Inf2C - Computer Systems Wrap-Up

Boris Grot

School of Informatics

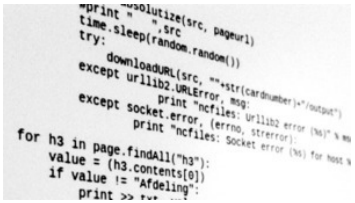
University of Edinburgh

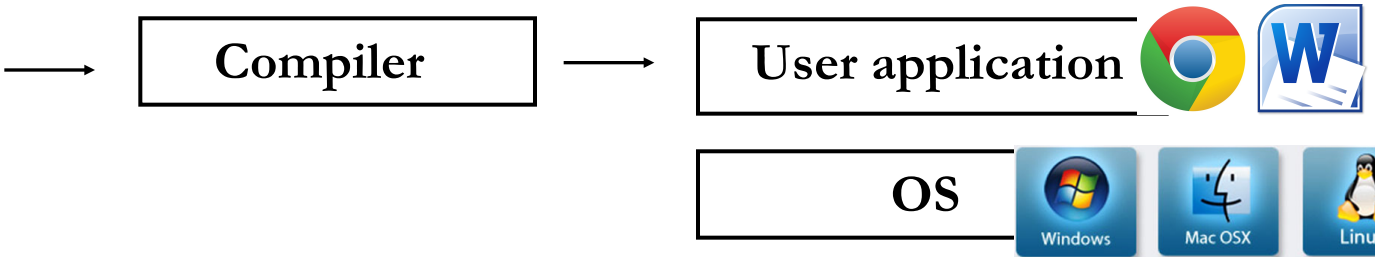
So what did we study in this course?

What did we learn?

- Hardware:
  - Data representation and operations
  - Basic circuits
  - Processor organisation
  - Exceptions and interrupts
  - The memory subsystem
  - Input/Output (I/O)
- Software:
  - Low-level (assembly) programming
  - Operating systems basics
  - C programming

Want to know more?





- Computer Organization & Design
  - Computer Design (UG3)
  - Computer Architecture (UG3)
  - Parallel Architectures (UG4)
- Operating Systems (UG3)
- Compilers
  - Compiling Techniques (UG3)
  - Compiler Optimisation (UG4)
- Extreme Computing (UG4) - cloud

Student Surveys

- Should be online first week of Dec
- Your feedback matters
  - This year's course is better than before thanks to previous years' feedback
- I want it all: the good, the bad, the ugly
  - The more feedback, the better

Exam

Friday, Dec 16 @ 2.30pm

- Check timetable to confirm date/time/place
- Similar format to previous years
- Covers all lectures
- Lecture material, notes and assigned reading all fair game

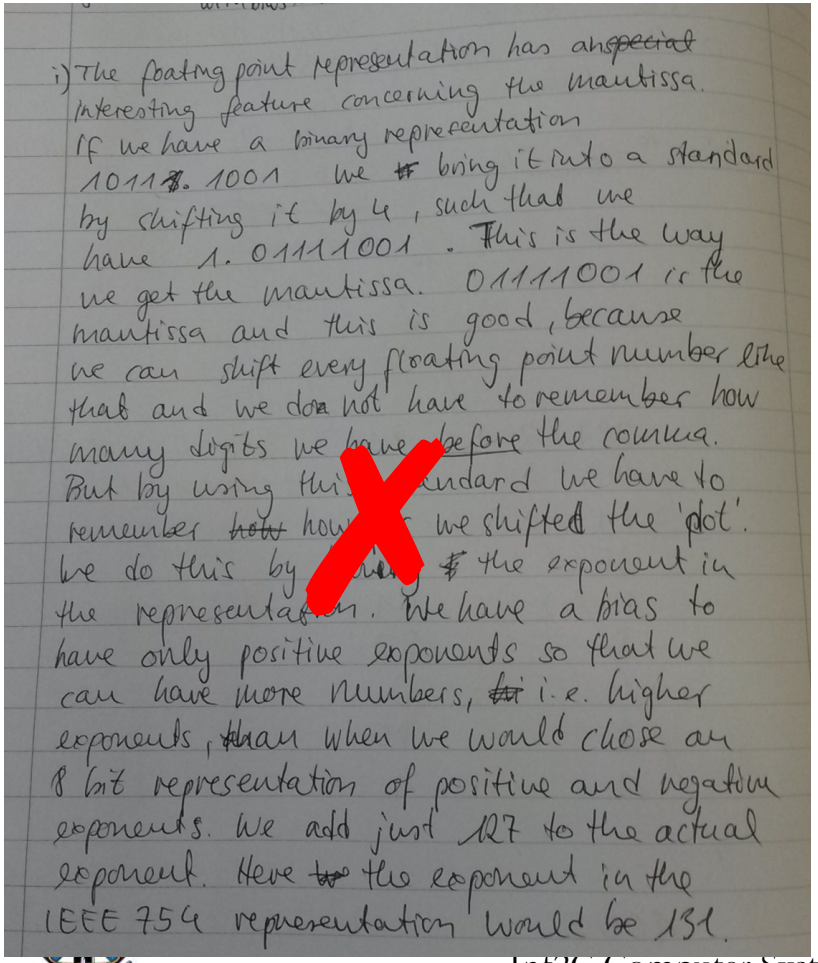
Exam: answering questions

Some questions will ask to explain your reasoning or justify your answer.

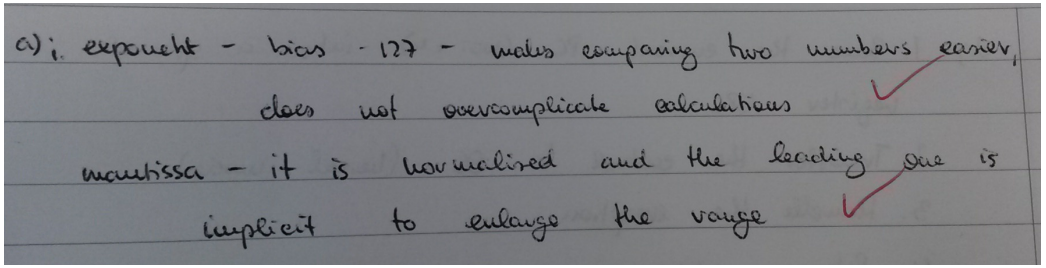
- Keep your responses **short** and **focused**
- Please **no essays!**

Example question (abridged, from last year's exam):
In the IEEE 754 FP standard, both exponent and mantissa have interesting features. What are they & why are they useful?

Bad answer (too long)


i) The floating point representation has ~~a~~ special interesting feature concerning the mantissa. If we have a binary representation 1011$. 1001 we bring it into a standard by shifting it by 4, such that we have 1.01111001. This is the way we get the mantissa. 01111001 is the mantissa and this is good, because we can shift every floating point number like that and we do not have to remember how many digits we have before the comma. But by using this standard we have to remember how we shifted the 'dot'. We do this by the exponent in the representation. We have a bias to have only positive exponents so that we can have more numbers, i.e. higher exponents, than when we would chose an 8 bit representation of positive and negative exponents. We add just 127 to the actual exponent. Here the exponent in the IEEE 754 representation would be 131.

Good answer (short & sweet)


a) i. exponent - bias - 127 - makes comparing two numbers easier, does not overcomplicate calculations

mantissa - it is normalised and the leading one is implicit to enlarge the range

Questions?