn't think so
          (COMP.STJUMP 'JUMP LP)
          (COMP.STTAG DONE)                                               ; there are really *ARGVAL* values on the stack
          [COMP.EFFECT '(AND *TAIL* (LISPERROR "UNUSUAL CDR ARG LIST" *TAIL*]
          [COMP.VAL '(PROG1 *ARGVAL* (SETQ *ARGVAL*]                      ; push number of arguments --- mark frame as done (see
                                                                          ; \DEADBLIPFRAME on LLNINTERP)
          (COMP.VAL '*FN*)                                                ; push name of function to call
          (COMP.STFN '.APPLYFN. 2)                                        ; this will execute applyfn opcode, which really takes N+2 args
                                                                          ; even though compiler thinks it takes 2
          (RETURN (COMP.STRETURN])
```

(**COMP.PUSHCALL**

```
  [LAMBDA (N FORM)                                                        (* lmm "16-APR-82 00:39")
    (COMP.VAL N)
    (PROG ((CHK (create TAG))
           (LP (create TAG))
           (LEV LEVEL)
           (FR FRAME))
          (COMP.STJUMP 'JUMP CHK)
          (SETQ LEVEL LEV)
          (SETQ FRAME FR)
          (COMP.STTAG LP)
          (COMP.STCONST)
          (COMP.STFN '.SWAPNIL. 2)
          (COMP.STCONST 1)
          (COMP.STFN 'IDIFFERENCE 2)
          (COMP.STTAG CHK)
          (COMP.STCOPY)
          (COMP.STCONST 0)
          (COMP.STFN 'IGREATERP 2)
          (COMP.STJUMP 'TJUMP LP)
          (COMP.STPOP)
          (RETURN (COMP.EXP1 FORM])
```

(**COMP.APPLY***

```
  [LAMBDA (A)                                                             (* lmm "13-Jul-84 21:18")
    (PROG (FN)
          (replace EXTCALL of FRAME with T)
          (COND
             ([AND (EQ (CAR (LISTP (CAR A)))
                       'FUNCTION)
                   (NULL (CDDAR A))
                   (LISTP (SETQ FN (CADR (CAR A]
              (SELECTQ (ARGTYPE FN)
                  ((0 2)
                       [RETURN (COMP.EXP1 (CONS FN (CDR A])
                  (3                                                      ; APPLY* of NLAMBDA nospread
                                                                          (* (AND (LITATOM FN) (RETURN
                                                                          (COMP.CALL FN (LIST (CONS (QUOTE LIST)
                                                                          (CDR A))) 0))))
                     )
                  (1                                                      ; APPLY* of NLAMBDA spread
                                                                          (* (AND (LITATOM FN) (RETURN
                                                                          (COMP.CALL FN (CDR A) 0))))
                     )
                  NIL)))
          (RETURN (COMP.EXP1 `((OPENLAMBDA (%, (SETQ FN (GENSYM)))
                                  ((OPCODES CHECKAPPLY* APPLYFN)
                                   %,@
                                   (CDR A)
                                   %,
                                   (LENGTH (CDR A))
                                   %, FN))
                               %,
                               (CAR A])
)
```

;; for ARG and SETARG

(PUTPROPS **ARG DMACRO COMP.ARG**)

(PUTPROPS **SETARG DMACRO COMP.SETARG**)

(PUTPROPS **NAMEDLET DMACRO COMP.NAMEDLET**)

(DEFINEQ

(**COMP.ARG**

