COMP262-01
CPU MC Project: a PEP8 CU Micro code implementation
PART5(MC5) CODING of the: **CALL** and **RETn** instructions

Continue with the micro coding.
In MC4, we have implemented the ASL, ASR, ROR, ROL and a 'new' instruction: REVLOA.
For this part (MC5), we will implement the: **CALL** and **RETn** instructions.
Grade: CALL-30%, RETn-70%
The trinary **CALL** instruction pushes the content of the program counter onto the runtime stack and then loads the Instruction operand into the program counter.
Here is the RTL specification of the CALL instruction:
SP <- SP - 2; Mem[SP] <- PC; PC <- IOprnd

The unary **RETn** instruction has a three-bit **nnn** field, with the following bit allocation: 0101 1nnn
There are eight versions of the RETn instruction, namely RETO, RET1,..., RET7, where **n** is the number of bytes occupied by the local variables in the runtime stack.
Here is the RTL specification of RETn:
SP <— SP + n; PC <- Mem[SP]; SP <- SP + 2
First, the instruction deallocates storage for the local variables by adding **n** to the stack pointer.
After the deallocation, the return address should be on top of the runtime stack. Then, the instruction moves the return address from the top of the stack into the program counter. Finally, it adds 2 to the stack pointer, which completes the pop operation.

We have provided a 'shell' for the micro code. You must supply the RTL as comments as was provided to you in the previous assignments(MC0-1-2).
Complete and verify that the required micro code produces the correct results, which are deterministic.

Deliverables
The completed micro code;
YOU MUST PROVIDE DETAILED comments indicating:
1) WHAT parts of the implementation are coded but not working and
2) WHAT is left to be coded.