

# INTERPRETER

“Interpreter.c” is a C-Program which reads a SimpleRisc assembly language program (written in a file) and executes its instructions to give the desired output(s).

## Features of the Interpreter

- It supports all 21 SimpleRisc instructions including modifiers u and h.
- The assemble language program can have a *free form*. An instruction has to be in one line but there can be *any* number of spaces or tabs between the fields and before the instruction. There can also be any number of empty lines between statements.
- The interpreter specifies any errors in the assembly file and *also mentions the line number* where the error is present.
- The assembly file can have any number of labels and a label can either be on the same line as the instruction it points to, or it can be anywhere after the previous instruction.
- The program *must* contain a *.main* label from where the program execution starts.
- The stack pointer, r14, can be accessed as sp and the return address register, r15, can be accessed as ra.
- 16-bit immediates (in either decimal or hex) are supported and can have the form 21, -53, 0xAB12, 0x a25 or 0x 5      A 4 C.
- It supports a 32-bit memory system. However, only the bottom 4096 entries are allowed to be used. The interpreter allows only memory addresses that are multiples of 4 between 0 and 4095 to be accessed using the ld and st instructions.
- The stack pointer is initially automatically initialized to 0xFFC, i.e. 4092 (the upper end of the memory) and hence the program need *not* initialize the stack pointer.
- The interpreter also supports the two types of comments:-
  1. The single line comments which begin with @
  2. The multi line comments which are of the form /\* ..... \*/
- The interpreter supports *two extra features*: 1) the .print macro and 2) the .encode macro.

## The .print macro

The .print macro is used to print the value of a register on the console. One can print the decimal values of any number of registers using one .print macro.

For example, “.print r0, r2, r11, r9, sp, ra, r6” will print the values of the registers r0, r2, r11, r9, r14(sp), r15(ra) and r6 in decimal on new lines on the console.

Every instruction containing the .print macro is treated as *any other valid SimpleRisc instruction*.

## The .encode macro

The .encode macro is used to print the encoding of a valid SimpleRisc instruction on the console. For example, “.encode addh r15, sp, 0x135” will print the encoding of “addh r15, sp, 0x135” which is 0x07fa0135.

The instruction which is being encoded is ***NOT*** *executed*. For example, addh r15, sp, 0x135 is ***not*** executed but just encoded and printed in the above example.

Every instruction containing this .encode macro is treated as *any other valid SimpleRisc instruction* (i.e. it can be preceded by a label, etc.).

## How to use the Interpreter?

- Compile the interpreter, “Interpreter.c” to obtain the executable. (you can use the makefile provided using the “make” command if you want)
- To execute the assembly file, run the command
  - ⊙ “./<name of executable> <path/name of the assembly file>” in linux and the command “<name of executable> <path/name of the assembly file>” in windows or
  - ⊙ “./run.sh <path/name of the assembly file>” in linux if you used the makefile to compile the interpreter.