Computer Fundamentals Examples Sheet

Dr Robert Harle
Michaelmas 2012

Numbers and Bases (Complete before Lecture 2)

Q1. Convert the following decimal numbers to binary, octal, hexadecimal, and base-5: i) 10 ii) 567 iii) 111

Q2. Treating a decimal number as an addition of powers of ten extends across the decimal point—so 23.65=$2.10^1+3.10^0+6.10^{-1}+5.10^{-2}$. In binary, the same is true, but we use powers of two. Therefore what does the binary fractional number 101.101 represent in decimal? What is 1.25_{10} in binary?

Q3. How many bits are needed to represent a number that has d decimal digits? More generally for any number N, how many base-b digits are needed to represent it if it takes k base-c digits?

Q4. The classical subtraction algorithm taught at schools involves a system of subtractions and 'borrows'. Demonstrate that the same algorithm works in binary for (a-b), where $a > b$. What if $b > a$?

Q5. (Harder) In an even base, if the last (least significant) digit is odd, then the entire number is odd. Derive a similar rule to check for an odd number when the base is also odd.

CPU and Data Representations

Q6. With the aid of a diagram, describe the *fetch-execute cycle*.

Q7. How does the register size of a system affect the amount of memory a system can support? How much system memory can we use if we have 8, 32 or 64 bit registers?

Q8. Demonstrate that the circuitry needed to handle the addition of unsigned binary numbers is the same as for signed binary numbers if they use two's complement. Give some insight into why this is the case.

Q9. With different examples from those in the notes, explain the use of the carry and overflow flags in integer addition/subtraction.

Q10. An alternative technique for representing negative numbers is *one's complement*. Here we invert all of the bits of a number to find its negative. i.e. +2 is 0010, so -2 is 1101. Compare this representation with sign-magnitude and two's complement.

Q11. Imagine a 4-bit computer (i.e. all its registers are 4 bits wide). Show how it could ever compute the result of (47+17)

Modern Systems

Q12. Compare and contrast CPU registers, RAM, and hard discs.

Q13. Compare the use of mercury delay lines to the use of RAM for computer storage

Q14. A computer designer proposes replacing the 2GB of main system RAM in a desktop computer with millions more registers to make the same level of storage. Comment on the advantages and disadvantages of this approach.

Q15. Today's computers feature 'standby' and 'hibernate' low power modes. The standby mode keeps the data and CPU state in RAM; the hibernate mode copies it to the hard disc. Explain why standby draws more power but resume is faster.

Q16. Modern machines have single-digit gigabytes of RAM. Whilst this is a lot, it's often not enough. One approach to extending the availably memory is to use the hard disc as "swap space". Here we free up RAM by pushing chunks of data to the hard disc, reading them back into RAM only when the CPU needs them. Comment on the advantages and disadvantages of this approach.

Memory Manipulation

Q17. Imagine you have a pointer to the start of a string that is properly terminated. Describe algorithms to i) find the length of string; ii) test whether it is a palindrome (the same backwards as forward). You are not expected to write any code, only to describe the algorithm in words.

Operating System Basics

Q18. Explain the purpose of an Operating System.

Q19. Microsoft Windows uses "quantum stretching", where the current, in-focus application gets a longer time slot on the CPU. Explain why.

Q20. Apple recently started using Intel processors that support x86 instructions. This means Apple machines can now run Microsoft Windows. However, off-the-shelf PC software (which is compiled for x86) does not run directly on a Mac that is using the Apple operating system compiled for the Intel processor. Why not?

Compilers, Architectures and Programming

Q21. Why do you think CPU manufacturers haven't all agreed on a single Instruction Set Architecture?

Q22. Explain what is meant by i) a compiler; ii) an interpreter.

Q23. Distinguish between the imperative and functional programming paradigms.