File created: 19-Jan-93 10:45:33 **{DSK}<python>lde>lispcore>sources>LLCODE.;2**

changes to: (RECORDS COMPILED-CLOSURE CODEARRAY OPCODE UFNENTRY)

previous date: 5-Jan-93 00:05:55 {DSK}<python>lde>lispcore>sources>LLCODE.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;

(RPAQQ **LLCODECOMS**

```
[;; THIS FILE IS DUPLICATED on <Lispcore>Sources> and <Lispcore>Sources>2-byte>, with the latter being the old 2-byte-atom version.  IF
 ;; YOU CHANGE THIS ONE, CHANGE THE OTHER ONE!

 [COMS                                                    ; reading in compiled code
       (FNS DCODERD DCODESKIP \ALLOC.CODE.BLOCK \REALNAMEP \RENAMEDFN)
       (DECLARE%: DONTEVAL@LOAD DOCOPY [VARS (CODERDTBL (COPYREADTABLE 'ORIG]
              (P (SETSYNTAX 25 '[MACRO (LAMBDA (FILE RDTBL)
                                               (EVAL (READ FILE RDTBL]
                                         CODERDTBL)
                 (SETSYNTAX 124 '(MACRO ALWAYS READVBAR)
                                         CODERDTBL)
                 (READTABLEPROP CODERDTBL 'USESILPACKAGE NIL)))
       (GLOBALVARS CODERDTBL FILERDTBL)
       (COMS ;; CODEINDICATOR is the token the compiler puts out in front of compiled definitions.  To switch to an incompatible compiled
             ;; code version, choose a new value for CODEINDICATOR.  If old compiled code is still loadable in the new system, retain
             ;; the CODEREADER prop for an indicators that are still loadable.

             ;; CODEINDICATOR changed to :D6 4/6/90 by JDS for Medley 1.15, because of additional opcodes emitted by compiler.

             ;; CODEINDICATOR changed to :D7 by JDS 3/4/91 for Medley 1.3, because of 3-byte atoms.  Old CODEREADER properties
             ;; removed at the same time.

             ;; Changed to :D8 by JDS 11/12/92 for Medley 2.1/3.0 because of 4-byte pointers/4-byte atoms.  Old CODEREADER
             ;; property removed as well, since old code is not readable.

             (VARS (CODEINDICATOR ':D8))
             (GLOBALVARS CODEINDICATOR)
             (PROP CODEREADER * (LIST CODEINDICATOR]
 [COMS                                                    ; Compiled CLOSURE type
       (FNS MAKE-COMPILED-CLOSURE \CCLOSURE.DEFPRINT \GET-COMPILED-DEFINITION \GET-COMPILED-CODE-BASE
            EQDEFP)
       (DECLARE%: EVAL@COMPILE DONTCOPY (EXPORT (RECORDS COMPILED-CLOSURE)
                                                (CONSTANTS \COMPILED-CLOSURE)
                                                (MACROS \EXTENDED.EQP)))
       (INITRECORDS COMPILED-CLOSURE)
       (DECLARE%: DONTEVAL@LOAD DOCOPY (P (DEFPRINT 'COMPILED-CLOSURE '\CCLOSURE.DEFPRINT]
 [COMS                                                    ; utilities
       (FNS \FINDOP OP#)
       ;; List of opcodes known to the system.  Used to drive the compilers and build the UFN table.
       ;; Format of an entry: (op# name #-extra-bytes ?? stack-effect
       (VARS \OPCODES)
       (ADDVARS (\OPCODEARRAY))
       (GLOBALVARS \OPCODEARRAY \OPCODES)
       (DECLARE%: EVAL@COMPILE DONTCOPY (FNS WORDSPERNAMEENTRY)
              (EXPORT (MACROS DPUTCODE MCODEP)
                      (MACROS CODELT CODELT2 CODESETA2 CODESETA)
                      (MACROS BYTESPERNAMEENTRY BYTESPERNTOFFSETENTRY GETNAMEENTRY GETNTFLAGS GETNTOFFSET
                              GETNTOFFSETENTRY GETNTTAG SETNAMEENTRY WORDSPERNTOFFSETENTRY NTSLOT-OFFSET)
                      (FUNCTIONS NEW-SYMBOL-CODE)
                      (OPTIMIZERS BIG-VMEM-CODE SETSTKNAMEENTRY SETSTKNTOFFSETENTRY GETSTKNAMEENTRY
                              GETSTKNTOFFSETENTRY WORDSPERNAMEENTRY SETSTKNTOFFSET SETSTKNAME-RAW
                              SETSTKNTOFFSET-RAW NEW-SYMBOL-CODE MAKE-NTENTRY NULL-NTENTRY)
                      (OPTIMIZERS NTSLOT-VARTYPE)
                      (RECORDS CODEARRAY)
                      (RECORDS OPCODE)
                      (GLOBALVARS \OPCODES)
                      (CONSTANTS PVARCODE FVARCODE IVARCODE VARCODEMASK)
                      (CONSTANTS \NT.IVARCODE \NT.PVARCODE \NT.FVARCODE]
 [COMS                                                    ; ufns
       (FNS INITUFNTABLE \SETUFNENTRY \GETUFNENTRY)
       (FNS \UNKNOWN.UFN)
       [DECLARE%: DONTEVAL@LOAD DOCOPY                    ; To go into the INIT
              ;; INITIALIZE THE TARGET ARCHITECTURE.
              (INITVARS (COMPILER::*TARGET-ARCHITECTURE* '(:4-BYTE :3-BYTE))
                     (COMPILER::*HOST-ARCHITECTURE* '(:4-BYTE :3-BYTE]
       (DECLARE%: DONTCOPY (RECORDS UFNENTRY)
              (ADDVARS (INEWCOMS (FNS INITUFNTABLE \SETUFNENTRY)))
              EVAL@COMPILE
              (ADDVARS (DONTCOMPILEFNS INITUFNTABLE]
```

```
        [COMS                                                    ; for MAKEINIT and READSYS
              (DECLARE%: DONTCOPY (ADDVARS (INEWCOMS (FNS DCODERD)
                                                   [VARS \OPCODES (CODERDTBL (COPYREADTABLE 'ORIG]
                                                   (P (SETSYNTAX (CHARCODE ^Y)
                                                                 '[MACRO (LAMBDA (FILE RDTBL)
                                                                                 (EVALFORMAKEINIT (READ FILE RDTBL]
                                                                 CODERDTBL)
                                                          (SETSYNTAX (CHARCODE %|)
                                                                 '(MACRO ALWAYS READVBAR)
                                                                 CODERDTBL)
                                                          (READTABLEPROP CODERDTBL 'USESILPACKAGE NIL)))
                                                (MKI.SUBFNS (\CODEARRAY . SCRATCHARRAY)
                                                       (DPUTCODE . I.PUTDEFN)
                                                       (CODERDTBL . I.CODERDTBL)
                                                       (SETSTKNTOFFSET . I.SETSTKNTOFFSET)
                                                       (WORDSPERNAMEENTRY . I.WORDSPERNAMEENTRY))
                                                (EXPANDMACROFNS CODELT CODELT2 CODESETA CODESETA2 DPUTCODE MCODEP
                                                       BYTESPERNAMEENTRY BYTESPERNTOFFSETENTRY WORDSPERNAMEENTRY)
                                                (RD.SUBFNS (CODELT . VGETBASEBYTE)
                                                       (CODESETA . VPUTBASEBYTE))
                                                (RDCOMS (FNS \GET-COMPILED-CODE-BASE]
        (PROP FILETYPE LLCODE)
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA OP#)
                                                                                (NLAML)
                                                                                (LAMA])
```

;; THIS FILE IS DUPLICATED on <Lispcore>Sources> and <Lispcore>Sources>2-byte>, with the latter being the old 2-byte-atom version. IF YOU ;; CHANGE THIS ONE, CHANGE THE OTHER ONE!

