|  |  |  |  |
| --- | --- | --- | --- |
| Machine language | mnemonic | Long name | action |
| 0000 0000 xxxx xxxx | LODD | Load direct | AC 🡨 M[x] |
| 0001 0000 xxxx xxxx | STOD | Store direct | M[x] 🡨 AC |
| 0010 0000 xxxx xxxx | ADDD | Add direct | AC 🡨 AC + M[x] |
| 0011 0000 xxxx xxxx | SUBD | Substract direct | AC 🡨 AC - M[x] |
| 0100 0000 xxxx xxxx | JPOS | Jump positive | If AC >= 0 then PC 🡨 x |
| 0101 0000 xxxx xxxx | JZER | Jump zero | If AC = 0 then PC 🡨 x |
| 0110 0000 xxxx xxxx | JUMP | jump | PC 🡨 x |
| 0111 0000 xxxx xxxx | LOCO | Load constant | AC 🡨 x, x unsigned |
| 1000 0000 xxxx xxxx | LODL | Load local | AC 🡨 M[SP + x] |
| 1001 0000 xxxx xxxx | STOL | Store local | M[x + SP] 🡨 AC |
| 1010 0000 xxxx xxxx | ADDL | Add local | AC 🡨 AC + M[SP + x] |
| 1011 0000 xxxx xxxx | SUBL | Subtract local | AC 🡨 AC - M[SP + x] |
| 1100 0000 xxxx xxxx | JNEG | Jump negative | If AC < 0 then PC 🡨 x |
| 1101 0000 xxxx xxxx | JNZE | Jump nonzero | If AC != 0 then PC 🡨 x |
| 1110 0000 xxxx xxxx | CALL | Call procedure | SP🡨SP-1;M[SP]🡨PC; PC🡨x |
| 1111 0000 0000 0000 | PUSHI | Push indirect | SP🡨SP-1;M[SP]🡨M[AC] |
| 1111 0001 0000 0000 | POPI | Pop indirect | M[AC]🡨M[SP]; SP🡨SP + 1 |
| 1111 0010 0000 0000 | PUSH | push | SP🡨SP-1; M[SP]🡨AC |
| 1111 0011 0000 0000 | POP | pop | AC🡨M[SP]; SP🡨SP + 1 |
| 1111 0100 0000 0000 | RETN | return | PC🡨M[SP]; SP🡨SP + 1 |
| 1111 0101 0000 0000 | SWAP | Swap AC and SP | TMP🡨AC;AC🡨SP;SP🡨TMP |
| 1111 0110 yyyy yyyy | INSP | Increment SP | SP 🡨 SP + y; y unsigned |
| 1111 0111 yyyy yyyy | DESP | Decrement SP | SP 🡨 SP - y; y unsigned |
| 1111 1000 0000 0000 | INPAC | Input AC | AC 🡨 Din |
| 1111 1001 0000 0000 | OUTAC | Output AC | Dout 🡨 AC |
| 1111 1010 0000 0000 | HALT | Stop program |  |