|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **The Instruction Table** | | | | |
| **Mnemonic** | **Opcode** | **Operand** | **Formal Specification** | **Description** |
| data |  | value |  | Reserve a memory location initialized to the value specified |
| ldc | 0 | value | B:=A;A:=value | Load accumulator with the value specified |
| adc | 1 | Value | A:=A+value | Add the value specified to the accumulator |
| ldl | 2 | offset | B:=A;A:+memory[SP+offset]; | Load local |
| stl | 3 | offsets | memory[SP+offset]=A;A:=B; | Store Local |
| ldnl | 4 | offset | A:=memory[A+offset] | Load non-local |
| stnl | 5 | offset | Memory[A+offset]:=B; | Store non-local |
| add | 6 |  | A:=B+A; | Addition |
| sub | 7 |  | A:=B-A; | Subtraction |
| shl | 8 |  | A:=B<<A; | Shift Left |
| shr | 9 |  | A:=B>>A; | Shift Right |
| adj | 10 | value | SP:=SP+value; | Adjust SP |
| a2sp | 11 |  | SP:=A;A:=B; | Transfer A to SP; |
| sp2a | 12 |  | B:=A;A:=SP | Transfer SP to A; |
| call | 13 | offset | B := A; A := PC; PC := PC + offset | Call procedure |
| return | 14 |  | PC := A; A := B; | Return from procedure |
| brz | 15 | offset | if A == 0 then PC := PC + offset; | If accumulator is zero, branch to specified offset |
| brlz | 16 | offset | if A < 0 then PC := PC + offset; | If accumulator is less than zero, branch to specified offset |
| br | 17 | offset | PC := PC + offset; | Branch to specified offset |
| HALT | 18 |  |  | Stop the emulator. This is not a `real' instruction, but needed to tell your emulator when to finish. |
| SET |  | value |  | Set the label on this line to the specified value (rather than the PC). This is an optional extension, for which additional marks are available. |