;; reading in compiled code

```
(DEFINEQ
```

(**DCODERD**

```
  [LAMBDA (FN)                                               ; Edited 28-Jan-91 15:45 by jds
```

;; Read a function definition from an LCOM file.

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

```
    (READC)
    (LET ((INSTREAM (GETSTREAM NIL 'INPUT))
          (*READTABLE* (if (EQ *READTABLE* FILERDTBL)
                           then                              ; old style file, read code with different read table!
                                CODERDTBL
                         else                                ; read code in same readtable
                              *READTABLE*)))
         (PROG ((NAMETABLE (PROG1 (READ)
                                  (READC)))
                (CODELEN (IPLUS (LLSH (\BIN INSTREAM)
                                      8)
                                (\BIN INSTREAM)))
                (NLOCALS (\BIN INSTREAM))
                (NFREEVARS (\BIN INSTREAM))
                (ARGTYPE (\BIN INSTREAM))
                (NARGS (\BIN INSTREAM))
                (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 [ADD1 (UNFOLD 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 CODELEN)
                                   BYTESPERQUAD))
              (SETQ CA (\CODEARRAY REALSIZE (CEIL (ADD1 (FOLDHI STARTPC BYTESPERCELL))
                                                  CELLSPERQUAD)))
              (AIN CA STARTPC CODELEN INSTREAM)
```

;; 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 (LLSH \NT.PVARCODE 14)))
                                (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 (LLSH \NT.FVARCODE 14)))
                                (I (CONSTANT (LLSH \NT.IVARCODE 14)))
                                (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 (CADR X)
                                                                         -1)
                                                                ; Name in first half
                                                                 (SETSTKNTOFFSET CA (IPLUS NT LOCALSIZE)
                                                                        (CONSTANT (LLSH \NT.IVARCODE 14))
                                                                        (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))
```

;; Now read fixups: 3 lists in plist format of function fixups, symbol fixups and random pointer fixups.

```
              (for X on (READ) by (CDDR X) do (\FIXCODESYM CA (IPLUS (CAR X)
                                                                     STARTPC)
                                                             (CADR X)
                                                             -1))
              (for X on (READ) by (CDDR X) do (\FIXCODESYM CA (IPLUS (CAR X)
                                                                     STARTPC)
                                                             (CADR X)
                                                             -1))
              [for X on (READ) by (CDDR X) do (\FIXCODEPTR CA (IPLUS (CAR X)
                                                                     STARTPC)
                                                             (EVQ (CADR X]
              (DPUTCODE FN CA (IPLUS STARTPC CODELEN])
```

(**DCODESKIP**

```
  [LAMBDA (FN FLG)                                              (* bvm%: " 2-Oct-86 21:39")
```

;;; If FLG is true then copy code from input to output, else just skip code on input. For copy case, source and destination read tables must be the same!

```
    (PROG ((INSTREAM (GETSTREAM NIL 'INPUT))
```

```
           (RDTBL (if (EQ *READTABLE* FILERDTBL)
                      then                                    ; old style file, read code with different read table!
                           CODERDTBL
                    else                                      ; read code in same readtable
                         *READTABLE*))
           CODELEN START)
          (READC INSTREAM)                                    ; Skip EOL after code indicator
          [COND
             (FLG                                             ; In both cases, scan over the code.  When FLG is true, we will
                                                              ; copy when done
                  (SETQ START (GETFILEPTR INSTREAM]
          (SKREAD INSTREAM)                                   ; Skip localvar args
          (READC INSTREAM)
          (SETQ CODELEN (IPLUS (LLSH (\BIN INSTREAM)
                                     8)
                               (\BIN INSTREAM)))
          (\BIN INSTREAM)
          (\BIN INSTREAM)
          (\BIN INSTREAM)
          (\BIN INSTREAM)
          (SETFILEPTR INSTREAM (IPLUS (GETFILEPTR INSTREAM)
                                      CODELEN))               ; Skip the code itself
          (SKREAD INSTREAM NIL RDTBL)                         ; Skip 3 lists of fixups
          (SKREAD INSTREAM NIL RDTBL)
          (SKREAD INSTREAM NIL RDTBL)
          (READC INSTREAM RDTBL)
          (COND
             (FLG                                             ; copy it all to destination.  We assume reader environments are
                                                              ; the same
                  (PRIN4 FN)
                  (PRIN3 " ")
                  (PRIN4 CODEINDICATOR)
                  (TERPRI)
                  (COPYBYTES INSTREAM NIL START (GETFILEPTR])
```

(\ALLOC.CODE.BLOCK

```
  [LAMBDA (NBYTES INITONPAGE)                                 (* bvm%: " 8-Jul-86 17:09")
    (\ALLOCBLOCK (FOLDHI NBYTES BYTESPERCELL)
           CODEBLOCK.GCT INITONPAGE CELLSPERQUAD])
```

(\REALNAMEP

```
  [LAMBDA (X)                                                 (* lmm "15-OCT-81 00:16")
    (AND (NEQ X 'ERRORSET)
         (NEQ (NTHCHAR X 1)
              '\])
```

(\RENAMEDFN

```
  [LAMBDA (DEF FN)                                            ; Edited  3-Mar-87 22:32 by bvm:
    ;; Used by PUTD when doing movds from one function to another
    (LET* [[CODEBASE (fetch (COMPILED-CLOSURE FNHEADER) of (\DTEST DEF 'COMPILED-CLOSURE]
           (WORDSIZE (UNFOLD (\#BLOCKDATACELLS CODEBASE)
                             WORDSPERCELL))
           (NEWCA (\ALLOC.CODE.BLOCK (UNFOLD WORDSIZE BYTESPERWORD)
                                     (CEIL (ADD1 (FOLDHI (fetch (FNHEADER STARTPC) of CODEBASE)
                                                         BYTESPERCELL))
                                           CELLSPERQUAD]
          (\COPYCODEBLOCK NEWCA CODEBASE WORDSIZE FN)
          (create COMPILED-CLOSURE using DEF FNHEADER _ NEWCA])

)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(RPAQ CODERDTBL (COPYREADTABLE 'ORIG))

(SETSYNTAX 25 '[MACRO (LAMBDA (FILE RDTBL)
                      (EVAL (READ FILE RDTBL]
       CODERDTBL)

(SETSYNTAX 124 '(MACRO ALWAYS READVBAR)
       CODERDTBL)

(READTABLEPROP CODERDTBL 'USESILPACKAGE NIL)
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS CODERDTBL FILERDTBL)
)

;; CODEINDICATOR is the token the compiler puts out in front of compiled definitions.  To switch to an incompatible compiled code version, choose a
;; new value for CODEINDICATOR.  If old compiled code is still loadable in the new system, retain the CODEREADER prop for an indicators that are still
;; loadable.
```

;; CODEINDICATOR changed to :D6 4/6/90 by JDS for Medley 1.15, because of additional opcodes emitted by compiler.

;; CODEINDICATOR changed to :D7 by JDS 3/4/91 for Medley 1.3, because of 3-byte atoms. Old CODEREADER properties removed at the same ;; time.

;; Changed to :D8 by JDS 11/12/92 for Medley 2.1/3.0 because of 4-byte pointers/4-byte atoms. Old CODEREADER property removed as well, since ;; old code is not readable.

