Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Mobile and Energy Efficient Systems

Date:	Tuesday 22nd May 2018
Time:	09:45 - 11:45

Please answer ALL THREE Questions

Each Question is worth 20 marks.

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

[PTO]

1. Questions concerning programming the ARM architecture

   a) Write a program in ARM assembly language that computes the number of bits that are set in register R0. The result shall be stored in register R1.
   (3 marks)

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

   c) Write ARM assembly code to perform the following array initialisation function (Assume that each integer is four bytes): (5 marks)

```
function set_x42(int[] a, int elements)
   for(i=0;i<elements;i++)
      a[i] = 0x42
```

   Additionally, assume that `a` is stored in memory and that the corresponding start address is in `R0`. Respectively, `elements` is stored in memory referenced by `R1`. Note that you don't have to save any registers you might overwrite.

   d) 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)

   e) The SWI (called SVC on later architectures) has an immediate value which can be passed 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 in the SWI handler? (2 marks)

2. Questions concerning heterogeneous computing

Let us consider a heterogeneous computing system consisting of a single core 32-bit ARM CPU running at 1 GHz and an FPGA running at 200 MHz.
This system shall run the following problem given as pseudo code :

```
1  for (int k=0; k < 1,000; k++)     //  a thousand iterations
2   {
3      for (int i=0; i < 1,000,000; i++)     //  a million iterations
4        {
5          p_random = p_random + count_the_number_of_1_bits(p_random);
6          if p_random == 0 then
7              p_random = 42;
8        }
9      $\alpha$ = (double)p_random / $2^{32}$
10     for (int i=0; i < 20,000,000; i=i+2)     //  ten million iterations
11       {
12         x = vector_array[i] $\cdot cos(\alpha) -$ vector_array[i+1] $\cdot sin(\alpha)$;
13         y = vector_array[i] $\cdot sin(\alpha) +$ vector_array[i+1] $\cdot cos(\alpha)$;
14         vector_array[i] = x;
15         vector_array[i+1] = y;
16       }
17  }
```

The function count_the_number_of_bits counts the number of bits that are set in the input argument. You can use the result from Question 1a) for count_the_number_of_bits or estimate a sensible number and reason your estimate. Assume that the processor computes all arithmetic instructions in a single RISC machine cycle regardless of the arithmetic operation or the data type used (e.g., integer of floating point).

a) Estimate the execution time of the problem when running on the ARM core only! Make sensible assumptions on execution times for all the functions and reason your assumptions briefly! There are various possibilities to compute trigonometric functions (e.g., *sin* or *cos*) and you can decide for any (e.g., an approximation using a polynomial of degree 5). Do not optimise the given code at this point. (8 marks)

b) How does your execution changes if you compute the trigonometric functions outside the for-loop starting at line 10? Explain your answer! (2 marks)

c) Let us now consider that we execute the problem using an FPGA instead of the ARM. Analyse the for-loops in the given code and explain if these loops can be parallelised using loop unrolling or if dependencies prevent this! (3 marks)

d) With the knowledge of possible loop unrolling from the previous question, estimate the execution time if you execute the problem on an FPGA! Consider an implementation

that is optimized for performance.
It is not important to understand details of FPGA hardware design for this question. Assume for simplicity that the FPGA has unlimited logic and memory resources in the sense that the FPGA can accommodate the entire array. Explain your answer!
(5 marks)

e) Make a sensible suggestion on how the FPGA and the ARM could work together on the given problem to save resources! Explain briefly your suggestion!
(2 marks)

3. 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) How does increasing the cache line size impact cost (resources), performance and power consumption of a cache?
   Explain your answer! (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 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)

END OF EXAMINATION