

| Код | Мнемоника | Длина | Описание |
| --- | --- | --- | --- |
| 0 | **STOP**  (stop executing) | 1 | Остановить работу |
| 1 | **LDC\_X *c***  (load constant in X) | 2 | Загрузить константу ***c*** (параметр команды) в регистр **X** |
| 2 | **LDC\_Y *c***  (load constant in Y) | 2 | Загрузить константу ***c*** (параметр команды) в регистр **Y** |
| 3 | **LDC\_P *c***  (load constant in P) | 2 | Загрузить константу ***c*** (параметр команды) в регистр **P** |
| 4 | **LDA\_X *a***  (load by address in X) | 2 | Загрузить значение из памяти по адресу ***a*** (параметр команды) в регистр **X** |
| 5 | **LDA\_Y *a***  (load by address in Y) | 2 | Загрузить значение из памяти по адресу ***a*** (параметр команды) в регистр **Y** |
| 6 | **LDA\_P *a***  (load by address in P) | 2 | Загрузить значение из памяти по адресу ***a*** (параметр команды) в регистр **P** |
| 7 | **LDP\_X**  (load by P in X) | 1 | Загрузить значение из памяти по адресу, расположенному в регистре **P**, в регистр **X** |
| 8 | **LDP\_Y**  (load by P in X) | 1 | Загрузить значение из памяти по адресу, расположенному в регистре **P**, в регистр **Y** |
| 9 | **STA\_X *a***  (store X by address) | 2 | Сохранить значение регистра **X** в памяти по адресу ***a*** (параметр команды) |
| 10 | **STA\_Y *a***  (store Y by address) | 2 | Сохранить значение регистра **Y** в памяти по адресу ***a*** (параметр команды) |
| 11 | **STA\_Z *a***  (store Z by address) | 2 | Сохранить значение регистра **Z** в памяти по адресу ***a*** (параметр команды) |
| 12 | **STA\_P *a***  (store P by address) | 2 | Сохранить значение регистра **P** в памяти по адресу ***a*** (параметр команды) |
| 13 | **STP\_X**  (store X by P) | 1 | Сохранить значение регистра **X** в памяти по адресу, расположенному в регистре **P** |
| 14 | **STP\_Y**  (store Y by P) | 1 | Сохранить значение регистра **Y** в памяти по адресу, расположенному в регистре **P** |
| 15 | **STP\_Z**  (store Z by P) | 1 | Сохранить значение регистра **Z** в памяти по адресу, расположенному в регистре **P** |
| 16 | **IN\_X**  (input X) | 1 | Ввести в регистр **X** значение с устройства ввода |
| 17 | **IN\_Y**  (input Y) | 1 | Ввести в регистр **Y** значение с устройства ввода |
| 18 | **OUT\_X**  (output X) | 1 | Вывести в устройство вывода значение регистра **X** |
| 19 | **OUT\_Y**  (output Y) | 1 | Вывести в устройство вывода значение регистра **Y** |
| 20 | **OUT\_Z**  (output Z) | 1 | Вывести в устройство вывода значение регистра **Z** |
| 21 | **CZP**  (copy from Z to P) | 1 | Скопировать значение регистра **Z** в регистр **P** |
| 22 | **CPX**  (copy from P to X) | 1 | Скопировать значение регистра **P** в регистр **X** |
| 23 | **CPY**  (copy from P to Y) | 1 | Скопировать значение регистра **P** в регистр **Y** |
| 24 | **JMP *a***  (jump to address) | 2 | Перейти на комманду по адресу ***a*** (параметр команды) |
| 25 | **JIZ *a***  (jump if zero to address) | 2 | Перейти на комманду по адресу ***a*** (параметр команды), если в регистре **Z** ноль |
| 26 | **JIP *a***  (jump if positive to address) | 2 | Перейти на комманду по адресу ***a*** (параметр команды), если в регистре **Z** положительное число |
| 27 | **JIN *a***  (jump if negative to address) | 2 | Перейти на комманду по адресу ***a*** (параметр команды), если в регистре **Z** отрицательное число |
| 28 | **ADD**  (add) | 1 | Сложить значения регистров **X** и **Y**, результат поместить в регистр **Z** |
| 29 | **SUB**  (subtract) | 1 | Вычесть из значения регистра **X** значение регистра **Y**, результат поместить в регистр **Z** |
| 30 | **MUL**  (multiply) | 1 | Перемножить значения регистров **X** и **Y**, результат поместить в регистр **Z** |
| 31 | **DIV**  (divide) | 1 | Разделить значение регистра **X** на значение регистра **Y**, результат поместить в регистр **Z** |