

```
-- file: OpTable.mesa
-- edited by Sandman, July 17, 1977 8:51 PM
```

DIRECTORY

```
AltoDefs: FROM "altodefs",
InlineDefs: FROM "inlinedefs",
ListerDefs: FROM "listerdefs";
```

DEFINITIONS FROM AltoDefs, InlineDefs;

```
OpTable: PROGRAM EXPORTS ListerDefs =
BEGIN
```

```
P: TYPE = PRIVATE RECORD [s0, s1: Q];
```

```
Q: TYPE = PRIVATE RECORD [push, pop: [0..7], length: [0..3]];
```

```
OpParms: PRIVATE ARRAY [0..400B/2) OF P = [
```

```
--      NOOP,      NOOP,      NOOP,      NOOP,
P[Q[0,0,0], Q[0,0,0]], P[Q[0,0,0], Q[0,0,0]],
--      NOOP,      NOOP,      NOOP,      NOOP,
P[Q[0,0,0], Q[0,0,0]], P[Q[0,0,0], Q[0,0,0]],
--      LG0,      LG1,      LG2,      LG3,
P[Q[1,0,1], Q[1,0,1]], P[Q[1,0,1], Q[1,0,1]],
--      LG4,      LG5,      LG6,      LG7,
P[Q[1,0,1], Q[1,0,1]], P[Q[1,0,1], Q[1,0,1]],
--      LGB,      LGS,      SG0,      SG1,
P[Q[1,0,2], Q[1,1,1]], P[Q[0,1,1], Q[0,1,1]],
--      SG2,      SG3,      SGB,      SGS,
P[Q[0,1,1], Q[0,1,1]], P[Q[0,1,2], Q[0,2,1]],
--      LL0,      LL1,      LL2,      LL3,
P[Q[1,0,1], Q[1,0,1]], P[Q[1,0,1], Q[1,0,1]],
--      LL4,      LL5,      LL6,      LL7,
P[Q[1,0,1], Q[1,0,1]], P[Q[1,0,1], Q[1,0,1]],
--      LLB,      LLS,      SL0,      SL1,
P[Q[1,0,2], Q[1,1,1]], P[Q[0,1,1], Q[0,1,1]],
--      SL2,      SL3,      SL4,      SL5,
P[Q[0,1,1], Q[0,1,1]], P[Q[0,1,1], Q[0,1,1]],
--      SL6,      SL7,      SLB,      SLS,
P[Q[0,1,1], Q[0,1,1]], P[Q[0,1,2], Q[0,2,1]],
--      LI0,      LI1,      LI2,      LI3,
P[Q[1,0,1], Q[1,0,1]], P[Q[1,0,1], Q[1,0,1]],
--      LI4,      LI5,      LI6,      LIN1,
P[Q[1,0,1], Q[1,0,1]], P[Q[1,0,1], Q[1,0,1]],
--      LIB,      LIW,      LINB,      NOOP,
P[Q[1,0,2], Q[1,0,3]], P[Q[1,0,2], Q[0,0,0]],
--      LGDB,      LGDS,      SGDB,      SGDS,
P[Q[2,0,2], Q[2,1,1]], P[Q[0,2,2], Q[0,3,1]],
--      LLDB,      LLDS,      SLDB,      SLDS,
P[Q[2,0,2], Q[2,1,1]], P[Q[0,2,2], Q[0,3,1]],
--      R0,      R1,      R2,      R3,
P[Q[1,1,1], Q[1,1,1]], P[Q[1,1,1], Q[1,1,1]],
--      R4,      RB,      W0,      W1,
P[Q[1,1,1], Q[1,1,2]], P[Q[0,2,1], Q[0,2,1]],
