COMP262-01
MC Project: PEP8 CPU CU Micro-coding: Description

Description and Background

The PEP8 machine is an Abstract construct, created by Prof. J. Stanley Warford, as a pedagogical tool to study the basic Architecture and Organization of a computer. It is described in his textbook "Computer Systems" and a Software Simulation of such machine was developed, made available and named "PEP/8".
In the PEP8 CPU SIM Project, the students implements a HLL program (JAVA) which emulates the 'RUN/STEP' Part of the PEP/8 APP. This is an instruction-level execution Simulator, for a subset of the PEP8 instruction set. This instruction-level Simulator models the execution behavior of each instruction.
In addition, prof. Warford has provided another level of Software Simulation for the PEP8 machine, in this case depicting the internal activities that the CPU's CU conducts in order to 'Execute' a Machine Instruction. This interactive Simulation was developed and made available under the name of "PEP8-CPU".

Goals/Constraints/Guidelines

For this Project, the COMP262 student will simulate (program) the steps that the PEP8-CPU Control Unit must follow in order to control the 'Execution of a machine instruction.
To this effect, the student will be developing, the necessary Micro Operations, using the PEP8 Symbolic Micro Code, which may be understood to be analogous to what the PEP8 Symbolic Pep8 Assembly language is to the actual machine instructions.
The CPU Simulator will "run/execute" the series of Micro operations, one at a time, for each 'Clock' cycle.
The student will use the Interactive App in order to:
1) describe the Micro operations
2) validate the syntax, which is done by the 'Encoder' part of the App
3) 'Step/Cycle/Clock' the actual execution of the series of micro operations.

The project will be implemented in weekly Parts, and will implement selective parts of the Von Neumann cycle that was implemented in the CPU SIM project.
For example: since this implementation of the Micro code does not provide a way to alter the control flow, that is to say, a way to implement logical decision making, we are constrained to carry out what amounts to a 'Linear' execution and therefore unable to implement the DI(Decode instruction) step.
We will be able to implement the 'FI (Fetch Instruction) cycle and the equivalent of the:
'CO (Calculate the Operand's EA), the FO (Fetch operand) and WO (Write operand) for an instruction.
These are very standard operations that once coded, apply to all instructions.
Since we are unable to perform a DI (Decode instruction), therefore we will assume WHAT a given instruction is and that it has already been Fetched into the IR and proceed with the CO/FO/EI (Execute instruction) steps. The App provides for the 'initializing' of both Memory and Registers, so that we can 'test' the instruction by pre loading memory with the needed operands.
We will not be able to implement the 'Conditional Branch' instructions, since they require the ability to make a logical decision that is not available in the App.
For the rest of the instructions, we will selectively implement many of them, and will illustrate the 'power' that the Micro coded CU implementation provides, by 'creating/implementing' new instructions.
By the end of the project, the student would have acquired the knowledge and skills necessary to fully describe the implementation of a Computer Processor.