COMP25111 Lab Exercise 1: MU0 Control Signals (for students who took COMP15111)

Duration: 1 session

Aims

To give you some insights into computer architecture.

Learning outcomes

A student who successfully completes this exercise will

- understand the data-paths in, and
- have designed the control signals necessary to implement,

a very simple CPU (MU0).

Summary

Edit the given latex file `answer.ltx` to show the signal needed for each control line for each step (as defined below), for the design of MU0 outlined in the lectures.

The unextended deadline is the end of your scheduled lab session.
If you attend the lab you will, if you need it, automatically get an extension until one week after your scheduled lab session commences - you must use submit to prove you finished in time.
You must get your work marked in this lab or in your next scheduled lab.
You can also get an extension for good reason e.g. medical problems.

Description

For this lab exercise you should do all your work in your `COMP25111/ex1` directory.

Copy the starting (latex) file `answer.ltx` into your `COMP25111/ex1` directory from `/opt/info/courses/COMP25111/ex1`

Edit `answer.ltx` (**not** .tex, so that labprint works) to show the signal needed for each control line for each step (as defined below), for the design of MU0 outlined in the lectures.

The 9 **control lines** are:
To enable various internal registers to be modified:
- `En_IR` (enable write to IR),
- `En_PC` (enable write to PC),
- `En_ACC` (enable write to ACC).
To set the action performed by the ALU (so at most one of these signals can be set at once):
- `byp` (bypass),
- `add` (add),
- `sub` (subtract).
To set the action performed by the RAM (so at most one of "Read" or "Write" can be set at once):
- `Ren` (Read),
- `Wen` (Write),
- `addr_Mux` (set means the address used by the RAM is copied from the PC register, clear means it is copied from the S part of the IR register).

The 8 possible **steps** are:
- during the Fetch phase, and
- during the Execture phase, for each different possible instruction (LDA, STA, ADD, SUB, STP, Jump, No Jump)

Note:
- "Jump" means either an (unconditional) JMP instruction, or else a (conditional) JGE or JNE instruction when the condition is true so the jump happens.
- "No Jump" means a (conditional) JGE or JNE instruction when the condition is not true so the jump does not happen.

For each control line, for each step, the **signal** that can be used is one of:
- "0" (clear/false),
- "1" (set/true), or
- "X" (don't care).

Assessment

You must use `labprint` and `submit` as normal. They will look for: `answer.ltx`
(*Don't worry if either program also asks about e.g. "add.mu0" and "divide.mu0" - these files are part of the alternative lab exercise for students who did not take COMP15111, and should be ignored.*)

The marks are awarded as follows:
4 - Fetch phase
4 - Execute phase for STA
4 - Execute phase for LDA
4 - Execute phase for ADD
4 - Execute phase for SUB
4 - Execute phase for STP
4 - Execute phase for Jump
4 - Execute phase for No Jump
Total 32

For each step, a mark is lost for each incorrect signal (up to the maximum for that step)

Remember to get your solution marked as soon as possible after it has been submitted.