



## RAM/Registers

load(addr1, addr2) → load \*addr1 from RAM into \*addr2 in the registers.  
mov(addr1, addr2) → store \*addr1 from Registers into \*addr2 in the RAM.

## Registers Programs

set(addr, val) → Set \*addr in the registers to val.  
copy(addr1, addr2) → Set \*addr2 of the registers to \*addr1 of the registers.

## Calculus

add(addr1, addr2) ⇒ Add \*addr1 from the register to \*addr2 by overwriting addr1.  
sub(addr1, addr2) ⇒ Sub \*addr1 from the register to \*addr2 by overwriting addr1.  
mul(addr1, addr2) ⇒ Mul \*addr1 from the register to \*addr2 by overwriting addr1.  
div(addr1, addr2) ⇒ Div \*addr1 from the register to \*addr2 by overwriting addr1.  
cmps(addr1, addr2) ⇒ Write 0/1 to addr1 if \*addr1 is > than \*addr2.  
cmp(addr1, addr2) ⇒ Write 0/1 to addr1 if \*addr1 is == to \*addr2.  
cmpi(addr1, addr2) ⇒ Write 0/1 to addr1 if \*addr1 is < than \*addr2.

## Serial

read\_serial(addr1, addr2) ⇒ Write the value of addr1 of the serial output port into addr2.  
write\_serial(addr1, addr2) ⇒ Write the value of addr2 to the serial input port of addr1.

## Clock editor

goto(valeur, none) ⇒ Jump to the value of the bitcode in the clock.  
gotoif(valeur, addr) ⇒ Jump to the value if \*addr in registers is == 1.