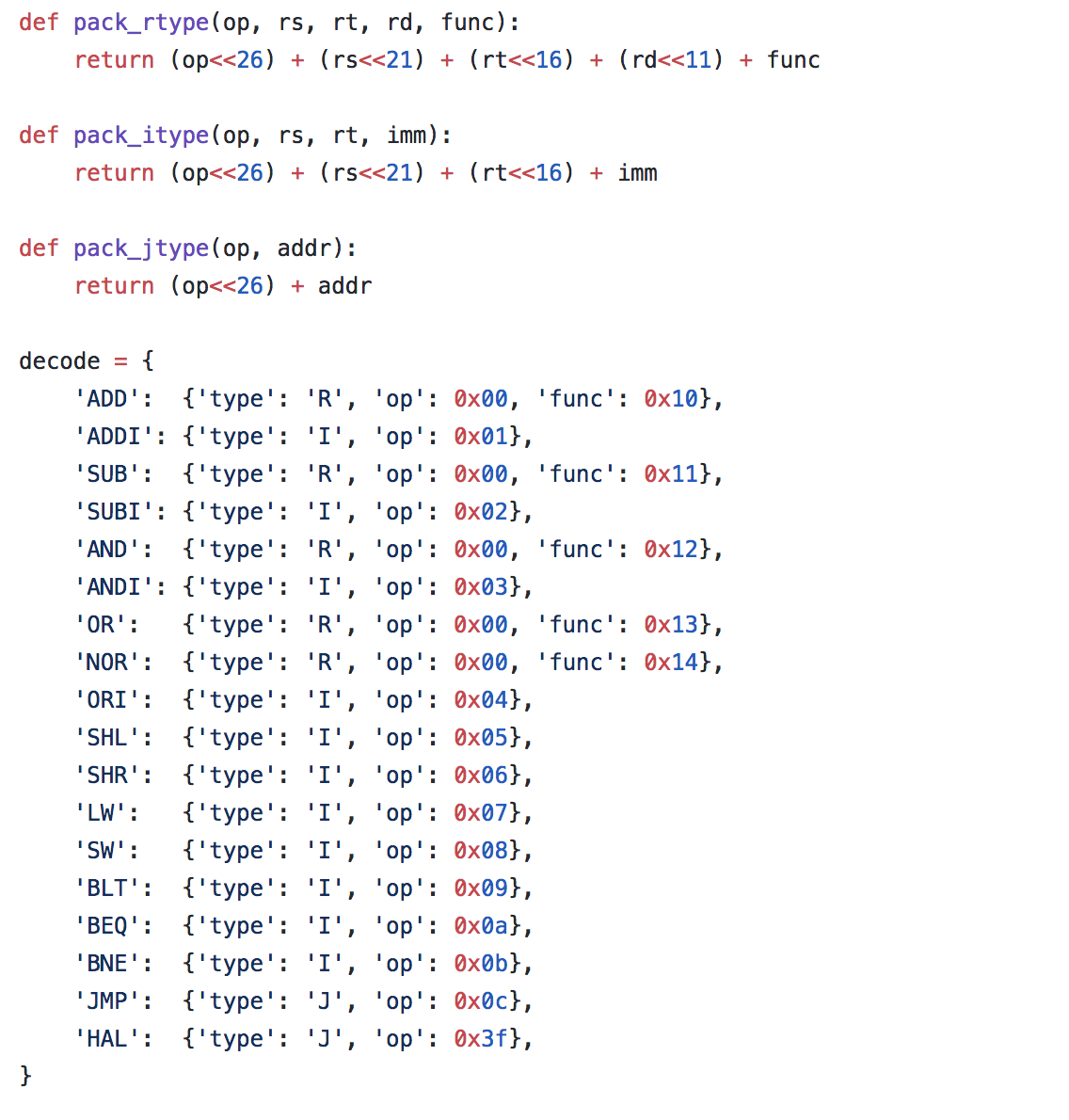
2.3.1 Assmebly Compiler

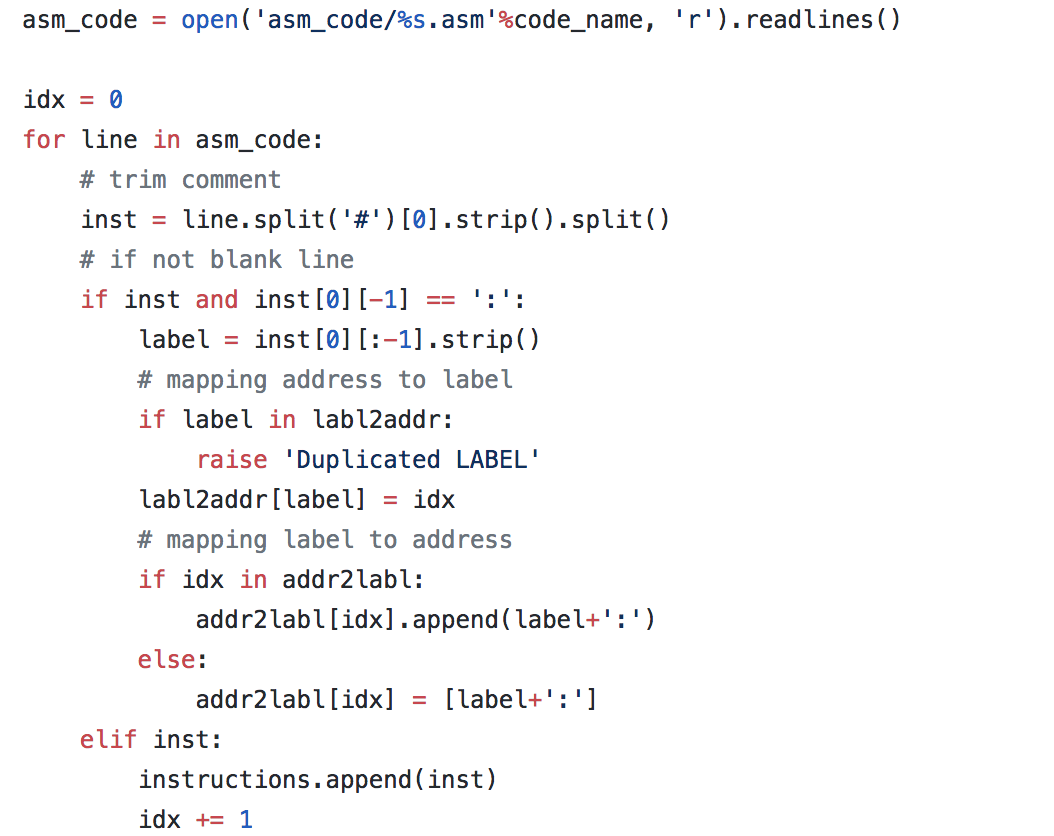
We wrote the RC5 code in assembly language (rc5.asm & rc5\_optimized.asm) to give instructions of how the MIPS workd. We also wrote a simple compiler in python (load\_tb\_instructions.py) to help convert assembly to machine code (rc5.binary & rc5\_optimized.binary) , which can be conducted directly on the FPGA board.

In our python, first, we define three functions seperately for I, R and J type to use bit shifting to combine the input such as op, rs, rt, rd, imm and addr to its machine code format. Then, in an Array, we define the details about type, opcode and funcode of each of the needed instructions as dictionary.