```
(RPAQQ CODEINDICATOR :D8)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS CODEINDICATOR)
)

(PUTPROPS :D8 CODEREADER (DCODERD . DCODESKIP))
```

;; Compiled CLOSURE type

```
(DEFINEQ

(MAKE-COMPILED-CLOSURE
  [LAMBDA (CODEBASE ENVIRONMENT)                                    (* bvm%: " 7-Jul-86 11:32")
    (create COMPILED-CLOSURE
            FNHEADER _ CODEBASE
            ENVIRONMENT _ ENVIRONMENT])

(\CCLOSURE.DEFPRINT
  [LAMBDA (CLOSURE STREAM)                                          (* bvm%: " 7-Jul-86 15:50")
```

;;; Print closure object as, for example, #<Compiled Closure FOOBAR/76,5432>

```
    (LET [(NAME (fetch (COMPILED-CLOSURE FRAMENAME) of CLOSURE))
          (TYPE (COND
                    ((fetch (COMPILED-CLOSURE ENVIRONMENT) of CLOSURE)
                     "Closure")
                    (T "Function"]
          (.SPACECHECK. STREAM (IPLUS (CONSTANT (NCHARS "<Compiled Function />"))
                                      (PROGN                        ; Longest stack address is 177,177777
                                             10)
                                      (COND
                                         ((OR (LITATOM NAME)
                                              (STRINGP NAME))
                                          (NCHARS NAME (LITATOM NAME)))
                                         (T (SETQ NAME)
                                            0))
                                      1))
          (\OUTCHAR STREAM (fetch (READTABLEP HASHMACROCHAR) of *READTABLE*))
          (\SOUT "<Compiled " STREAM)
          (\SOUT TYPE STREAM)
          [COND
             (NAME (\OUTCHAR STREAM (CHARCODE SPACE))
                   (COND
                      ((STRINGP NAME)
                       (\SOUT NAME STREAM))
                      (T (\PRINDATUM NAME STREAM]
          (\OUTCHAR STREAM (CHARCODE /))
          (\PRINTADDR CLOSURE STREAM)
          (\OUTCHAR STREAM (CHARCODE >))
          T])
```

```
(\GET-COMPILED-DEFINITION
  [LAMBDA (X)                                                       (* bvm%: "11-Jul-86 16:28")
```

;;; X is an object denoting a function somehow. If it represents a compiled function, return a CLOSURE object for it

```
    (PROG NIL
          [COND
             ((LITATOM X)
              (COND
                 ((PROG1 (fetch (LITATOM CCODEP) of X)
                         (SETQ X (fetch (LITATOM DEFPOINTER) of X)))
                  (RETURN (MAKE-COMPILED-CLOSURE X]
          (RETURN (AND (type? COMPILED-CLOSURE X)
                       X])
```

```
(\GET-COMPILED-CODE-BASE
  [LAMBDA (X)                                                       (* bvm%: "11-Jul-86 16:26")
```

;;; X is an object denoting a function somehow. If it represents a compiled function, return its code base

```
    (PROG NIL
          [COND
             ((LITATOM X)
```

```
              (COND
                 ((PROG1 (fetch (LITATOM CCODEP) of X)
                      (SETQ X (fetch (LITATOM DEFPOINTER) of X)))
                  (RETURN X]
           (RETURN (AND (EQ (NTYPX X)
                            \COMPILED-CLOSURE)
                        (fetch (COMPILED-CLOSURE FNHEADER) of X])
```

(**EQDEFP**

```
  [LAMBDA (CA1 CA2)                                                    (* bvm%: " 7-Jul-86 22:36")
    ;; determines whether two code arrays CA1 and CA2 are equivalent (same except for framename)
    (COND
       ((AND (TYPEP CA1 'COMPILED-CLOSURE)
             (TYPEP CA2 'COMPILED-CLOSURE)
             (EQ (fetch (COMPILED-CLOSURE ENVIRONMENT) of CA1)
                 (fetch (COMPILED-CLOSURE ENVIRONMENT) of CA2)))
        (SETQ CA1 (fetch (COMPILED-CLOSURE FNHEADER) of CA1))
        (SETQ CA2 (fetch (COMPILED-CLOSURE FNHEADER) of CA2))
        (for I from 0 to (SUB1 (UNFOLD (IMIN (\#BLOCKDATACELLS CA1)
                                             (\#BLOCKDATACELLS CA2))
                                       WORDSPERCELL))
           always (OR (EQ (\GETBASE CA1 I)
                          (\GETBASE CA2 I))
                      [EQ I (INDEXF (fetch (FNHEADER %#FRAMENAME]
                      (EQ I (ADD1 (INDEXF (fetch (FNHEADER %#FRAMENAME])

)

(DECLARE%: EVAL@COMPILE DONTCOPY

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

(DATATYPE COMPILED-CLOSURE (FNHEADER ENVIRONMENT))
)

(/DECLAREDATATYPE 'COMPILED-CLOSURE '(POINTER POINTER)
        ;; ---field descriptor list elided by lister---
        '4)

(DECLARE%: EVAL@COMPILE

(RPAQQ \COMPILED-CLOSURE 13)

(CONSTANTS \COMPILED-CLOSURE)
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \EXTENDED.EQP MACRO [OPENLAMBDA (X Y)
                                  (COND
                                     ((EQ (NTYPX X)
                                          (NTYPX Y))
                                      (SELECTC (NTYPX X)
                                           (\STACKP (EQ (fetch (STACKP EDFXP) of X)
                                                        (fetch (STACKP EDFXP) of Y)))
                                           (\COMPILED-CLOSURE
                                                (EQDEFP X Y))
                                           NIL])
)
)

;; END EXPORTED DEFINITIONS

(/DECLAREDATATYPE 'COMPILED-CLOSURE '(POINTER POINTER)
        ;; ---field descriptor list elided by lister---
        '4)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(DEFPRINT 'COMPILED-CLOSURE '\CCLOSURE.DEFPRINT)
)

;; utilities

(DEFINEQ
```

(**FINDOP**

```
  [LAMBDA (OPNAME FLG)                                                 (* lmm "22-Mar-85 10:20")
    (ALLOCAL (PROGN [OR \OPCODEARRAY (PROGN (SETQ \OPCODEARRAY (ARRAY 256 'POINTER NIL 0))
                                            (for X in \OPCODES
```