```
  [LAMBDA (A)                                                          (* bvm%: "15-Oct-85 18:18")
    (COND
       ((AND (EQ COMPILE.CONTEXT 'EFFECT))
        (COMP.PROGN A))
       [(AND (EQ COMTYPE 2)
             (EQ (COMP.LOOKUPVAR (CAR A))
                 (CAR ARGVARS)))
        (COMP.VAL1 (CDR A))
        (COND
           ((AND COMPILE.ARG.FAST.FLG (EQ (fetch OPNAME of (CAR CODE))
                                          'CONST)
                 [FIXP (SETQ A (fetch OPARG of (CAR CODE]
                 (IGREATERP A 0)
                 (ILEQ A 255))
            (COMP.DELPUSH)
            (COMP.STFN [COND
                          ((IGREATERP A (OPCOUNT 'IVAR))
                           (LIST 'OPCODES 'IVARX (LLSH (SUB1 A)
                                                       1)))
                          (T (LIST 'OPCODES (LIST 'IVAR (SUB1 A]
                   0))
           (T (COMP.STFN '(OPCODES ARG0)
                     1]
       (T                                                              ; unreasonable ARG
          (COMP.CALL 'ARG (CONS (KWOTE (CAR A))
                                (CDR A))
                 0])
```

(**COMP.SETARG**

```
  [LAMBDA (A)                                                          (* lmm " 6-Dec-85 13:17")
    (COND
       [(AND (EQ COMTYPE 2)
             (EQ (COMP.LOOKUPVAR (CAR A))
                 (CAR ARGVARS)))
        (COMP.VAL (CADR A))
        (LET [(ARG (fetch OPARG of (CAR CODE]
             (COND
                ((AND COMPILE.ARG.FAST.FLG (EQ (fetch OPNAME of (CAR CODE))
                                               'CONST)
                      (FIXP ARG)
                      (IGREATERP ARG 0)
                      (ILEQ ARG 255))
                 (COMP.DELPUSH)
                 (COMP.VAL1 (CDDR A))
                 (COMP.STFN (LIST 'OPCODES 'IVARX_ (TIMES (SUB1 ARG)
                                                          2))
                        1))
                (T (COMP.VAL1 (CDDR A))
                   (COMP.STFN '\SETARG0 2]
       (T                                                              ; unreasonable ARG
          (COMP.CALL '\SETARG (CONS (KWOTE (CAR A))
                                    (CDR A))
                 0])
```

(**COMP.NAMEDLET**

```
  [LAMBDA (ARGS)                                                       (* lmm " 8-MAY-82 13:15")
    (PROG [(FN (COMP.LAM1 (CONS 'LAMBDA (CONS (MAPCAR (CAR (CDR ARGS))
                                                      (FUNCTION CAR))
                                              (CONS (LIST '\CALLME (KWOTE (CAR ARGS)))
                                                    (CDR (CDR ARGS]
          (RETURN (COMP.CALL FN [MAPCAR (CAR (CDR ARGS))
                                        (FUNCTION (LAMBDA (X)
                                                    (COND
                                                       ((CDR (CDR X))
                                                        (CONS 'PROG1 (CDR X)))
                                                       (T (CAR (CDR X]
                         0])
)
```

(PUTPROPS **LOADTIMECONSTANT DMACRO** [X (LIST 'QUOTE (CONS LOADTIMECONSTANTMARKER (CAR X])

(RPAQQ **LOADTIMECONSTANTMARKER** "LoadTimeConstant")

(PUTPROPS **DLAP FILETYPE** CL:COMPILE-FILE)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

```
(ACCESSFNS DASM [(FREEVARINDEX (GETHASH DATUM FVINDEXHARRAY)
                       (PUTHASH DATUM NEWVALUE FVINDEXHARRAY))
                 (VARINDEX (GETHASH DATUM VCA)
                        (PUTHASH DATUM NEWVALUE VCA))
                 (CLEAR (PROGN (OPT.INITHASH VCA)
                                (OPT.INITHASH FVINDEXHARRAY]        ; for alto assembler

       )
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS FVINDEXHARRAY)
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS PARENTP MACRO [LAMBDA (X Y)
                             (PROG NIL
                               LP  (RETURN (OR (EQ X Y)
                                               (AND (SETQ X (fetch PARENT of X))
                                                    (GO LP])

