

X85s

brexpr.x86
expr.x85
pseud2.x86

cmsubr.x86
instr.x85
symb.x86

dline.x86
pseud1.x86



\$nolist

/*

modified 4/13/81 R. Silberstein

*/

/* Symbol types : */

```

dcl
reg      lit      '0',    /* register */
pseudo   lit      '1',    /* pseudo instruction */
code     lit      '2',    /* instruction */
string   lit      '3',    /* character string */
spec     lit      '4',    /* special character */
number   lit      '5',    /* 8 or 16 bit number */
variable lit      '6',
lab       lit      '7',    /* label */
operator lit      '8',    /* operator in expressions */
doubleddefined lit    '0f9h', /* doubled defined symbol */
neglected lit    '0fah', /* neglected symb., never to be def. */
ident     lit    '0fbh', /* identifier, scanner output */
undefsymb lit    '0fdh', /* undefined symbol */
symbol    lit    '0feh', /* variable, label or undef. symb. */
deleted symb lit    '0ffh'; /* deleted symbol (not used) */

```

/* Symbol description values */

```

dcl
nil      lit      '0',    /* no specification */
byt      lit      '1',    /* symbol is 8-bit type */
word     lit      '2',    /* symbol is 16 bit type */
dword    lit      '4',    /* symbol is 2*16 bit type
                        or a segment register */

```

/* Register values : */

```

dcl
rbx      lit      '3',
rbp      lit      '5',
rsi      lit      '6',
rdi      lit      '7',

res      lit      '0',    /* segment registers */
rcs      lit      '1',
rss      lit      '2',
rds      lit      '3';

```

/* Symbolic operators */

```

dcl
oshort   lit      '0',    /* 8-bit value of expression */
oor       lit      '1',    /* logical OR */
oxor     lit      '2',    /* logical XOR */
oand     lit      '3',    /* logical AND */

```

CCP/M-86 2.0 V.3

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # CC-104

```

onot      lit '4',      /* logical NOT */
oeq       lit '5',      /* equal */
ogt       lit '6',      /* greater */
oge       lit '7',      /* greater or equal */
olt       lit '8',      /* less */
ole       lit '9',      /* less or equal */
one       lit '10',     /* not equal */
omod      lit '11',     /* arithmetic MOD */
oshl      lit '12',     /* shift left */
oshr      lit '13',     /* shift right */
optr      lit '14',     /* take type of 1. op, value of 2. */
offset    lit '15',     /* offset value of operand */
oseg      lit '16',     /* segment value of operand */
otype     lit '17',     /* type value of operand */
olength   lit '18',     /* length attribute of variables */
olast     lit '19',     /* length - 1 */
leftbracket lit '[[',
rightbracket lit ']]';

```

```
dcl
```

```

operandstruc  lit  "struc(
length        addr,
stype         byte,
sflag         byte,
segment       addr,
offset        addr,
baseindex     byte)",

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

```
/* define bits of SFLAG of structures above */
```

```

typesbit    lit '7h', /* bit 0-2 */
segtypesbit lit '18h', /* bit 3-4 */
segmbit     lit '20h', /* bit 5 */
iregbit     lit '40h', /* bit 6 */
bregbit     lit '80h', /* bit 7 */

```

```
/* left-shift counters */
```

```

typecount   lit '0',
segtypecount lit '3',
segmcount   lit '5',
iregcount   lit '6',
bregcount   lit '7',

```

```
/* define bits of BASEINDEX byte of structures above */
```

```

indexregbit lit '01h', /* bit 0 */
baseregbit  lit '02h', /* bit 1 */
nooverridebit lit '40h', /* bit 6 */

```

```
/* left shift counters */
```

```

indexregcount lit '0',
baseregcount  lit '1',
noovercount   lit '6';

```

```
scan: proc external;
```

20-APR-1983 18:05

Page 3

BREXPR.X86;3

end scan;

specialtoken: proc (tok) byte external;
dcl tok byte;
end specialtoken;

\$list

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

\$nolist

/*

modified 4/13/81 R. Silberstein
modified 9/2/81 R. Silberstein

*/

/* Error numbers: */

dcl
laboutofrange lit '22', /* label out of range */
misssegminfo lit '23'; /* missing segment info in operand */

/* Structures: */

dcl

symbolstruc lit 'struct(
length addr,
stype byte,
sflag byte,
segment addr,
offset addr,
baseindex byte)';
operandstruc lit 'symbolstruc';

/* define bits of SFLAG of structures above */

dcl
type\$bit lit '7h', /* bit 0-2 */
segtypebit lit '18h', /* bit 3-4 */
segmbit lit '20h', /* bit 5 */
iregbit lit '40h', /* bit 6 */
bregbit lit '80h', /* bit 7 */

/* left-shift counters */

typecount lit '0',
segtypecount lit '3',
segmcount lit '5',
iregcount lit '6',
bregcount lit '7',

/* define bits of BASEINDEX byte of structures above */

indexregbit lit '01h', /* bit 0 */
baseregbit lit '02h', /* bit 1 */
nooverridebit lit '40h', /* bit 6 */

/* left shift counters */

indexregcount lit '0',
baseregcount lit '1',
noovercount lit '6';

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950

SER. # _____

/* Miscellaneous global variables: */

