# DataCodeOne Order Code

| | Mnemonic | Order | Registers maybe Used | Index Registers Used | |
|---|---|---|---|---|---|
| ARITH. + LOGIC. | SET | Fetch (N) to register | ARST | PRST | |
| | STR | Store (register) to N | ARST | PRST | |
| | ADD | Add (N) to (register) | ARST | PRST | |
| | CMP | Compare (register) with (N) | ARST | PRST | |
| | SUB | Subtract (N) from (register) | AR ) | PRST | |
| | AND | Collate (N) with (register) | AR )ST if )literal | PRST | |
| | NEQ | Not equivalent (N) with (register) | AR ) | PRST | |
| | IOR | Inclusive or (N) with (register) | AR | PRST | |
| MLT. | MLT | Multiply (Register A) by (N) | - | PRST | |
| | | | | Conditions | |
| SHIFT | SHL | Shift (register) left N places | ARST ) | A (if N = 1,..,15) | |
| | | | ) | L (if N = 1,..,15) | |
| | SHR | Shift (register) right N places | ARST ) | LC(if N = 1) | |
| JUMP ON REG. | JEZ | Jump to N if (register) = Ø | ARS | P only. *See Note | JEZA |
| | JNZ | Jump to N if (register) ≠ Ø | ARS | '' | JNZA |
| | JPZ | Jump to N if (register) ⩾ Ø | ARS | '' | JPZA |
| | JNG | Jump to N if (register) < Ø | ARS | '' | JNGA |
| JUMP ON GGER | JVS | Jump to N if V trigger is set | - | '' | |
| | JVN | Jump to N if V trigger is not set | - | '' | |
| | JCS | Jump to N if C trigger is set | - | '' | |
| | JCN | Jump to N if C trigger is not set | - | '' | |
| JUMP ON COMPARISON | JLT | Jump if (register) < (N) in comparison | - | '' | |
| | JGE | Jump if (register) ⩾ (N) in comparison | - | '' | |
| | JEQ | Jump if (register) =(N) in comparison | - | '' | |
| | JNE | Jump if (register) ≠(N) in comparison | - | '' | |
| JUMP | J | Jump to location N | - | P,R,S,T *See Note | |
| | JS | Jump to N and store return address in S register | - | '' | |
| MISC. | SETL | Set operating Level | - | - | |
| | SETK | Set prohibitions | - | - | |
| | CLRK | Clear prohibitions | - | - | |
| | NUL | Do nothing | - | - | |

* Note

Jumps may be direct or indirect. When the index is P, the symbols ,P are always omitted. Thus the address required is specified by itself for direct jumps. For indirect jumps write address,,I if the implied index is P and write displacement,indexI for index R,S,T.