(PUTPROPS AST MACRO ((X)
                      (SETQ CD (CONS X CD))
                      (SETQ CODELOC (ADD1 CODELOC))))

(PUTPROPS OPCOUNT MACRO [LAMBDA (X)
                             (ADD1 (LET [(OP (fetch OP# of (\FINDOP X]
                                        (IDIFFERENCE (CADR OP)
                                               (CAR OP])
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS CHECKRANGE MACRO [(X N MSG)
                                (COND
                                   ((IGREATERP X (CONSTANT N))
                                    (COMPERRM (LIST X MSG '%, 'LIMIT 'IS (CONSTANT N])
)


(FILESLOAD (LOADCOMP)
       BYTECOMPILER LLCODE)
)
```

{MEDLEY}<sources>DLAP.;1   28-Jun-2024 18:34:03

-- Listed on 30-Jun-2024 13:15:42 --

FUNCTION INDEX

```
C.FLOATBOX .............2   COMP.GETBASEBITS .......19   COMP.SPREAD ............21   DASSEM.DPRINTLAP ........9
C.FLOATUNBOX ...........2   COMP.GETD ..............18   COMP.SPREADFN ..........20   DASSEM.DSTOREFN ........10
COMP.APPLY* ...........22   COMP.NAMEDLET ..........23   DASSEM.ASMAJ ...........10   DASSEM.DSTOREFNDEF ......7
COMP.ARG ..............22   COMP.PUSHCALL ..........22   DASSEM.CANSHAREBINDING ..9   DASSEM.DWRITEFN .........6
COMP.COMPARENUM .......17   COMP.PUSHNILS ..........21   DASSEM.CLEANFNTEST .....15   DASSEM.EQCONSTANTP ......9
COMP.EVALFORM .........21   COMP.RPLACD ............16   DASSEM.COUNTVARS ........9   DASSEM.MATCHVARS ........9
COMP.FMEMB ............18   COMP.SETARG ............23   DASSEM.DASMBIND ........10   DASSEM.SAVELOCALVARS ....7
COMP.GETBASE ..........19   COMP.SHIFT .............17   DASSEM.DASSEM ...........2
```

PROPERTY INDEX

```
.APPLYFN. ........11   FLESSP ...........12   LOGAND ...........12   \ADDBASE .........13   \GCRECLAIMCELL ...13
.SPREADCONS. .....21   FLOAT ............12   LOGOR ............12   \ARG0 ............13   \GCSCAN1 .........13
.SWAPNIL. ........21   FLOATP ...........12   LOGXOR ...........12   \BIN .............13   \GCSCAN2 .........13
ARRAYP ...........11   FMEMB ............12   LRSH1 ............12   \BLKEXPONENT .....13   \IBLT1 ...........13
ASSOC ............11   FPLUS ............12   LRSH8 ............12   \BLKFDIFF ........13   \IBLT2 ...........13
BIN ..............11   FQUOTIENT ........12   LSH ..............12   \BLKFLOATP2COMP ..13   \MAKENUMBER ......13
CAR ..............12   FTIMES ...........12   NFJUMP ...........11   \BLKFPLUS ........13   \MTIMES3 .........13
CDR ..............12   GREATERP .........12   NTJUMP ...........11   \BLKFTIMES .......13   \MTIMES4 .........13
CONS .............12   IDIFFERENCE ......12   NTYPX ............12   \BLKMAG ..........13   \MYALINK .........13
CREATECELL .......12   IGREATERP ........12   NULL .............12   \BLKPERM .........13   \MYARGCOUNT ......13
DIFFERENCE .......12   ILESSP ...........12   NUMBERP ..........12   \BLKSEP ..........13   \PILOTBITBLT .....13
DLAP .............23   IPLUS ............12   PLUS .............13   \BLKSMALLP2FLOAT .13   \PIXELBLT ........13
DMACRO ...........18   IQUOTIENT ........12   QUOTIENT .........13   \BLT .............13   \RCLK ............13
EQ ...............12   IREMAINDER .......12   READPRINTERPORT ..13   \BOXIDIFFERENCE ..13   \READFLAGS .......14
EQL ..............12   ITIMES ...........12   RPLACA ...........13   \BOXIPLUS ........13   \READRP ..........14
EQUAL ............12   JUMP .............11   RPLACD ...........13   \CONTEXTSWITCH ...13   \RPLCONS .........14
FDIFFERENCE ......12   LESSP ............12   SMALLP ...........13   \DRAWLINE.UFN ....13   \STKSCAN .........14
FGREATERP ........12   LISTGET ..........12   STACKP ...........13   \EVAL ............13   \WRITEMAP ........14
FIX ..............12   LISTP ............12   TIMES ............13   \FLOAT.BOX .......13   \\ADDBASE ........14
FIXP .............12   LLSH1 ............12   TJUMP ............11   \FLOATBOX ........13
FJUMP ............11   LLSH8 ............12   WRITEPRINTERPORT .13   \FLOATUNBOX ......13
```

MACRO INDEX

