```
aa, aaaa, aaaaa - an address, unsigned integer in range:
aa   : from 00    to 99
aaaa : from 0000  to 9999
aaaaa: from 00000 to 99999
```

```
ss, ssss, sssss - value, signed integer in range:
ss   : from 19 (-9)       to 09 (+9)
ssss : from 1999  (-999)  to 0999 (+999)
sssss: from 19999 (-9999) to 09999 (+9999)
```

The most significant bits (leftmost bit) code value sign:
0: positive value
1: negative value

Flags:
NEGATIVE
ZERO

Flags are set by arithmetic operations: ADD, SUB, MUL, INC, DEC.

Addressing modes:

```
             91... - direct, two byte length
             92... - immediate, one byte length
             93... - immediate, two byte length
             94... - indirect, one byte length
             95... - indirect, two byte length
```

| Mnemonic | Machine code | Meaning |
|---|---|---|
| HLT | 00000 | Stop the cpu. |
| CPA aaaa | 1aaaa | Copy value from memory at address aaaa to accumulator, A := M[aaaa]. |
| CPA (ss) | 921ss | Copy exact value ss to accumulator, A := ss. |
| CPA (sssss) | 93100 sssss | Copy exact value sssss (located in next byte) to accumulator, A := sssss. |
| CPA [aa] | 941aa | Copy value from memory at address given in memory at address aa to accumulator, A := M[M[aa]]. |
| CPA [aaaaa] | 95100 aaaaa | Copy value from memory at address given in memory at address aaaaa to accumulator, A := M[M[aaaaa]]. |
| STO aaaa | 2aaaa | Copy value from accumulator to memory at address aaaa, M[aaaa] := A. |
| STO [aa] | 942aa | Copy value from accumulator to memory at address given in memory at address aa, M[M[aa]] := A. |
| STO [aaaaa] | 95200 aaaaa | Copy value from accumulator to memory at address given in memory at address aaaaa, M[M[aaaaa]] := A. |
| ADD aaaa | 3aaaa | Add value at specified address aaaa to accumulator. Result is stored in accumulator, A := A + M[aaaa]. |
| ADD (ss) | 923ss | Add exact value ss to accumulator. Result is stored in accumulator, A := A + ss. |
| ADD (sssss) | 93300 sssss | Add exact value sssss (located in next byte) to accumulator. Result is stored in accumulator, A := A + sssss. |
| ADD [aa] | 943aa | Add value from memory at address given in memory at address aa to accumulator, A := A + M[M[aa]]. |
| ADD [aaaaa] | 95300 aaaaa | Add value from memory at address given in memory at address aaaaa to accumulator, A := A + M[M[aaaaa]]. |
| SUB aaaa | 4aaaa | Subtract value at specified address aaaa from accumulator. Result is stored in accumulator, A := A - M[aaaa]. |
| SUB (ss) | 924ss | Subtract exact value ss from accumulator. Result is stored in accumulator, A := A - ss. |
| SUB (sssss) | 93400 sssss | Subtract exact value sssss (located in next byte) from accumulator. Result is stored in accumulator, A := A - sssss. |
| SUB [aa] | 944aa | Subtract value from memory at address given |

| | | |
|---|---|---|
| | | in memory at address aa from accumulator, A := A - M[M[aa]]. |
| SUB [aaaaa] | 95400 aaaaa | Subtract value from memory at address given in memory at address aaaaa from accumulator, A := A - M[M[aaaaa]]. |
| MUL aaaa | 5aaaa | Multiply value from accumulator by value at specified address aaaa. Result is stored in accumulator A := A * M[aaaa]. |
| MUL (ss) | 925ss | Multiply value from accumulator by exact value ss. Result is stored in accumulator, A := A * ss. |
| MUL (sssss) | 93500 sssss | Multiply value from accumulator by exact value sssss (located in next byte). Result is stored in accumulator, A := A * sssss. |
| MUL [aa] | 945aa | Multiply value from accumulator by value from memory at address given in memory at address aa, A := A * M[M[aa]]. |
| MUL [aaaaa] | 95500 aaaaa | Multiply value from accumulator by value from memory at address given in memory at address aaaaa, A := A * M[M[aaaaa]]. |
| BRA aaaa | 6aaaa | Unconditional branch to instruction located at address aaaa. |
| BRN aaaa | 7aaaa | Conditional branch to instruction located at address aaaa if value stored in accumulator is negative. |
| BRNF aa | 907aa | Conditional branch to instruction located at address aa if flag NEGATIVE is TRUE. |
| BRZ aaaa | 8aaaa | Conditional branch to instruction located at address aaaa if value stored in accumulator is equal to zero. |
| BRZF aa | 908aa | Conditional branch to instruction located at address aa if flag ZERO is TRUE. |
| INC aaa | 01aaa | Increase value at address aaa by 1, M[aaa] := M[aaa] + 1. |
| DEC aaa | 02aaa | Decrease value at address aaa by 1, M[aaa] := M[aaa] - 1. |
| PUSH | 03000 | Push value from accumulator onto the stack, A -> STACK. |
| PUSH aaaaa | 91030 aaaaa | Push value at address aaaaa onto the stack, M[aaaaa] -> STACK. |
| PUSH (sssss) | 93030 sssss | Push exact value sssss onto the stack, ssssss -> STACK. |
| PUSH [aaaaa] | 95030 aaaaa | Push value at address specified at address aaaaa onto the stack, M[M[aaaaa]] -> STACK. |
| POP | 04000 | Pop value from the stack to accumulator, STACK -> A. |
| POP aaaaa | 91040 aaaaa | Pop value from the stack and put at address aaaaa, STACK -> M[aaaaa]. |
| POP [aaaaa] | 95040 aaaaa | Pop value from the stack and put at address specified at address aaaaa, STACK -> M[M[aaaaa]]. |