```
                                                    do (PUTPROP (fetch OPCODENAME of X)
                                                               'DOPCODE X)
                                                       (if (LISTP (fetch OP# of X))
                                                           then (for I from (CAR (fetch OP# of X))
                                                                   to (CADR (fetch OP# of X)) by 1
                                                                   do (SETA \OPCODEARRAY I X))
                                                         else (SETA \OPCODEARRAY (fetch OP# of X)
                                                                    X]
                    (OR (COND
                           ((LITATOM OPNAME)
                            (GETPROP OPNAME 'DOPCODE))
                           ((FIXP OPNAME)
                            (ELT \OPCODEARRAY OPNAME)))
                        (AND FLG (ERROR OPNAME FLG])
```

(**OP#**

```
  [NLAMBDA X                                                    (* lmm "12-FEB-82 23:50")
    (CAR (\FINDOP (CAR X])
```

)

;; List of opcodes known to the system. Used to drive the compilers and build the UFN table.

;; Format of an entry: (op# name #-extra-bytes ?? stack-effect

```
(RPAQQ \OPCODES
       ((0 -X- 0)
        (1 CAR 0 T 0 \CAR.UFN)
        (2 CDR 0 T 0 \CDR.UFN)
        (3 LISTP 0 T 0 LISTP)
        (4 NTYPX 0 T 0 NTYPX)
        (5 TYPEP 1 TYPEP 0 \TYPEP.UFN)
        (6 DTEST 4 ATOM 0 \DTEST.UFN)
        (7 UNWIND 2 T (UNWIND 1)
           \UNWIND.UFN)
        (8 FN0 4 FN 1)
        (9 FN1 4 FN 0)
        (10 FN2 4 FN -1)
        (11 FN3 4 FN -2)
        (12 FN4 4 FN -3)
        (13 FNX 5 FNX FNX)
        (14 APPLYFN 0 T -1)
        (15 CHECKAPPLY* 0 T 0 \CHECKAPPLY* (4K 12K))
        (16 RETURN 0 T (JUMP 1)
            \HARDRETURN)
        (17 BIND 2)
        (18 UNBIND 0)
        (19 DUNBIND 0)
        (20 RPLPTR.N 1 T -1 \RPLPTR.UFN (4K))
        (21 GCREF 1 T 0 \HTFIND)
        (22 ASSOC 0 T -1 ASSOC (4K DORADO))
        (23 GVAR_ 4 ATOM 0 \SETGLOBALVAL.UFN)
        (24 RPLACA 0 T -1 \RPLACA.UFN 4K)
        (25 RPLACD 0 T -1 \RPLACD.UFN 4K)
        (26 CONS 0 T -1 \CONS.UFN)
        (27 CMLASSOC 0 T -1 CL::%%SIMPLE-ASSOC (4K DORADO))
        (28 FMEMB 0 T -1 FMEMB (4K DORADO))
        (29 CMLMEMBER 0 T -1 CL::%%SIMPLE-MEMBER (4K DORADO))
        (30 FINDKEY 1 T 0 \FINDKEY.UFN)
        (31 CREATECELL 0 T 0 \CREATECELL 4K)
        (32 BIN 0 T 0 \BIN 4K)
        (33 BOUT 0 T -1 \BOUT (4K DORADO))
        (34 POPDISP 0 T 0 \POPDISP.UFN (4K DORADO))
        (35 RESTLIST 1 T -1 \RESTLIST.UFN)
        (36 MISCN 2 T 1 \MISCN.UFN (DORADO DLION DBREAK))
        (37 unused)
        (38 RPLCONS 0 T -1 \RPLCONS (4K DORADO))
        (39 LISTGET 0 T -1 LISTGET (4K DORADO))
        (40 unused)
        (41 unused)
        (42 unused)
        (43 unused)
        (44 EVAL 0 T 0 \EVAL)
        (45 ENVCALL 0 T (JUMP 0)
            \ENVCALL.UFN)
        (46 TYPECHECK 0 T 0 \TYPECHECK.UFN)
        (47 STKSCAN 0 T 0 \STKSCAN)
        (48 BUSBLT 1 (WORDSOUT BYTESOUT BYTESOUTSWAPPED NYBBLESOUT WORDSIN BYTESIN BYTESINSWAPPED
                           NYBBLESINSWAPPED)
            -3 \BUSBLT.UFN (4K DORADO))
        (49 MISC8 1 (IBLT1 IBLT2)
            -7 \MISC8.UFN (4K DORADO))
        (50 UBFLOAT3 1 (POLY MATRIX.3X3 MATRIX.4X4 MATRIX.133 MATRIX.331 MATRIX.144 MATRIX.441 UBASET1)
            (-2 1)
            \UNBOXFLOAT3
            (4K DORADO))
```

```
        (51 TYPEMASK.N 1 T 0 \TYPEMASK.UFN)
        (52 RDPROLOGPTR 0 T 0 RAID (4K DORADO))
        (53 RDPROLOGTAG 0 T 0 RAID (4K DORADO))
        (54 WRTPTR&TAG 0 T -2 RAID (4K DORADO))
        (55 WRTPTR&0TAG 0 T -1 RAID (4K DORADO))
        (56 MISC7 1 (PSEUDOCOLOR \FASTBITMAPBIT)
            -6 \MISC7.UFN (4K DORADO))
        (57 DOVEMISC 1 (READIW WRITEIO WRITEMP RDTIMER BYTESWAP LOCKMEM NOTIFYIOP SETWP)
            (0 -1 0 0 0 -3 0 0))
        (58 EQL 0 T -1 EQL)
        (59 DRAWLINE 0 T -8 \DRAWLINE.UFN (4K DORADO))
        (60 STORE.N 1 T 0 \STORE.N.UFN)
        (61 COPY.N 1 T 1 \COPY.N.UFN)
        (62 RAID 0 T 0 RAID T)
        (63 \RETURN 0 T 0 \RETURN)
        ((64 70)
         IVAR 0 IVAR 1)
        (71 IVARX 1 IVAR 1)
        ((72 78)
         PVAR 0 PVAR 1)
        (79 PVARX 1 PVAR 1)
        ((80 86)
         FVAR 0 FVAR 1)
        (87 FVARX 1 FVAR 1)
        ((88 94)
         PVAR_ 0 PVAR 0)
        (95 PVARX_ 1 PVAR 0)
        (96 GVAR 4 ATOM 1)
        (97 ARG0 0 T 0 \ARG0 T)
        (98 IVARX_ 1 IVAR 0)
        (99 FVARX_ 1 FVAR 0)
        (100 COPY 0 T 1)
        (101 MYARGCOUNT 0 T 1 \MYARGCOUNT T)
        (102 MYALINK 0 T 1)
        (103 ACONST 4 ATOM 1)
        (104 %'NIL 0 T 1)
        (105 %'T 0 T 1)
        (106 %'0 0 T 1)
        (107 %'1 0 T 1)
        (108 SIC 1 SIC 1)
        (109 SNIC 1 SNIC 1)
        (110 SICX 2 SICX 1)
        (111 GCONST 4 GCONST 1)
        (112 unused)
        (113 READFLAGS 0 T 0 \READFLAGS)
        (114 READRP 0 T 0 \READRP)
        (115 WRITEMAP 0 T -2 \WRITEMAP DORADO)
        (116 READPRINTERPORT 0 T 1 \READPRINTERPORT.UFN 4K)
        (117 WRITEPRINTERPORT 0 T 0 \WRITEPRINTERPORT.UFN 4K)
        (118 PILOTBITBLT 0 T -1 \PILOTBITBLT)
        (119 RCLK 0 T 0 \RCLKSUBR)
        (120 MISC1 1 (error INPUT OUTPUT error error error error error RWMUFMAN)
             0 \MISC1.UFN)
        (121 MISC2 1 (?0 ?1 ?2 ?3 ?4 ?5 ?6 ?7 ?10)
             -1 \MISC2.UFN)
        (122 RECLAIMCELL 0 T 0 \GCRECLAIMCELL DORADO)
        (123 GCSCAN1 0 T 0 \GCSCAN1)
        (124 GCSCAN2 0 T 0 \GCSCAN2)
        (125 SUBRCALL 2 SUBRCALL)
        (126 CONTEXTSWITCH 0 T 0 \CONTEXTSWITCH)
        (127 RETCALL 4 FNX (JUMP 1)
             \RETCALL)
        ((128 143)
         JUMP 0 JUMP JUMP NIL)
        ((144 159)
         FJUMP 0 JUMP CJUMP NIL)
        ((160 175)
         TJUMP 0 JUMP CJUMP NIL)
        (176 JUMPX 1 JUMPX JUMP)
        (177 JUMPXX 2 JUMPXX JUMP)
        (178 FJUMPX 1 JUMPX CJUMP)
        (179 TJUMPX 1 JUMPX CJUMP)
        (180 NFJUMPX 1 JUMPX NCJUMP)
        (181 NTJUMPX 1 JUMPX NCJUMP)
        (182 AREF1 0 T -1 %%AREF1 (4K DORADO))
        (183 ASET1 0 T -2 %%ASET1 (4K DORADO))
        ((184 190)
         PVAR_^ 0 PVAR -1 NIL)
        (191 POP 0 T -1)
        (192 POP.N 1 T (POP.N 1)
             \POP.N.UFN)
        (193 ATOMCELL.N 1 T 0 \ATOMCELL)
        (194 GETBASEBYTE 0 T -1 \GETBASEBYTE)
        (195 INSTANCEP 4 ATOM 0 \INSTANCEP.UFN NIL)
        (196 BLT 0 T -2 \BLT)
        (197 MISC10 1 T -9 \MISC10.UFN (4K DORADO))
        (198 P-MISC2 1 (GET-NEXT-RUN)
```

