Assembly Syntax for Processor Simulators and code template

Instructions types	Meaning
Transfer instructions, register <-> memory	
li, si	load or store integer data from / toward register
lf, sf	load or store floating point data from / toward register
move	copy from one register to another
cmovz, cmovnz, cmovn, cmovp	Conditionnal copy from one register to another registres
Arithmetique and logic instructions	
add, addf	addition between integer or floating point data
sub, subf	substraction between integer or floating point data
mult, multf	muliplication between integer or floating point data
and, nand	logic operation, and, nand, etc
or, nor, xor	logic operation, or, nor, xor
Control instructions	
br	Inconditionnal jump/branch
brz, brnz	branch if register is equals/no equals to 0
brp, brn	branch if register is equals/no is positif/negative
nop	instruction doing nothing
exit	end of program

Table 1.1 – Instructions set

```
.data   X             -- Beginning of data memory section, starting at address X
.global values1       -- Label definition
      12 3 18 37 61   -- Values to store at the addresses X, X+1, X+2, etc
.data   Y             -- New data memory section
.global values2      -- Label definition
      10  110  24     -- Values to store at addresses Y, Y+1, Y+2, etc
.global n              -- Label definition
    32 -- value to store
.alias Nb 16          -- Alias definition
    ...
    ...
.program Z            -- Beginning of program section, starting at address Z
    li R1, X
    li R3, Y
    li R2, (R1)
    ...
    ...
.boucle               ; Label definition, for example for a loop
    ...
    ...
    add R9, R9, 1
    sub R3, R3, 1
    brnz R3, boucle
    ...
    ...
    exit              ; end of program
.end                  ; End of program section
```

Listing 1 – Code template for processor simulator.