<u>Two hours</u>

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Mobile Systems

Date: Tuesday 31st May 2016

Time: 14:00 - 16:00

Please answer any THREE Questions from the FOUR Questions provided

---

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

[PTO]

1. Questions concerning the ARM instruction set

   a) The design of the ARM instruction set was greatly influenced by the RISC (Reduced Instruction Set Computer) ideas being put forward at the time of its development. Which of the following aspects are characteristic features of RISC processors and which are not? Which aspects are more specific to ARM?

   (In each case give a sentence explaining what the feature is and say whether it is a general RISC characteristic or specific to ARM. Where appropriate, give an example.)

      i)   conditional execution of all instructions
      ii)   a large regular register file
      iii) fixed length instruction encoding
      iv)   support for a second compressed instruction set
      v)   a load-store architecture
      vi)   single cycle execution of all instructions
      vii) microcoded instructions
      viii) auto-indexed addressing modes
      ix)   single instructions that load or store many registers
      x)   combined shift and add instructions

   (10 marks)

   b) The following ARM assembly code sequence could be generated by a compiler:
   ('...' indicates further instructions or constants)

```
        ADR  r1, TABLE
        CMP  r0, #N ; TABLEMAX
        LDRLE  pc, [r1,r0,LSL #2]
        ... ; statementsD
        B  EXIT
  TABLE DCD  L0
        DCD  L1
        ...
        DCD  LN
  L0    ... ; statements0
        B  EXIT
  L1    ... ; statements1
        B  EXIT
        ...
  LN    ... ; statementsN
  EXIT ...
```

   i) Sketch source code (e.g., written in C) that would result in the given assembly code.   (3 marks)

   Explain the operation and role of the following aspects of the code:

   ii) the 'ADR' in the 1st instruction   (1 mark)

   iii) the 'LE' (as part of 'LDRLE') in the 3rd instruction   (1 mark)

   iv) the 'LSL' in the 3rd instruction   (1 mark)

   v) What has to be changed when using branch instructions (B or BL) inside the table instead of the DCD directives? Write down the assembly code!   (4 marks)

2. Questions concerning programming the ARM architecture

   a) When do we use the branch command (`B`) and when the branch with link command (`BL`)? (2 marks)

   b) The ARM architecture from the lab has various sources for exceptions and some of them can occur at the same time.

      i) Which events can never occur simultaneously and why? (2 marks)

      ii) Describe a scenario that results in the greatest possible clash of simultaneous occurring events. (3 marks)

   c) Describe the steps that have to be carried out by a (simple) operating system running on an ARM core to swap between two tasks (running in user mode). The swapping shall be triggered by a timer interrupt event. (8 marks)

   d) The SWI (called SVC on later architectures) allows to pass an immediate value to the SWI (or SVC handler). For example, `SVC # &42` is encoded as `EF 00 00 42`. How do you load an 8-bit value that is passed with an SWI or SVC command into register R0? (2 marks)

   e) Write a program in ARM assembly language that computes the (even) parity of the content of register R0. Note that this is identical to a bitwise XOR over all 32 bits. The result shall be stored in register R1 (for example at bit position 0). (3 marks)

3. This question is concerned with using ARM Thumb in low-power system design.

   a) Assuming that a Thumb program is typically 70% of the size of the equivalent ARM program, estimate the relative performance of the Thumb and ARM programs when both are running from either:

      i) zero wait-state 32-bit on-chip RAM, or (3 marks)
      ii) slow 16-bit off-chip memory. (3 marks)

   b) In a particular complex low-power application based around a 20 MHz ARM7TDMI core, some speed critical routines are held in zero wait-state 32-bit on-chip RAM while the rest of the software runs from 100 ns 16-bit off-chip memory.

      Describe a procedure based on the ARM toolkit for identifying the routines which should be run from on-chip memory. (3 marks)

c) What are the performance and power-efficiency benefits of the on-chip memory? (Quantify where you can.) (2 marks)

d) What are the power-efficiency benefits of using Thumb code in the off-chip memory? (3 marks)

e) Why do we have to return to ARM 32-bit mode when we are in 16-bit Thumb mode and an exception occurs? (3 marks)

f) Describe the mechanisms employed in the ARM7TDMI and ARM9TDMI to support Thumb code, highlighting and explaining any differences between how the two processors provide this support. (3 marks)

4. 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 memories. (4 marks)

b) What can be changed in the cache organization to save power consumption? Explain briefly why this saves power. (2 marks)

c) What can be changed in the cache organization to increase the hit rate without making the cache memory size larger? How does this impact implementation cost? (2 marks)

d) This question is related to paged memory management as it is used in ARM.

   i) Sketch the organisation of a simple 1-level paged memory system with 32-bit address space and 4Kbytes page size and describe the principles of its operation. (4 marks)

   ii) How big is the page table (in terms of bits)? (1 mark)

e) Describe briefly how a hard disk can be used to provide more memory than what is physically available as RAM? (3 marks)

f) Sketch the basic organisation of a Protection Unit for 8 memory regions as used in ARM and describe the principles of its operation? (4 marks)