```
          -1 \P-MISC2.UFN)
        (199 PUTBASEBYTE 0 T -2 \PUTBASEBYTE)
        (200 GETBASE.N 1 T 0)
        (201 GETBASEPTR.N 1 T 0)
        (202 GETBITS.N.FD 2 T 0)
        (203 unused)
        (204 CMLEQUAL 0 T -1 CL:EQUAL (4K 12K DORADO))
        (205 PUTBASE.N 1 T -1 \PUTBASE.UFN)
        (206 PUTBASEPTR.N 1 T -1 \PUTBASEPTR.UFN)
        (207 PUTBITS.N.FD 2 T -1 \PUTBITS.UFN)
        (208 ADDBASE 0 T -1 \ADDBASE)
        (209 VAG2 0 T -1 \VAG2)
        (210 HILOC 0 T 0)
        (211 LOLOC 0 T 0)
        (212 PLUS2 0 T -1 \SLOWPLUS2 *)
        (213 DIFFERENCE 0 T -1 \SLOWDIFFERENCE *)
        (214 TIMES2 0 T -1 \SLOWTIMES2 *)
        (215 QUOTIENT 0 T -1 \SLOWQUOTIENT *)
        (216 IPLUS2 0 T -1 \SLOWIPLUS2)
        (217 IDIFFERENCE 0 T -1 \SLOWIDIFFERENCE)
        (218 ITIMES2 0 T -1 \SLOWITIMES2)
        (219 IQUOTIENT 0 T -1 \SLOWIQUOTIENT)
        (220 IREMAINDER 0 T -1 IREMAINDER)
        (221 IPLUS.N 1 T 0 \SLOWIPLUS2 (4K 12K))
        (222 IDIFFERENCE.N 1 T 0 \SLOWIDIFFERENCE (4K 12K))
        (223 BASE-< 0 T -1 \BASE-<.UFN (4K 12K DORADO))
        (224 LLSH1 0 T 0 \SLOWLLSH1)
        (225 LLSH8 0 T 0 \SLOWLLSH8)
        (226 LRSH1 0 T 0 \SLOWLRSH1)
        (227 LRSH8 0 T 0 \SLOWLRSH8)
        (228 LOGOR2 0 T -1 \SLOWLOGOR2)
        (229 LOGAND2 0 T -1 \SLOWLOGAND2)
        (230 LOGXOR2 0 T -1 \SLOWLOGXOR2)
        (231 LSH 0 T -1 LSH T)
        (232 FPLUS2 0 T -1 \SLOWFPLUS2 4K)
        (233 FDIFFERENCE 0 T -1 \SLOWFDIFFERENCE 4K)
        (234 FTIMES2 0 T -1 \SLOWFTIMES2 4K)
        (235 FQUOTIENT 0 T -1 \SLOWFQUOTIENT 4K)
        (236 UBFLOAT2 1 (UFADD UFSUB UFISUB UFMULT UFDIV UFGREAT UFMAX UFMIN UFREM UBAREF1)
             (-1 1)
             \UNBOXFLOAT2
             (4K DORADO))
        (237 UBFLOAT1 1 (BOX UNBOX UFABS UFNEGATE UFIX)
             (0 1)
             \UNBOXFLOAT1
             (4K DORADO))
        (238 AREF2 0 T -2 %%AREF2 (4K DORADO))
        (239 ASET2 0 T -3 %%ASET2 (4K DORADO))
        (240 EQ 0 T -1)
        (241 IGREATERP 0 T -1 \SLOWIGREATERP)
        (242 FGREATERP 0 T -1 \SLOWFGREATERP)
        (243 GREATERP 0 T -1 GREATERP)
        (244 EQUAL 0 T -1 EQUAL)
        (245 MAKENUMBER 0 T -1 \MAKENUMBER 4K)
        (246 BOXIPLUS 0 T -1 \BOXIPLUS 4K)
        (247 BOXIDIFFERENCE 0 T -1 \BOXIDIFFERENCE 4K)
        (248 FLOATBLT 1 (FLOATWRAP FLOATUNWRAP FLOAT FIX FPLUS FDIFFERENCE FDIFFERENCE FPLUSABS ABSDIFFERENCE
                         ABSFPLUS FTIMES)
             -3 \FLOATBLT (4K DORADO))
        (249 FFTSTEP 0 T -1 \FFTSTEP (4K DORADO))
        (250 MISC3 1 (EXPONENT MAGNITUDE FLOAT COMP BLKFMAX BLKFMIN BLKFABSMAX BLKFABSMIN FLOATTOBYTE ARRAYREAD
                      LINES-EQUAL-P)
             -2 \MISC3.UFN (4K DORADO))
        (251 MISC4 1 (ARRAY.TIMES ARRAY.PERM ARRAY.PLUS ARRAY.DIFFERENCE ARRAY.MAGIC 3MATCH BMBIT ARRAYWRITE)
             -3 \MISC4.UFN)
        (252 UPCTRACE 0 T 0 NILL (4K 12K))
        (253 SWAP 0 T 0)
        (254 NOP 0 T 0)
        (255 = 0 T -1 CL::%%= (4K DORADO))))

