```
0000xxxxxxxxxxxx LODD   AC := M[x]
0001xxxxxxxxxxxx STOD   M[x] := AC
0010xxxxxxxxxxxx ADDD   AC := AC + M[x]
0011xxxxxxxxxxxx SUBD   AC := AC - M[x]
0100xxxxxxxxxxxx JPOS   se AC ≥ 0 então PC := x
0101xxxxxxxxxxxx JZER   se AC = 0 então PC := x
0110xxxxxxxxxxxx JUMP   PC := x
0111xxxxxxxxxxxx LOCO   AC := x
1000xxxxxxxxxxxx LODL   AC := M[SP + x]
1001xxxxxxxxxxxx STOL   M[SP + x] := AC
1010xxxxxxxxxxxx ADDL   AC := AC + M[SP +x]
1011xxxxxxxxxxxx SUBL   AC := AC - M[SP + x]
1100xxxxxxxxxxxx JNEG   se AC < 0 então PC :=  x
1101xxxxxxxxxxxx JNZE   se AC /= 0 então PC :=  x
1110xxxxxxxxxxxx CALL   SP := SP - 1; M[SP] :=  PC;
                        PC :=  x
1111000000000000 PSHI   SP := SP - 1; M[SP] :=M[AC]
1111001000000000 POPI   M[AC] := M[SP]; SP :=SP+1
1111010000000000 PUSH   SP := SP - 1; M[SP] := AC
1111011000000000 POP    AC := M[SP]; SP := SP + 1
1111100000000000 RETN   PC := M[SP]; SP := SP + 1
1111101000000000 SWAP   TMP := AC; AC := SP;
                                SP := TMP
11111100yyyyyyyy INSP   SP := SP + y
11111110yyyyyyyy DESP   SP := SP - y
```