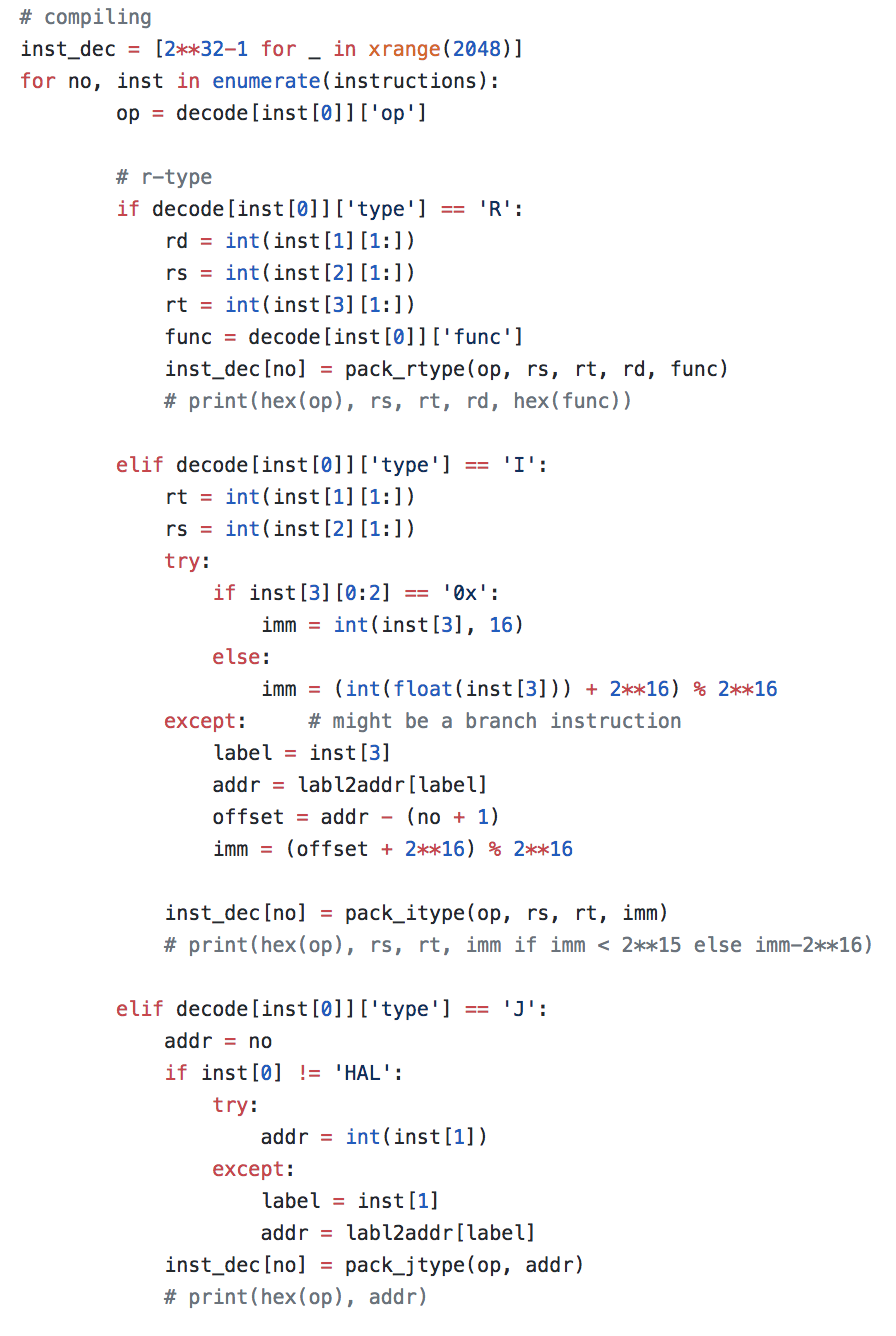
2.3.1.1 divide instruction types

Then, we use python File readline fuction to get the assembly instructions.