--      W2,      WB,      RF,      WF,
P[Q[0,2,1], Q[0,2,2]], P[Q[1,1,3], Q[0,2,3]],
--      RDB,      RDO,      WDB,      WDO,
```

```

P[Q[2,1,2], Q[2,1,1]], P[Q[0,3,2], Q[0,3,1]],
-- RSTR, WSTR, RXL0, RXL1,
P[Q[1,2,2], Q[0,3,2]], P[Q[1,1,1], Q[1,1,1]],
-- RXL2, RXL3, WXL0, RIG0,
P[Q[1,1,1], Q[1,1,1]], P[Q[0,2,1], Q[1,0,1]],
-- RIG1, RIG2, RIG3, WIG0,
P[Q[1,0,1], Q[1,0,1]], P[Q[1,0,1], Q[0,1,1]],
-- RIL0, RIL1, RIL2, RIL3,
P[Q[1,0,1], Q[1,0,1]], P[Q[1,0,1], Q[1,0,1]],
-- WILO, WSO, WSB, WSF,
P[Q[0,1,1], Q[0,2,1]], P[Q[0,2,2], Q[0,2,3]],
-- WSDB, WSDS, NOOP, NOOP,
P[Q[0,3,2], Q[0,3,1]], P[Q[0,0,0], Q[0,0,0]],
-- J1E, J2E, J3E, J4E,
P[Q[0,0,1], Q[0,0,1]], P[Q[0,0,1], Q[0,0,1]],
-- J10, J20, J30, J40,
P[Q[0,0,1], Q[0,0,1]], P[Q[0,0,1], Q[0,0,1]],
-- JBE, JBO, JWE, JWO,
P[Q[0,0,2], Q[0,0,2]], P[Q[0,0,3], Q[0,0,3]],
-- NJBE, NJBO, JEQ1E, JEQ2E,
P[Q[0,0,2], Q[0,0,2]], P[Q[0,2,1], Q[0,2,1]],
-- JEQ3E, JEQ4E, JEQ10, JEQ20,
P[Q[0,2,1], Q[0,2,1]], P[Q[0,2,1], Q[0,2,1]],
-- JEQ30, JEQ40, JEQBE, JEQBO,
P[Q[0,2,1], Q[0,2,1]], P[Q[0,2,2], Q[0,2,2]],
-- JNE1E, JNE2E, JNE3E, JNE4E,
P[Q[0,2,1], Q[0,2,1]], P[Q[0,2,1], Q[0,2,1]],
-- JNE10, JNE20, JNE30, JNE40,
P[Q[0,2,1], Q[0,2,1]], P[Q[0,2,1], Q[0,2,1]],
-- JNEBE, JNEBO, JLBE, JLBO,
P[Q[0,2,2], Q[0,2,2]], P[Q[0,2,2], Q[0,2,2]],
-- JGEBE, JGEB0, JGBE, JGBO,
P[Q[0,2,2], Q[0,2,2]], P[Q[0,2,2], Q[0,2,2]],
-- JLEBE, JLEBO, JULBE, JULBO,
P[Q[0,2,2], Q[0,2,2]], P[Q[0,2,2], Q[0,2,2]],
-- JUGEBE, JUGEB0, JUGBE, JUGBO,
P[Q[0,2,2], Q[0,2,2]], P[Q[0,2,2], Q[0,2,2]],
-- JULEBE, JULEBO, JZQEBE, JZQEBO,
P[Q[0,2,2], Q[0,2,2]], P[Q[0,1,2], Q[0,1,2]],
-- JZNEBE, JZNEBO, JDEQEBE, JDEQEBO,
P[Q[0,1,2], Q[0,1,2]], P[Q[0,4,2], Q[0,4,2]],
-- JDNEBE, JDNEBO, JIB, JIW,
P[Q[0,4,2], Q[0,4,2]], P[Q[0,2,3], Q[0,2,3]],
-- NOOP, NOOP, NOOP, NOOP,
P[Q[0,0,0], Q[0,0,0]], P[Q[0,0,0], Q[0,0,0]],
-- ADD, SUB, MUL, DBL,
P[Q[1,2,1], Q[1,2,1]], P[Q[1,2,1], Q[1,1,1]],
-- DIV, LDIV, NEG, INC,
P[Q[2,2,1], Q[2,3,1]], P[Q[1,1,1], Q[1,1,1]],
-- AND, OR, XOR, SHIFT,
P[Q[1,2,1], Q[1,2,1]], P[Q[1,2,1], Q[1,2,1]],

