

Two hours

**UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE**

Mobile and Energy Efficient Systems

Date: Tuesday 21st May 2019

Time: 09:45 - 11:45

**Please answer all THREE Questions
Each Question is worth 20 marks**

© The University of Manchester, 2019

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

[PTO]

1. Questions concerning the ARM architecture

a) What is a load-store architecture?
What are the main characteristics of a load-store architecture? (2 marks)

b) The ARM architecture provides very good code density.
Let us consider the following arithmetic instruction:
`ADD r0, r1, r2`
Which additional operation(s) can the ARM architecture provide together with a data processing instruction?
Give for each operation(s) a use case where this improves code density. (4 marks)

c) ARM processors such as the ARM7TDMI can execute two instruction sets: standard-32bit ARM instructions and 16-bit Thumb instructions.
Discuss the main factors that impact code size, processor performance and processor power consumption when using Thumb instructions instead of standard 32-bit ARM instructions. (4 marks)

d) Many mobile systems have multicore CPUs.
Why does this help reduce power consumption? (3 marks)

e) Describe briefly the idea of implementing case statements as jump tables in ARM!
Provide a small code example which could be used in a SWI/SVC handler. (7 marks)

2. Questions about memory, memory management, and cache

- a) Why is it necessary/useful to have several levels of memory in a system? Give an example for a system with at least four levels of memory! (4 marks)
- b) In some embedded systems, tightly coupled on-chip RAM is used in preference to a cache. Give the most important reasons for this. (2 marks)
- c) Given is a system using 32-bit physical and virtual address space providing one level of cache of size 64 KB. What memory resources are required for the tag memory when using :
- i) a direct mapped cache with 64 bytes cache line size (2 marks)
 - ii) a two-way set associative cache with 128 bytes cache line size (2 marks)

Always explain your answers.

- d) Explain briefly what is stored in the tag memory and how that is used to determine if the present memory address is located in cache! (2 marks)
- e) Given is a 64-bit wide DDR memory module with 8 banks. Each bank provides 8,192 (8K) rows and rows have a size of 1KB. The memory is running at 1200 MHz. Consecutive column memory read accesses can be issued every 8 clock cycles for transfers of burst-8 transfers) and changing a read row memory address or changing a bank costs a penalty of 8 extra clock cycles (i.e. 16 clock cycles in total for a burst-8). Furthermore, we run a memory refresh cycle once every second which takes 100 ms. With this:
- i) What is the capacity of the memory? Explain your answer! (2 marks)
 - ii) How many bytes can be read in one second? It is sufficient to give this as a relative value with respect to peak read performance (in %). Explain your answer! (2 marks)
 - iii) What is the relative performance when reading random 64-byte sequences (e.g., individual cache lines)? Explain your answer! (2 marks)
 - iv) Estimate the relative performance when reading randomly across the entire memory 128-byte long sequences (e.g., individual larger cache lines)? Explain your answer! (2 marks)

3. Questions concerning design space exploration

You are asked to design a system that has to execute the following programs on an ARM processor (similar to discussed in the lectures):

program	number of instructions (in 32-bit ARM mode)	code size	
		32-bit ARM	Thumb mode
P1	100 M	40 KB	30 KB
P2	200 M	200 KB	150 KB
P3	500 M	400 KB	300 KB

Each program is executed once and one after another (i.e. first P1 then P2 then P3). Programs can run either in ARM-32-bit mode or Thumb mode. Additionally, programs can only run entirely from fast on-chip memory or slow off-chip memory.

To build the system, you have a CPU and can select different memory configurations:

component	speed	energy	cost
CPU	500 MHz	1 nJ ^{*)}	1 USD
on-chip memory (16-bit)	500 MHz	1 nJ ^{*)}	1 USD/10 KB
off-chip memory (16-bit)	100 MHz	10 nJ ^{*)}	1 USD/100 KB

^{*)}: the energy is only used in active (memory and ARM/Thumb CPU) cycles

For example, a system with a CPU, 20 KB on-chip memory and 300 KB off-chip memory costs $1+2+3=6$ USD (other cost factors are ignored here). When executing from slow off-chip memory, reading a single 32-bit ARM instruction accounts for 2×10 nJ for the memory subsystem and 1 nJ for the CPU execution (static power is omitted in this question). For simplicity, we don't consider caches or the impact of load/store instructions and we look only into instruction storage.

- Discuss briefly the usefulness of Thumb instructions in this system when running on the fast 16-bit on-chip memory and the slow 16-bit off-chip memory! Quantify where appropriate! (4 marks)
- What would be the cheapest system? How much energy and execution time does that system take? Explain your answers! (5 marks)
- What would be the fastest system configuration? How much execution time and energy does this system take and how expensive is this system? Explain your answers! (5 marks)
- What would be the most energy efficient system configuration that you can build for 20 USD? How much energy and execution time does this system take? Explain your answers. Note again that a program should be entirely in either on-chip memory or off-chip memory. (6 marks)