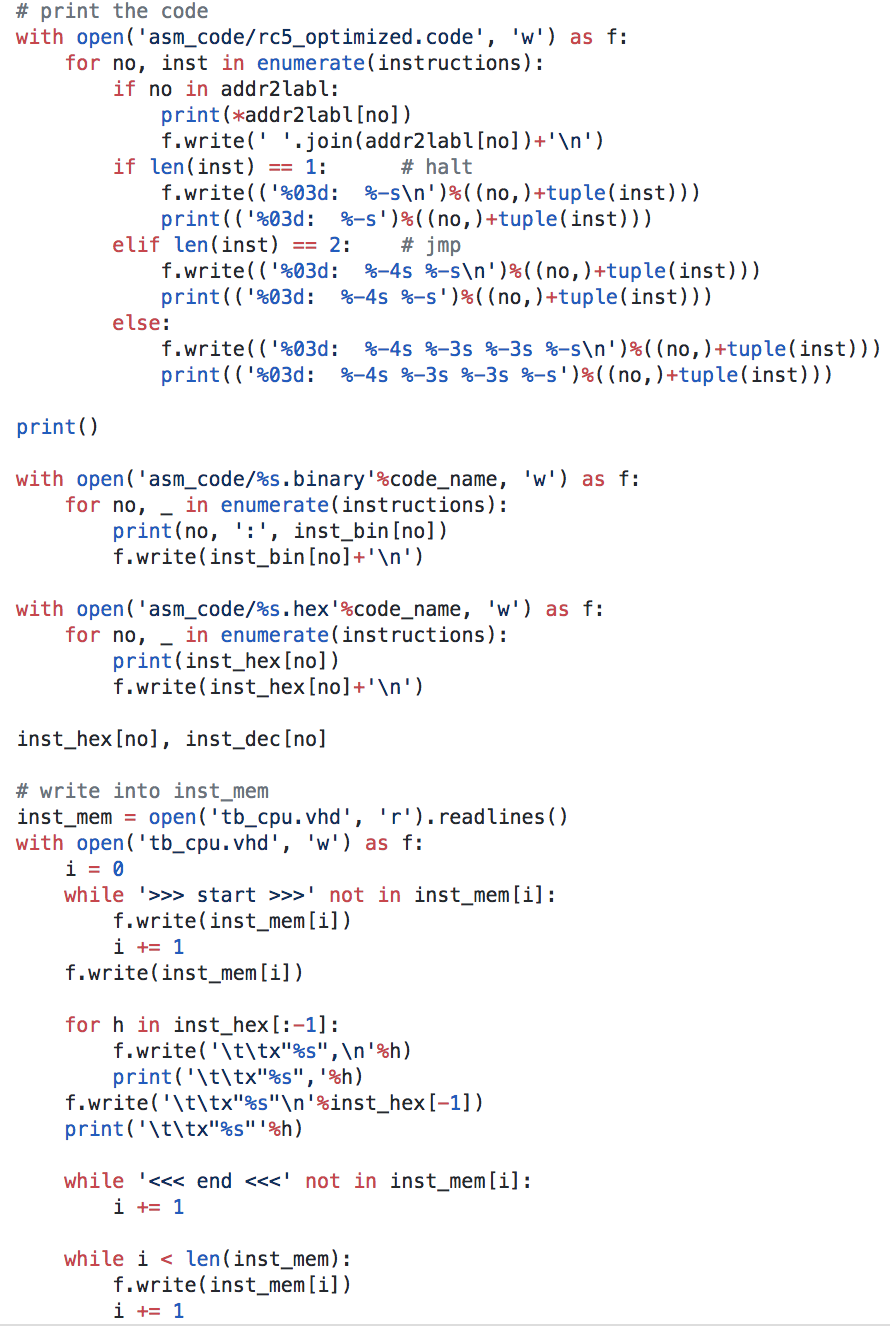
2.3.1.2 read input stream

Sequentially, for each current instruction, we split it by language pattern.



2.3.1.3 judge the instruction type

Then, find out what type it is and call the according function for the machine code. At last, just write these machine code into the inst\_mem part of the tb\_cpu.vhd.



2.3.1.4 write output into the right place into inst\_mem of tb\_cpu.vhd