```
.CALLAFTERPUSHINGNILS. .20   FLESSP .................16   LRSH ...................16   \FLOATUNBOX ............16
.EVALFORM. .............20   FMEMB ..................16   NAMEDLET ...............22   \GETBASE ...............19
.PUSHNILS. .............20   FRPLACD ................16   NILAPPLY ...............20   \GETBASEPTR ............19
.SPREAD. ...............20   GETD ...................16   OPCOUNT ................24   \GETBITS ...............19
APPLY* .................21   GREATERP ...............16   PARENTP ................24   \PUTBASE ...............19
ARG ....................22   IGREATERP ..............16   PRINTNUM ...............16   \PUTBASEPTR ............19
AST ....................24   ILESSP .................16   RPLACD .................16   \PUTBITS ...............19
CHECKRANGE .............24   LESSP ..................16   SETARG .................22   \RPLPTR ................19
FGETD ..................16   LLSH ...................16   SPREADAPPLY ............20
FGREATERP ..............16   LOADTIMECONSTANT .......23   \FLOATBOX ..............16
```

VARIABLE INDEX

```
8BITEXTS ...............14   CONDITIONALS ...........14   LAMBDANOBIND ...........14   NUMBERFNS ..............14
BYTEASSEMFN ............14   CONST.FNS ..............14   LOADTIMECONSTANTMARKER .23   OPCODEPROP .............14
BYTECOMPFLG ............14   CONSTFNS ...............14   MACROPROPS .............14   SELECTQFMEMB ...........14
CLEANFNLIST ............14   CONSTOPS ...............14   MAXARGS ................14   SELECTVARTYPES .........14
CLEANFNTEST ............14   DOPVALS ................11   MAXBVALS ...............14   VCONDITIONALS ..........14
COMPILE.ARG.FAST.FLG ...14   EMFLAG .................11   MERGEFRAMEFLG ..........14   XVARFLG ................14
COMPILEMODE ............11   EQCONSTFN ..............14   MERGEFRAMEMAX ..........14
COMPILERMACROPROPS .....14   IPLUSNFLG ..............14   NOFREEVARFNS ...........14
```

OPTIMIZER INDEX

```
APPEND ...........20   FRPLACA ..........15   LITATOM ..........15   SPREADAPPLY* .....16   \LOLOC ...........18
ATOM .............15   GETATOMVAL .......15   MINUSP ...........15   SYSTEMTYPE .......16   \PUTBASEBYTE .....18
EVALV ............15   IEQP .............15   NCONC ............20   \GETBASEBYTE .....18   \VAG2 ............19
FASSOC ...........16   LIST .............15   SETATOMVAL .......16   \HILOC ...........18
```

CONSTANT INDEX

```
NARGMAX ..........10   NFREEMAX .........10   NLOCALMAX ........10   SHALLOWFLG .......15   SPAGHETTIFLG .....15
```

RECORD INDEX

DASM 23