Little Man Computer references

|  |  |  |  |
| --- | --- | --- | --- |
| Numeric code | Mnemonic code | Instruction | Description |
| 1xx | ADD | ADD | Add the value stored in mailbox xx to whatever value is currently on the accumulator (calculator).  Note: the contents of the mailbox are not changed, and the actions of the accumulator (calculator) are not defined for add instructions that cause sums larger than 3 digits. |
| 2xx | SUB | SUBTRACT | Subtract the value stored in mailbox xx from whatever value is currently on the accumulator (calculator).  Note: the contents of the mailbox are not changed, and the actions of the accumulator are not defined for subtract instructions that cause negative results - however, a negative flag will be set so that **8xx (BRP)** can be used properly. |
| 3xx | STA | STORE | Store the contents of the accumulator in mailbox xx (destructive).  Note: the contents of the accumulator (calculator) are not changed (non-destructive), but contents of mailbox are replaced regardless of what was in there (destructive) |
| 5xx | LDA | LOAD | Load the value from mailbox xx (non-destructive) and enter it in the accumulator (destructive). |
| 6xx | BRA | BRANCH(unconditional) | Set the program counter to the given address (value xx). That is, value xx will be the next instruction executed. |
| 7xx | BRZ | BRANCH IF ZERO (conditional) | If the accumulator (calculator) contains the value 000, set the program counter to the value xx. Otherwise, do nothing.  Note: since the program is stored in memory, data and program instructions all have the same address/location format. |
| 8xx | BRP | BRANCH IF POSITIVE (conditional) | If the accumulator (calculator) is 0 or positive, set the program counter to the value xx. Otherwise, do nothing.  Note: since the program is stored in memory, data and program instructions all have the same address/location format. |
| 901 | INP | INPUT | Go to the INBOX, fetch the value from the user, and put it in the accumulator (calculator)  Note: this will overwrite whatever value was in the accumulator (destructive) |
| 902 | OUT | OUTPUT | Copy the value from the accumulator (calculator) to the OUTBOX.  Note: the contents of the accumulator are not changed (non-destructive). |
| 0 | HLT | HALT | Stop working. |
|  | DAT | DATA | This is an assembler instruction which simply loads the value into the next available mailbox. DAT can also be used in conjunction with labels to declare variables. For example, DAT 984 will store the value 984 into a mailbox. |