(ADDTOVAR \OPCODEARRAY )

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \OPCODEARRAY \OPCODES)
)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DEFINEQ
```

(**WORDSPERNAMEENTRY**

```
  [LAMBDA NIL                                                    ; Edited 25-Jan-91 00:00 by jds
    ;; Run-time equivalent of the optimizer; this must match it in result (but use COMPILER::*TARGET-ARCHITECTURE* here where you'd use
    ;; (COMPILER::ENV-TARGET-ARCHITECTURE ENV) there).
    (COND
```

```
       ((FMEMB :3-BYTE COMPILER::*TARGET-ARCHITECTURE*)
        2)
       ((AND CROSSCOMPILING (FMEMB :3-BYTE-INIT COMPILER::*TARGET-ARCHITECTURE*))
        2)
       (T 1])

)

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

(PUTPROPS DPUTCODE MACRO ((FN CA SIZE)
                          (SELECTQ (SYSTEMTYPE)
                               (D (DEFC FN CA))
                               (/PUTPROP FN 'DCODE CA))))

(PUTPROPS MCODEP MACRO [(X)
                        (OR (ARRAYP X)
                            (AND (LITATOM X)
                                 (ARRAYP (SELECTQ (SYSTEMTYPE)
                                                  (D (GETD X))
                                                  (GETPROP X 'DCODE])
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS CODELT MACRO ((CA N)
                        (\BYTELT CA N)))

(PUTPROPS CODELT2 MACRO [OPENLAMBDA (DEF LC)
                          (LOGOR (LLSH (CODELT DEF LC)
                                       BITSPERBYTE)
                                 (CODELT DEF (ADD1 LC])

(PUTPROPS CODESETA2 MACRO [OPENLAMBDA (DEF LC VALUE)
                           (CODESETA DEF LC (LRSH VALUE BITSPERBYTE))
                           (CODESETA DEF (ADD1 LC)
                                  (IMOD VALUE (CONSTANT (LLSH 1 BITSPERBYTE])

(PUTPROPS CODESETA MACRO ((CA N NV)
                          (\BYTESETA CA N NV)))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS BYTESPERNAMEENTRY MACRO (NIL (UNFOLD (CONSTANT (WORDSPERNAMEENTRY))
                                               BYTESPERWORD)))

(PUTPROPS BYTESPERNTOFFSETENTRY MACRO (NIL (UNFOLD (WORDSPERNAMEENTRY)
                                                   BYTESPERWORD)))

(PUTPROPS GETNAMEENTRY MACRO (OPENLAMBDA (DEF LC)
                              (LET ((NUMBER 0))
                                   ;; Must ALWAYS be called with DEF really being either a FNHEADER or a nametable
                                   ;; pseudo-fnheader.  Never use addbase to offset from it.  This is because CODEBASEELT checks
                                   ;; the BYTESWAPPED flag in the fnheader....
                                   [FOR I FROM 0 TO (CONSTANT (SUB1 (BYTESPERNAMEENTRY)))
                                      DO (SETQ NUMBER (LOGOR (LLSH NUMBER BITSPERBYTE)
                                                             (CODEBASELT DEF (IPLUS LC I]
                                   NUMBER)))

(PUTPROPS GETNTFLAGS MACRO (OPENLAMBDA (DEF LC)
                            (CODEBASELT DEF LC)))

(PUTPROPS GETNTOFFSET MACRO (OPENLAMBDA (DEF LC)
                             (NTSLOT-OFFSET (GETNTOFFSETENTRY DEF LC))))

(PUTPROPS GETNTOFFSETENTRY MACRO (OPENLAMBDA (DEF LC)
                                  (LET ((NUMBER 0))
                                       [for I from 0 to (CONSTANT (SUB1 (BYTESPERNTOFFSETENTRY)))
                                          do (SETQ NUMBER (LOGOR (LLSH NUMBER BITSPERBYTE)
                                                                 (CODEBASELT DEF (IPLUS LC I]
                                       NUMBER)))

(PUTPROPS GETNTTAG MACRO (OPENLAMBDA (DEF LC)
                          (CODEBASELT DEF (ADD1 LC))))

(PUTPROPS SETNAMEENTRY MACRO [OPENLAMBDA (DEF LC VALUE)
                              (FOR I FROM (CONSTANT (SUB1 (BYTESPERNAMEENTRY))) TO 0 BY -1
                                 DO [CODEBASESETA DEF (IPLUS LC I)
                                           (LOGAND VALUE (CONSTANT (SUB1 (LLSH 1 BITSPERBYTE]
                                    (SETQ VALUE (LRSH VALUE BITSPERBYTE])

(PUTPROPS WORDSPERNTOFFSETENTRY MACRO (NIL (WORDSPERNAMEENTRY)))
```

```
(PUTPROPS NTSLOT-OFFSET MACRO ((X)
                               (LOGAND 255 X)))
)
```

```
(DEFMACRO NEW-SYMBOL-CODE (NEW-SYMBOL-FORM OLD-SYMBOL-FORM)
    ;; Use one form or another, depending on whether we're compiling for new 3-byte atoms or old 2-byte atom numbers.
    [COND
       ((FMEMB :3-BYTE COMPILER::*TARGET-ARCHITECTURE*)          ; NEW ATOMS
        `,NEW-SYMBOL-FORM)
       (T `,OLD-SYMBOL-FORM])

(DEFOPTIMIZER BIG-VMEM-CODE (NEW-SYMBOL-FORM OLD-SYMBOL-FORM &ENVIRONMENT ENV)
                            ;; Allow for differences between 4-byte pointers and 3-byte pointers..
                            [COND
                               ((FMEMB :4-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
                                `,NEW-SYMBOL-FORM)
                               (T `,OLD-SYMBOL-FORM])

(DEFOPTIMIZER SETSTKNAMEENTRY (CODEARRAY OFFSET VAL &ENVIRONMENT ENV)
                              ;; Set the name entry for a name-table entry.
                              [COND
                                 [(FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
                                  `(LET ((BASE (fetch (ARRAYP BASE) of ,CODEARRAY))
                                         (VALUE ,VAL))
                                        (COND
                                           ((FIXP VALUE)   ; A 20-byte atom #.  Make it an atom.
                                            (\PUTBASEPTR BASE ,OFFSET (\VAG2 \AtomHI VALUE)))
                                           (T              ; A 3-byte atom.  Just use it.
                                              (\PUTBASEPTR BASE ,OFFSET VALUE]
                                 (T `(LET [(BASE (fetch (ARRAYP BASE) of ,CODEARRAY]
                                          (\PUTBASE BASE ,OFFSET ,VAL])

(DEFOPTIMIZER SETSTKNTOFFSETENTRY (BASE OFFSET VAL &ENVIRONMENT ENV)
                                  ;; Set the offset entry for a name-table entry.
                                  [COND
                                     [(FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
                                      `(\PUTBASEFIXP ,BASE ,OFFSET ,VAL]
                                     (T `(\PUTBASE ,BASE ,OFFSET ,VAL])