```
del
ABSADDR(4)  BYTE EXTERNAL, /* ABSOLUTE ADDRESS FIELD */
cip         addr external, /* current instruction pointer */
csegvalue   addr external, /* current segment value */
noerror     byte external, /* errorflag in codemacro decoding */
firstmacroptr address external, /* pointer at first codemacro */
macroptr    address external, /* current pointer within macros */
fullsymtab  byte external, /* full if symboltable is full */
nooper      byte external, /* no of instruction operands */
operands(4) operandstruc /* instruction operands, max 4 */
              external;

$list
```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

\$nolist

dcl
pass byte external, /* current pass no, 1,2,3 */

prefix (240) byte external, /* prefix to source line */
prefixptr byte external, /* pointer to prefix buffer */

accumsave(80) byte external,
acclensave byte external,

/* Miscellaneous variables: */

fullsymtab byte external, /* full if symboltable is full */
currentsymbol symbolstruc /* current scanned symbol */
 external,
symtabadr address external, /* pointer at symbol in table */
codemacroptr address external, /* pointer to found codemacro */
codemacro\$flag byte external; /* true if building a codemacro */

\$list

COPYRIGHT © 1981, 1982, 1983
DIGITAL RESEARCH
P. O. BOX 579
PACIFIC GROVE, CA 93950
SER. # _____

\$nolist

/*

modified 4/24/81 R. Silberstein

*/

/* Symbol types : */

```

dcl
reg      lit      "0",    /* register */
pseudo   lit      "1",    /* pseudo instruction */
code     lit      "2",    /* instruction */
string   lit      "3",    /* character string */
spec     lit      "4",    /* special character */
number   lit      "5",    /* 8 or 16 bit number */
variable lit      "6",
lab      lit      "7",    /* label */
operator lit      "8",    /* operator in expressions */
doubleddefined lit    "0f9h", /* doubled defined symbol */
neglected lit    "0fah", /* neglected symb., never to be def. */
ident    lit      "0fbh", /* identifier, scanner output */
undefsymb lit    "0fdh", /* undefined symbol */
symbol   lit      "0feh", /* variable, label or undef. symb. */
deleted symb lit    "0ffh"; /* deleted symbol (not used) */

```

/* Symbol description values */

```

dcl
nil      lit      "0",    /* no specification */
byt      lit      "1",    /* symbol is 8-bit type */
wr      lit      "2",    /* symbol is 16 bit type */
dwr      lit      "4",    /* symbol is 2*16 bit type
                        or a segment register */

```

/* Register values : */

```

dcl
rbx      lit      "3",
rbp      lit      "5",
rsi      lit      "6",
rdi      lit      "7",

res      lit      "0",    /* segment registers */
rcs      lit      "1",
rss      lit      "2",
rds      lit      "3";

```

/* Symbolic operators */

```

dcl
oshort   lit      "0",    /* 8-bit value of expression */
oor      lit      "1",    /* logical OR */
oxor     lit      "2",    /* logical XOR */
oand     lit      "3",    /* logical AND */
onot     lit      "4",    /* logical NOT */

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 573

PACIFIC GROVE, CA 93950

SER. # _____

```

oeq      lit "5",      /* equal */
ogt      lit "6",      /* greater */
oge      lit "7",      /* greater or equal */
olt      lit "8",      /* less */
ole      lit "9",      /* less or equal */
one      lit "10",     /* not equal */
omod     lit "11",     /* arithmetic MOD */
oshl     lit "12",     /* shift left */
oshr     lit "13",     /* shift right */
optr     lit "14",     /* take type of 1. op, value of 2. */
ooffset  lit "15",     /* offset value of operand */
oseg     lit "16",     /* segment value of operand */
otype    lit "17",     /* type value of operand */
olength  lit "18",     /* length attribute of variables */
olast    lit "19",     /* length - 1 */
leftbracket lit "[",
rightbracket lit "];

```

dcl

```

operandstruc  lit "struct(
length        addr,
stype         byte,
sflag         byte,
segment       addr,
offset        addr,
baseindex     byte);

