Op Codes

|  |  |  |  |
| --- | --- | --- | --- |
| Machine Code | Assembly Command | Operation | Parameters |
| 0000 | \_add\_ | add | num1, num2, output register |
| 0001 | \_sub\_ | subtract | num1, num2, output register |
| 0010 | \_grt\_ | greater than | num1, num2, output register |
| 0011 | \_eql\_ | equals to | num1, num2, output register |
| 0100 | \_jmp\_ | unconditional jump | lineNum |
| 1111 | \_cjp\_ | conditional jump | lineNum, num1, num2, comparison operation |
| 0101 | \_rst\_ | register set | register, ornum |
| 0110 | \_rrd\_ | register read | register, output register |
| 0111 | \_rcl\_ | clear registers |  |
| 1000 | \_and\_ | and | num1, num2, output register |
| 1001 | \_bor\_ | or | num1, num2, output register |
| 1010 | \_xor\_ | exclusive or | num1, num2, output register |
| 1011 | \_not\_ | not | num, output register |
| 1100 | \_rld\_ | register load | ram address, register |
| 1101 | \_rms\_ | ram set | ram address, register |
| 1110 | \_inv\_ | invert | num, output register |

Bit Sizes:

* num: 4 bit (just another name for a register)
* ornum: 16 bit (this is a number that is directly inputted, not read from a register)
* lineNum: 16 bit
* line: 32 bit (an actual line of code)
* ram address: 8 bit
* register: 4 bit (input register)
* output register: 4 bit
* command: 4 bit
* comparison operation: 4 bit (this includes \_eql\_ and \_grt\_)

Syntax Examples:

* registers: rg1
* ram: rm1
* line number: ln1
* Op Code: \_rst\_ rg1 4269
* Machine Code: 0101 0001 0001000010101101 00000000 ← zero padding at the end
* operation param1 param2… paramn