

COMS12200 lab. worksheet: week #8

- We intend this worksheet to be attempted in the associated lab. session, which represents a central form of help and feedback for this unit.
- The worksheet is not *directly* assessed. Rather, it simply provides guided, practical exploration of the material covered in the lecture(s): the only requirement is that you archive your work (complete or otherwise) in a portfolio via the appropriate component at

<https://www.fen.bris.ac.uk/COMS12200/>

This forms the basis for assessment during a viva at the end of each teaching block, by acting as evidence that you have engaged with and understand the material.

- The deadline for submission to your portfolio is the end of the associated teaching block (i.e., in time for the viva): there is no requirement to complete the worksheet in the lab. itself (some questions require too much work to do so), but there is an emphasis on *you* to catch up with anything left incomplete.
- To accommodate the number of students registered on the unit, the single 3 hour lab. session shown in your timetable is split into two 1½ hour halves. You should only attend *one* half, selecting as follows:
 1. if you have a timetable clash that means you *must* attend one half or the other then do so, otherwise
 2. execute the following BASH command pipeline

```
id -n -u | shasum | cut -c-40 | tr 'a-f' 'A-F' | dc -e '16i ? 2 % p'
```

e.g., log into a lab. workstation and copy-and-paste it into a terminal window, then check the output: 0 means attend the first half, whereas 1 means attend the second half.

- Keep in mind that hardware and software within the lab. is managed by the IT Services Zone E team. If you encounter a problem (e.g., a workstation that cannot be powered-on, an error when executing some software, some missing software, or you just cannot log into your account), *they* can help: either talk to them in room MVB-3.41, and/or submit a service request online via

<http://servicedesk.bristol.ac.uk>

Q1. Since this is the final worksheet for the first part of COMS12200, it makes sense to recap and take stock of the material covered so far: before moving on, try to complete and submit your work for *all* previous worksheets (making sure to ask for help if you cannot understand a given topic).

Q2. This question is intentionally different from those in previous worksheets, in that a) it is *very* open-ended, meaning b) it might take *much* longer to complete and c) you cannot assume enough information provided by the lecture(s) alone, and so research of your own *will* be required.

Using built-in components provided by Logisim (or those from previous worksheets as appropriate), your challenge is to design and then implement a simulated calculator. Make sure you thoroughly document both the design you arrive at and the resulting implementation: given that both are likely to be more complicated than your previous work, such documentation will allow you to recall and explain both aspects more easily.

The difficulty associated with this challenge ranges from relatively simple to *very* difficult, depending on the functionality¹ supported. Keep in mind that although a solution *is* provided, the goal is for *you* to think about what is possible and how to implement it (rather than to follow a pre-specified set of instructions or reproduce said solution). This means an emphasis on *you* developing the design and implementation on your own, and *no* requirement to focus on the HP-35 used as an example in the lecture(s).

¹To realistically be deemed a calculator, a given design should meet some arguable baseline level of functionality; an 8-bit adder, for example, does arithmetic but you would not class it as a calculator. As a guide, this baseline might be the ability to evaluate an expression such as $(19 - 5) \cdot (1 + 2)$ which we discussed in the lecture(s); a looser definition could simply be that it is useful for some non-trivial task. Either way, one might expect the design to include a data-path (e.g., an Arithmetic and Logic Unit (ALU) plus some registers for storage) and an associated control-path tasked with managing key presses by the user.