(DEFOPTIMIZER GETSTKNAMEENTRY (BASE OFFSET &ENVIRONMENT ENV)
                              ;; Get a name entry out of a name table.  BASE is the start of the name table; OFFSET is in words, not
                              ;; bytes or name entries.
                              [COND
                                 [(FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
                                  `(\GETBASEPTR ,BASE ,OFFSET]
                                 (T `(\GETBASE ,BASE ,OFFSET])

(DEFOPTIMIZER GETSTKNTOFFSETENTRY (BASE OFFSET &ENVIRONMENT ENV)
                                  [COND
                                     [(FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
                                      `(\GETBASEFIXP ,BASE ,OFFSET]
                                     (T `(\GETBASE ,BASE ,OFFSET])

(DEFOPTIMIZER WORDSPERNAMEENTRY (&ENVIRONMENT ENV)
                                ;; Number of words in a name-table "Name" entry--the space for the symbol.  1 for old symbol
                                ;; systems, 2 for 3-byte-atom systesm.
                                [COND
                                   ((FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
                                    `(PROGN 2))
                                   ((AND CROSSCOMPILING (FMEMB :3-BYTE-INIT (
                                                                    COMPILER::ENV-TARGET-ARCHITECTURE
                                                                          ENV)))
                                    `(PROGN 2))
                                   (T `(PROGN 1])

(DEFOPTIMIZER SETSTKNTOFFSET (BASE OFFSET TYPE VAL &ENVIRONMENT ENV)
                             ;; Set the offset entry for a name-table entry, from the symbol to fill in plus the variable-type marker value
                             ;; SHIFTED LEFT 14 BITS ALREADY.
                             [COND
                                [(FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
                                 `(PROGN (\FIXCODENUM ,BASE (IDIFFERENCE ,OFFSET BYTESPERWORD)
                                                      ,TYPE)
```

```
                                        (\FIXCODENUM ,BASE ,OFFSET ,VAL]
                                  (T `(\FIXCODENUM ,BASE ,OFFSET (IPLUS ,TYPE ,VAL])
```

(DEFOPTIMIZER **SETSTKNAME-RAW** (BASE OFFSET VAL &ENVIRONMENT ENV)

;; Set the name entry for a name-table entry. This version works with raw storage, as opposed to
;; SETSTKNAMEENTRY, which works on an ARRAYP.

;; If this optimizer changes, change SETSTKNAMEENTRY as well.

```
[COND
   [(FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
    `(LET ((VALUE ,VAL))
          (COND
             ((FIXP VALUE)    ; A 20-byte atom #.  Make it an atom.
              (\PUTBASEPTR ,BASE ,OFFSET (\VAG2 \AtomHI VALUE)))
             (T               ; A 3-byte atom.  Just use it.
                (\PUTBASEPTR ,BASE ,OFFSET VALUE]
    (T `(\PUTBASE ,BASE ,OFFSET ,VAL])
```

(DEFOPTIMIZER **SETSTKNTOFFSET-RAW** (BASE OFFSET TYPE VAL &ENVIRONMENT ENV)

;; Set the offset entry for a name-table entry. This version works on raw storage, vs
;; SETSTKNAMEOFFSETENTRY, which is supposed to work on codearrays. Any changes here
;; should be made there, as well. TYPE must already be shifted left by 14 bits.

```
   [COND
      [(FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
       `(PROGN (\PUTBASE ,BASE ,OFFSET ,TYPE)
               (\PUTBASE ,BASE (IPLUS ,OFFSET 1)
                         ,VAL]
      (T `(\PUTBASE ,BASE ,OFFSET (IPLUS ,TYPE ,VAL])
```

(DEFOPTIMIZER **NEW-SYMBOL-CODE** (NEW-SYMBOL-FORM OLD-SYMBOL-FORM &ENVIRONMENT ENV)

;; Allow for differences between 3-byte atoms and 2-byte atoms.

```
[COND
   ((FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
    `,NEW-SYMBOL-FORM)
   (T `,OLD-SYMBOL-FORM])
```

(DEFOPTIMIZER **MAKE-NTENTRY** (TYPE OFFSET &ENVIRONMENT ENV)

```
[COND
   [(FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
    `(IPLUS (CONSTANT (LLSH ,TYPE 16))
            ,OFFSET]
   (T `(IPLUS (CONSTANT ,TYPE)
              ,OFFSET])
```

(DEFOPTIMIZER **NULL-NTENTRY** (VALUE &ENVIRONMENT ENV)

;; Predicate: Is VALUE a null entry in a name table? I.e., does it result from fetching the entry at the end
;; that's all zeros? For 2-byte atoms, that's the same as being zero. For 3-byte atoms, it's the same as being
;; NIL.

```
[COND
   [(FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
    `(NULL ,VALUE]
   (T `(EQ ,VALUE 0])
```

(DEFOPTIMIZER **NTSLOT-VARTYPE** (X &ENVIRONMENT ENV)

;; Given the contents of a name-table Offset entry, return the variable-type bits at the top of the entry. THE
;; RESULT IS RETURNED SHEFTED LEFT 14 BITS, THE USUAL REPRESENTATION.

```
[COND
   [(FMEMB :3-BYTE (COMPILER::ENV-TARGET-ARCHITECTURE ENV))
    `(LOGAND 49153 (LRSH ,X 16]
   (T `(LOGAND ,X 49152])

(DECLARE%: EVAL@COMPILE

[ACCESSFNS CODEARRAY ((STKMIN (CODELT2 DATUM 0)
                              (CODESETA2 DATUM 0 NEWVALUE))
                      (NA (SIGNED (CODELT2 DATUM 2)
                                  BITSPERWORD)
                          (CODESETA2 DATUM 2 (UNSIGNED NEWVALUE BITSPERWORD)))
                      (PV (SIGNED (CODELT2 DATUM 4)
                                  BITSPERWORD)
                          (CODESETA2 DATUM 4 (UNSIGNED NEWVALUE BITSPERWORD)))
                      (STARTPC (CODELT2 DATUM 6)
                               (CODESETA2 DATUM 6 NEWVALUE))
                      [ARGTYPE (LOGAND (LRSH (CODELT DATUM 8)
                                             4)
                                       3)
                               (CODESETA DATUM 8 (LOGOR (LOGAND (CODELT DATUM 8)
```

```
                                                                207)
                                                          (LLSH (LOGAND NEWVALUE 3)
                                                                4]
                        (FRAMENAME (\VAG2 (LOGAND (CODELT2 DATUM 8)
                                                  4095)
                                          (CODELT2 DATUM 10))
                                   (\FIXCODEPTR DATUM 11 (EVQ NEWVALUE)))
                        (NTSIZE (CODELT2 DATUM 12)
                                (CODESETA2 DATUM 12 NEWVALUE))
                        (NLOCALS (CODELT DATUM 14)
                                (CODESETA DATUM 14 NEWVALUE))
                        (FVAROFFSET (CODELT DATUM 15)
                                (CODESETA DATUM 15 NEWVALUE)))
        (ACCESSFNS CODEARRAY ((LSTARP (ILESSP (fetch (CODEARRAY NA) of DATUM)
                                              0))
                              (OVERHEADWORDS (PROGN 8))
                              (ALIGNED (IPLUS (fetch (CODEARRAY NTSIZE) of DATUM)
                                              (fetch (CODEARRAY OVERHEADWORDS) of T)))
                              (FIXED NIL (replace (CODEARRAY STKMIN) of DATUM with (\STKMIN DATUM)))
                              (FRAMENAME# (PROGN 8]
)