```

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

/* define bits of SFLAG of structures above */

```

typesbit      lit "7h", /* bit 0-2 */
segtypebit    lit "18h", /* bit 3-4 */
segmbit       lit "20h", /* bit 5 */
iregbit       lit "40h", /* bit 6 */
bregbit       lit "80h", /* bit 7 */

```

/* left-shift counters */

```

typecount     lit "0",
segtypecount  lit "3",
segmcount     lit "5",
iregcount     lit "6",
bregcount     lit "7",

```

/* define bits of BASEINDEX byte of structures above */

```

indexregbit    lit "7", /* bit 0-2 */
baseregbit     lit "38h", /* bit 3-5 */
nooveridebit   lit "40h", /* bit 6 */

```

/* left shift counters */

```

indexregcount  lit "0",
baseregcount   lit "3",
noovercount    lit "6",

```

dcl

undefsymbol . lit "10"; /* undefined elem. of expression */

newsymbol: proc(lg, stradr, result) byte external;
 dcl lg byte, (stradr, result) addr;
end newsymbol;

findsymbol: proc(lg, stradr, result) byte external;
 dcl lg byte, (stradr, result) addr;
end findsymbol;

getattributes: proc(symbadr, dest) external;
 dcl (symbadr, dest) addr;
end getattributes;

enterattributes: proc(symbadr, source) external;
 dcl (symbadr, source) addr;
end enterattributes;

scan: proc external;
end scan;

specialtoken: proc(tok) byte external;
 dcl tok byte;
end specialtoken;

bracketexpr: proc(pt) byte external;
 dcl pt address;
end bracketexpr;

\$list

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

\$nolist

dcl

acclen byte external, /* accumulator length */
accum(80) byte external, /* actual token scanned */

/* Miscellaneous variables: */

noerror byte external, /* errorflag in codemacro decoding */
firstmacroptr address external, /* pointer at first codemacro */
macroptr address external, /* current pointer within macros */
nooper byte external, /* no of instruction operands */
operands(4) operandstruc /* instruction operands,max 4 */
 external,
codemacroptr address external; /* pointer to found codemacro */

\$list

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

PSEUD1.X86;3

20-APR-1983 18:05

Page 1

\$nolist

/*

modified 4/24/81 R. Silberstein

*/

```
dcl
end$of$line$err      lit '6',    /* garabage at end of line */
pseudooperr          lit '11',   /* illegal pseudo operand */
illexprelem          lit '20',   /* illegal expression element */
missegminfo          lit '23';  /* missing segment info in operand */
```

\$list

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

\$nolist

/*

modified 7/24/81 R. Silberstein

*/

dcl

pass byte external, /* current pass no, 1,2,3 */

/* print output parameters */

```

print$on      byte external, /* on/off flag */
print$switchoff byte external, /* set/reset by NOLIST/LIST */
IFLIST        BYTE EXTERNAL, /* SET/RESET BY IFLIST/NOIFLIST */
maxcol        byte external, /* pagewidth */
title (30)    byte external, /* user specified program title */
pagesize      byte external, /* page size */
simform       byte external, /* true if formfeed simulation */
sourceptr     byte external, /* source buffer pointer */

```

/* scanner variables: */

```

token struct( /* actual token scanned */
type      byte, /* token type, legal values :
                reg      - register
                pseudo   - pseudo code
                string    - text string
                spec     - special character
                number    - number
                operator  - arithmetic operator
                ident     - identifier */

```

```

descr      byte, /* token description, legal values:
                nil      - no specification
                byte     - 8 bit type
                word     - 16 bit type
                dword    - 32 bit type */

```

```

value      addr) external, /* token value */
nextch     byte external, /* next input character */
acclen     byte external, /* accumulator length */
accum(80)  byte external, /* actual token scanned */
eofset     byte external, /* true if end-of-file found */

```

/* Miscellaneous variables: */

```

include$on      byte external, /* true if INCLUDEfile input */
IFLEVEL         BYTE EXTERNAL, /* IF-ENDIF NESTING LEVEL */
operands(4)     operandstruct /* instruction operands,max 4 */
                external,
include$default byte external; /* default drive for include file */

$list

```

COPYRIGHT © 1981, 1982, 1983
 DIGITAL RESEARCH
 P. O. BOX 579
 PACIFIC GROVE, CA 93950

SER # _____

SYMB.X86:3

20-APR-1983 18:05

Page 1

\$nolist

declare endbuf address external;

\$list

COPYRIGHT © 1981, 1982, 1983

DIGITAL RESEARCH

P. O. BOX 579

PACIFIC GROVE, CA 93950

SER. # _____

