

Introduction to Computer Architecture: Supervision 3

Lectures covered by the supervision: <https://www.cl.cam.ac.uk/teaching/2627/IntComArch/>

- Lecture 9: Support for operating systems
- Lecture 10: Other instruction set architectures
- Lecture 11: Overview of Systems-on-Chip (SoCs) and DRAM
- Lecture 12: Multicore Processors part I

Past exam questions:

<https://www.cl.cam.ac.uk/teaching/exams/pastpapers/t-IntroductiontoComputerArchitecture.html>

<https://www.cl.cam.ac.uk/teaching/exams/pastpapers/t-ComputerDesign.html>

Supervision questions:

1. Sketch examples of a i) memory read and a ii) memory write, former cache hit, latter cache miss, considering a processor with a single level of cache, virtual memory (TLB, page table), and RAM. Discuss how these events impact execution time.
 - a. Does a CPU work with real or virtual memory?
 - b. How does a Translation Look-aside Buffer TLB improve performance of virtual memory accesses?
2. What is the purpose of interrupts? Is debugging an interrupt?
3. [2008 Paper 6 Question 2](#) - part a
4. [2006 Paper 6 Question 2](#)- part b
5. What is a page table, where is it stored and how is it initialised?
6. What are the main components of a modern SoC? Compare SoC to chiplets.
7. Suppose that there are four different types of processor: A does 15% of the work for an application, B does 20%, C does 5% and D does 60%. If you speed B up by 2× and D up by 3×, what is the overall application speed up:
 - a. When all four run in parallel?
 - b. When all four run sequentially?
 - c. What's the limit on the speedup you can get from only improving B and D?
8. Draw a 1T1C DRAM cell. Why does it need refresh and what is it? Compare it with a single SRAM cell.
9. Show, with the aid of a diagram, how DRAM is organised, making reference to devices, ranks, banks and arrays.
10. A memory controller converts processor requests into commands for the DRAM.
 - a. Why might it prefer to order requests so that they access different banks on consecutive memory operations, rather than the same bank?
 - b. When is it beneficial to continuously access the same bank?
 - c. Describe the difference between an open-page and closed-page row-buffer policy and the types of access patterns they benefit.
11. Describe the four classes in Flynn's taxonomy of computing systems. Explain where a multicore processor with a short-vector instruction set fits within Flynn's taxonomy?
12. Describe Amdahl's and Gustafson's law. Comment what is happening as we increase the number of cores in case as observed by Amdahl – how realistic it is to see the predicted results in real-world applications? What do you think, what is the implication of these laws on cloud computing?
13. Why is shared memory a useful concept in programming? Describe its disadvantages.
14. Considering discussed shared cache organisations, what are the benefits and downsides to each approach?

15. Consider inclusive, exclusive and non-inclusive cache policies. Can you develop a set of examples to demonstrate that under certain conditions one policy can outperform all others?
16. Describe MSI cache coherence protocol. Can you think of a software example/application problem where this becomes a bottleneck?
17. You hold data at address X when you snoop a BusRd transaction from the bus. What actions do you take with the MSI cache coherence protocol if you start in:
 - a. State M?
 - b. State S?
 - c. State I?
18. [2018 Paper 5 Question 3](#), part b.
19. Summarize the main message from lesson 9 in 1-3 sentences?
20. Summarize the main message from lesson 10 in 1-3 sentences?
21. Summarize the main message from lesson 11 in 1-3 sentences?
22. Summarize the main message from lesson 12 in 1-3 sentences?

Save your answers into MS Teams or email them to me. Please use the following naming pattern:

ICA_Supervision_3_Answers_<last name>_<first name>_Michaelmas_2026

Send your answers as a pdf, doc, image, or any other format of a document for which there exists an easily available software to open.

Jasmin JAHIC
jj542@cam.ac.uk
<https://www.cl.cam.ac.uk/~jj542/>