(DECLARE%: EVAL@COMPILE

(RECORD OPCODE (OP# OPCODENAME OPNARGS OPPRINT LEVADJ UFNFN UNIMPL))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \OPCODES)
)

(DECLARE%: EVAL@COMPILE

(RPAQQ PVARCODE 32768)

(RPAQQ FVARCODE 49152)

(RPAQQ IVARCODE 0)

(RPAQQ VARCODEMASK 49152)

(CONSTANTS PVARCODE FVARCODE IVARCODE VARCODEMASK)
)

(DECLARE%: EVAL@COMPILE

(RPAQQ \NT.IVARCODE 0)

(RPAQQ \NT.PVARCODE 2)

(RPAQQ \NT.FVARCODE 3)

(CONSTANTS \NT.IVARCODE \NT.PVARCODE \NT.FVARCODE)
)
)

;; END EXPORTED DEFINITIONS

;; ufns

(DEFINEQ

(INITUFNTABLE
  [LAMBDA NIL                                                    ; Edited 19-Nov-92 15:28 by sybalsky:mv:envos
    (CREATEPAGES \UFNTable \UFNTableSize NIL T)
    (for I from 0 to 255 do (\SETUFNENTRY I '\UNKNOWN.UFN 0 0))
    (for X in \OPCODES when (fetch (OPCODE UFNFN) of X)
       do (\SETUFNENTRY (PROG ((OP (fetch (OPCODE OP#) of X)))
                              (RETURN (if (LISTP OP)
                                          then (CAR OP)
                                          else OP)))
                (fetch (OPCODE UFNFN) of X)
                [COND
                   ((LISTP (fetch (OPCODE LEVADJ) of X))
                    (CADR (fetch (OPCODE LEVADJ) of X)))
                   (T (IDIFFERENCE (IPLUS 1 (COND
                                              ((EQ (fetch (OPCODE OPNARGS) of X)
                                                   0)
                                               0)
                                              (T 1)))
                                   (fetch (OPCODE LEVADJ) of X]
                (SELECTQ (fetch (OPCODE OPNARGS) of X)
                     (0 0)
                     (1 1)
                     (2 2)
                     (3 ;; Changed by JDS from (3 1) 12/26/90 as part of 3-byte-atom change.  I dunno why this was (3 1), since there were no
```

```
                        ;; UFNing args with 3-byte args???
                        3)
                      (4 4)
                      (5 5)
                      (SHOULDNT])
```

(\SETUFNENTRY

```
  [LAMBDA (INDEX FN NARGS NEXTRA)                                    (* lmm " 7-Jun-85 14:08")
    (SETQ INDEX (\ADDBASE (\ADDBASE \UFNTable INDEX)
                          INDEX))
    (change (fetch (UFNENTRY FNINDEX) of INDEX)
            (\ATOMDEFINDEX FN))
    (change (fetch (UFNENTRY NEXTRA) of INDEX)
            NEXTRA)
    (change (fetch (UFNENTRY NARGS) of INDEX)
            NARGS])
```

(\GETUFNENTRY

```
  [LAMBDA (OP)                                                       (* hdj "17-Jun-85 13:08")
    (LET [(INDEX (\ADDBASE2 \UFNTable (if (LITATOM OP)
                                          then (fetch (OPCODE OP#) of (\FINDOP OP))
                                        else OP]
          (\VAG2 0 (fetch (UFNENTRY FNINDEX) of INDEX])

)

(DEFINEQ
```

(\UNKNOWN.UFN

```
  [LAMBDA NIL                                                        (* bvm%: "23-Mar-84 15:52")
    (\MP.ERROR \MP.UNKNOWN.UFN "Compiler/microcode error: unknown UFN"])

)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(RPAQ? COMPILER::*TARGET-ARCHITECTURE* '(:4-BYTE :3-BYTE))

(RPAQ? COMPILER::*HOST-ARCHITECTURE* '(:4-BYTE :3-BYTE))
)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(BLOCKRECORD UFNENTRY (;; Describes a 32-bit entry in the "UFN Table", \UFNTable, which is used to find the function to call when an opcode isn't
                       ;; implemented in microcode.
                       (FNINDEX WORD)                                ; LO 16 bits of the symbol # for the UFN
                       (NEXTRA BYTE)                                 ; [Previously unused] HI 8 bits of the UFN symbol number.
                       (NARGS BYTE)                                  ; # of arguments to the UFN.

                       ))
)

(ADDTOVAR INEWCOMS (FNS INITUFNTABLE \SETUFNENTRY))


(ADDTOVAR DONTCOMPILEFNS INITUFNTABLE)
)

;; for MAKEINIT and READSYS

(DECLARE%: DONTCOPY

(ADDTOVAR INEWCOMS
          (FNS DCODERD)
          [VARS \OPCODES (CODERDTBL (COPYREADTABLE 'ORIG]
          (P (SETSYNTAX (CHARCODE ^Y)
                        '[MACRO (LAMBDA (FILE RDTBL)
                                        (EVALFORMAKEINIT (READ FILE RDTBL]
                        CODERDTBL)
             (SETSYNTAX (CHARCODE %|)
                        '(MACRO ALWAYS READVBAR)
                        CODERDTBL)
             (READTABLEPROP CODERDTBL 'USESILPACKAGE NIL)))

(ADDTOVAR MKI.SUBFNS (\CODEARRAY . SCRATCHARRAY)
                     (DPUTCODE . I.PUTDEFN)
                     (CODERDTBL . I.CODERDTBL)
                     (SETSTKNTOFFSET . I.SETSTKNTOFFSET)
                     (WORDSPERNAMEENTRY . I.WORDSPERNAMEENTRY))

(ADDTOVAR EXPANDMACROFNS CODELT CODELT2 CODESETA CODESETA2 DPUTCODE MCODEP BYTESPERNAMEENTRY
```

```
                                    BYTESPERNTOFFSETENTRY WORDSPERNAMEENTRY)

(ADDTOVAR RD.SUBFNS (CODELT . VGETBASEBYTE)
                    (CODESETA . VPUTBASEBYTE))

(ADDTOVAR RDCOMS (FNS \GET-COMPILED-CODE-BASE))
)

(PUTPROPS LLCODE FILETYPE CL:COMPILE-FILE)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR NLAMA OP#)

(ADDTOVAR NLAML )

(ADDTOVAR LAMA )
)

(PUTPROPS LLCODE COPYRIGHT ("Venue & Xerox Corporation" 1981 1982 1983 1984 1985 1986 1987 1988 1990 1991 1992
                                    1993))
```

FUNCTION INDEX

MACRO INDEX

OPTIMIZER INDEX

VARIABLE INDEX

CONSTANT INDEX

RECORD INDEX

PROPERTY INDEX