```

```

--      UCOMP,      NOOP,      NOOP,      NOOP,
P[Q[1,2,1]], Q[0,0,0]], P[Q[0,0,0]], Q[0,0,0]],
--      PUSH,      POP,      EXCH,      PUSHX,
P[Q[1,0,1]], Q[0,1,1]], P[Q[2,2,1]], Q[1,0,1]],
--      DUP,      NOOP,      NOOP,      NOOP,
P[Q[2,1,1]], Q[0,0,0]], P[Q[0,0,0]], Q[0,0,0]],
--      GFC0,      GFC1,      GFC2,      GFC3,
P[Q[0,0,1]], Q[0,0,1]], P[Q[0,0,1]], Q[0,0,1]],
--      GFC4,      GFC5,      GFC6,      GFC7,
P[Q[0,0,1]], Q[0,0,1]], P[Q[0,0,1]], Q[0,0,1]],
--      GFC8,      GFC9,      GFC10,     GFC11,
P[Q[0,0,1]], Q[0,0,1]], P[Q[0,0,1]], Q[0,0,1]],
--      GFC12,     GFC13,     GFC14,     GFC15,
P[Q[0,0,1]], Q[0,0,1]], P[Q[0,0,1]], Q[0,0,1]],
--      GFCB,      LFC1,      LFC2,      LFC3,
P[Q[0,0,2]], Q[0,0,1]], P[Q[0,0,1]], Q[0,0,1]],
--      LFC4,      LFC5,      LFC6,      LFC7,
P[Q[0,0,1]], Q[0,0,1]], P[Q[0,0,1]], Q[0,0,1]],
--      LFC8,      LFC9,      LFC10,     LFC11,
P[Q[0,0,1]], Q[0,0,1]], P[Q[0,0,1]], Q[0,0,1]],
--      LFC12,     LFC13,     LFC14,     LFC15,
P[Q[0,0,1]], Q[0,0,1]], P[Q[0,0,1]], Q[0,0,1]],
--      LFC16,     LFCB,      SFC,      RET,
P[Q[0,0,1]], Q[0,0,2]], P[Q[0,1,1]], Q[0,0,1]],
--      NOOP,      PORTO,     PORTI,     KFCB,
P[Q[0,0,0]], Q[0,1,1]], P[Q[0,0,1]], Q[0,0,2]],
--      ADDL,      ADDG,      BLT,      ALLOC,
P[Q[1,1,1]], Q[1,1,1]], P[Q[0,3,1]], Q[1,1,1]],
--      FREE,      IWDC,      DWDC,      NOOP,
P[Q[0,1,1]], Q[0,0,1]], P[Q[0,0,1]], Q[0,0,0]],
--      STOP,      CATCH,     CONVERT,   BLOCK,
P[Q[0,0,1]], Q[0,0,2]], P[Q[3,5,1]], Q[0,0,1]],
--      BITBLT,    STARTIO,   NOOP,     NOOP,
P[Q[0,1,1]], Q[0,1,1]], P[Q[0,0,0]], Q[0,0,0]],
--      DST,      LST,      LSTF,     NOOP,
P[Q[0,0,2]], Q[0,0,2]], P[Q[0,0,2]], Q[0,0,0]],
--      WR,      RR,      BRK,     NOOP,
P[Q[0,1,2]], Q[1,0,2]], P[Q[0,0,1]], Q[0,0,0]]];

```

```

pushstack: PUBLIC PROCEDURE [b: BYTE] RETURNS [[0..7]] =
BEGIN
RETURN[getQrecord[b].push]
END;

```

```

popstack: PUBLIC PROCEDURE [b: BYTE] RETURNS [[0..7]] =
BEGIN
RETURN[getQrecord[b].pop]
END;

```

```

instlength: PUBLIC PROCEDURE [b: BYTE] RETURNS [[0..3]] =
BEGIN
RETURN[getQrecord[b].length]
END;

```

```
getQrecord: PROCEDURE [b: BYTE] RETURNS [Q] =
  BEGIN
    i, j: CARDINAL;

    [i, j] ← DIVMOD[b, 2];
    RETURN[IF j = 0 THEN OpParms[i].s0 ELSE OpParms[i].s1]
  END;

END .
```