

Two hours

**UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE**

Mobile and Energy Efficient Systems

Date: Friday 19th May 2017

Time: 14:00 - 16:00

Please answer ALL THREE Questions

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

[PTO]

1. Questions concerning energy efficient computing

- a) How can a higher level of parallel running hardware instances (e.g., multicore) help in saving power? In addition, describe a scenario where parallel running hardware fails to reduce power. (3 marks)
- b) Consider a scenario for a heterogeneous computing system that uses different kinds of programmable data processing units to provide high performance at lower energy consumption. Name three different kinds of data processing units that were discussed in the lectures. Describe a beneficial mode of operation and an advantage for each kind of data processing unit. Provide an example/use case for each kind of data processing unit where it performs particularly well. (6 marks)
- c) Describe briefly the ARM big.LITTLE concept and describe the three different ways of how the processor cores can be arranged. (4 marks)
- d) Discuss pros and cons for the compiler optimizations `function inlining` and `loop unrolling` with respect to energy efficient computing. (3 marks)
- e) The compact ARM Thumb mode results typically in binaries that are 70% of the size of an equivalent ARM program. Assume that an active ARM or Thumb execution cycle takes 400 pJ and that a stall cycle takes half of this (200 pJ). Furthermore, assume that we use a 16-bit memory for the instructions that consumes 600 pJ per read cycle. The memory allows reading one Thumb instruction each cycle or one 32-bit ARM instruction every other CPU cycle (which requires one CPU stall cycle per instruction fetch when using 32-bit ARM mode). With these assumptions, estimate the energy consumption ratio of running a program in either:
- i) 32-bit ARM mode
 - ii) 16-bit Thumb Mode

Justify your estimates. There is no need to use a pocket calculator and any rational number can be given as a quotient, if needed. (4 marks)

2. Questions concerning programming the ARM architecture

- a) Why is it required to restore the status register and the PC in one atomic step for returning from a SWI (also called SVC) system call? Give an ARM assembly example for an instruction that returns from a system call. (3 marks)
- b) Write ARM assembly code that performs the given matrix multiplication function for multiplying two square matrices A and B of a size given by their dimension. Assume that each number and the result values are stored in unsigned byte values. There is no need to write optimized code or to deal with range overflows of the result. Provide sensible documentation such as the register usage and comments. (14 marks)

```
function matrix_mul(char[] A, char[] B, char[] result, int dimension)
    for(i=0; i<dimension; i++)
        for(j=0; j<dimension; j++)
            for(k=0; k<dimension; k++)
            {
                result[i][j] += a[i][k] * b[k][j];
            }
}
```

Assume that `dimension` is stored in R0 and that the start memory addresses of the matrices A, B and `result` are stored in R1, R2 and R3 respectively. Note that it is not required to save any registers that you might overwrite; you only have to return from the function after computing the matrix multiplication.

- c) Give two possible performance optimizations for the given basic matrix multiplication function (see the code above) that works well for multiplying large matrices on a Raspberry Pi Model A that features an ARMv6Z. Describe briefly the operation for those two optimizations. (3 marks)

3. Questions about memory

- a) The two main approaches for memory management are segmentation and paging.
- i) Why is memory management useful for allowing multiple active programs?
(2 marks)
 - ii) What is the difference between a logical address (sometimes called virtual address) and a physical address?
(You do not have to discuss issues related to caches at this point.)
(2 marks)
 - iii) Describe the basic principles behind segmentation and paging.
How are physical addresses created and protected in each variant?
(You can draw figures and/or use text for answering this question.)
Discuss the main advantages and disadvantages of each variant. (8 marks)
- b) Most systems use a DRAM variant as the main memory. DRAM arranges data in an array of memory cells which is accessed by a row access select (\overline{ras}) and by a column access select (\overline{cas}).
- i) What is the difference on power and performance when
 - 1) incrementally changing row access (going through \overline{ras} cycles) and
 - 2) incrementally changing col access (going through \overline{cas} cycles)?
(2 marks)
 - ii) Consider you have to connect a DRAM to a CPU. How would you allocate address signals for row select and for column select? Please justify your decision.
(2 marks)
- c) In some embedded systems, on-chip RAM is used in preference to a cache.
Give the most important reasons for this. (2 marks)
- d) 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)