UNIVERSITY OF MORATUWA

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

B.Sc. Engineering
2010 Intake Semester 8 Examination

CS4342 ADVANCED COMPUTER ARCHITECTURE

Time allowed: 2 Hours March 2015

ADDITIONAL MATERIAL: *None*

INSTRUCTIONS TO CANDIDATES:

1. This paper consists of **5** questions in **6** pages.
2. Answer any **4** questions.
3. Start answering each of the main questions on a new page.
4. The maximum attainable mark for each question is given in brackets.
5. This examination accounts for 50% of the module assessment.
6. This is a closed book examination.

   NB: It is an offence to be in possession of unauthorised material during the examination.
7. Only calculators approved by the Faculty of Engineering are permitted.
8. Assume reasonable values for any data not given in or with the examination paper. Clearly state such assumptions made on the script.
9. In case of any doubt as to the interpretation of the wording of a question, make suitable assumptions and clearly state them on the script.
10. This paper should be answered only in English.

Question 1 (25 marks)

Suppose, just after graduation you were hired by Apple Inc. Your team is responsible for designing the next processor for the 6th generation iPad to be released in 2018. iPad 6 is to be the thinnest tablet ever with high quality graphics, fast App performance, and extended battery life. As a preparation for the first project meeting, each team member is asked to think about how to implement various components of the processor.

(i) Briefly explain what type of a design you would recommend for each of the following items. Provide at least 2 justifications for each of your selections.

a) Number of processing elements (e.g., single core, multi core, or many core). [3]

b) Parallel execution of instructions (e.g., pipelines, vectors, multimedia extensions, many SIMD lanes, etc.). [4]

c) Static vs. dynamic scheduling of instructions. [3]

d) Memory hierarchy (e.g., number of levels, cache size, cache associatively, etc.). [6]

e) What design options would you recommend to reduce the power consumption of the overall iPad? Briefly discuss. [4]

(ii) After several performance tests it has been identified that the I/O sub-system on the 4th generation iPad was the primary performance bottleneck. The reason was that the waiting time for I/O was 60% of the total time. Therefore, one of your team members suggested that by speeding up the I/O system alone, it would be possible to gain an overall speedup of 2 times.

a) How much speed up in the I/O sub-system will be required to achieve the desired overall speed up?

You may use the Amdahl’s law given below for the calculation:

$$Speedup_{Overall} = \frac{1}{(1 - Fraction_{Enhanced}) + \frac{Fraction_{Enhanced}}{Speedup_{Enhanced}}}$$

[2]

b) Do you agree with your team member’s idea of gaining such a speedup by improving only the I/O sub-system? Briefly explain. [3]

Question 2 (25 marks)

(i) List 3 factors that limit compile time loop unrolling. [3]

(ii) Consider the following Assembly code. The values of `F10` and `R2` are predefined.

```
Loop:
      LD     F1,0(R1)
      LD     F2,100(R1)
      DADD   F3,F1,F2
      DADD   F4,F3,F10
      SD     F4,200(R1)
      DADDUI R1,R1,#-4
      BNZ    R1,R2,Loop
```

a) Identify all data dependencies in the above code. [4]

b) Calculate the number of clock cycles required by an iteration of the above loop. Use the following table for latencies of Floating Point (FP) operations in MIPS. [3]

Instruction Producing Result	**Instruction Using Result**	**Latency in Clock Cycles**
FP ALU operation	Another FP ALU operation	3
FP ALU operation	Store double	2
Load double	FP ALU operation	1
Load double	Load or Store double	0

c) Improve the above code to reduce the number of clock cycles. [3]

d) Recalculate the clock cycles required by an iteration of the new loop. [2]

e) Unroll the loop by a factor of 2 to further reduce the number of clock cycles. [8]

f) How much speedup can we gain per iteration compared to the original code? [2]

Question 3 (25 marks)

Following write up is extracted from the paper titled "The Greendroid Mobile Application Processor: An Architecture For Silicon's Dark Future" by Goulding-Hotta et al.

> "The GreenDroid architecture uses specialized, energy-efficient processors, called conservation cores, or c-cores, to execute frequently used portions of the application code.
>
> Each host CPU is a full-featured 32-bit, seven-stage, in-order pipeline, and features a single-precision floating-point unit (FPU), a multiplier, a 16-Kbyte instruction cache, a translation look-aside buffer (TLB), and a 32-Kbyte banked L1 data cache.
>
> Our frequency target of 1.5 GHz is set by the cache access time, and is a reasonably aggressive frequency for a 45-nm design. The tiles' L1 data caches are used to collectively provide a large L2 for the system. Cache coherence between cores is provided by lightweight L2 directories residing at the DRAM interfaces, which use the L1 caches of all the cores as a victim cache.
>
> In addition to sharing the data cache, the c-cores optionally share the FPU and multiplier with the CPU, depending on the code's execution requirements. Collectively, the tiles in the GreenDroid system exceed the system's power budget. As a result, most of the c-cores and tiles are usually power gated to reduce energy consumption."

(i) Based on the above description answer the following questions.

a) Briefly explain what is mean by "seven-stage, in-order pipeline". [3]

b) What is the functionality of the "translation look-aside buffer (TLB)"? [3]

c) Briefly explain what is mean by "cache coherence between cores is provided by lightweight L2 directories"? [3]

d) Draw the cache/memory hierarchy while labelling necessary components and capacities. No need to draw all the cores. [4]

e) Briefly explain what is mean by "power gated". [3]

(ii) On the same paper authors propose a mobile processor with a multiple cores, namely conservation cores (c-cores), to address the *utilization wall*. The utilization wall says that, with each process generation, the percentage of transistors that a chip design can switch at full frequency drops exponentially because of the power constraints. c-cores target the Android mobile-phone software stack and are automatically generated, highly specialized, and energy efficient.

a) Compare and contrast (i.e., similarities and dissimilarities) *utilization wall* and *power wall*. [3]

b) Briefly explain how do c-cores overcome the *utilization wall* while reducing the overall energy consumption of the mobile processor. [6]

Question 4 (25 marks)

(i) Briefly explain how RAW (Read After Write) hazards are resolved in the basic Tomasulo's algorithm. [5]

(ii) While Vector Mask Registers are available in Vector processors and GPUs they are usually not available in SIMD instruction sets such as SSE, MMX, and AVX. What is an advantage and a disadvantage of having Vector Mask Registers? [2]

(iii) NVIDIA Tesla K10 PCI card has 2 GPU accelerators. There are 32 SIMD lanes per processor. There are 48 SIMD processors per GPU. At most 1,024 threads per SIMD processor (block) can be executed at a given time.

a) Altogether, how many cores are in the Tesla K10 card? [3]

b) Suppose we want to run 25,000 threads on this system. Explain how such a large number of concurrent threads can run on the Tesla K10. [5]

(iv) On a given system 40% of the instructions are load and store. Miss penalty is 25 clock cycles. Instruction miss rate is 1% and data miss rate is 4%. If *n* instructions to be loaded and executed, how many clock cycles will be wasted due to memory stalls? Suppose Clocks Per Instruction (CPI) is 1. [4]

(v) Using suitable examples briefly describe any 3 advanced techniques used for optimizing cache (memory) performance in modern processors. [6]

Question 5 (25 marks)

Warehouse-Scale Computers (WSCs) are used to provide Internet-scale services such as search, social networking, video sharing, online shopping, email, and cloud computing. Some of the defining characteristics of WSCs include use of a large pool of commodity hardware, virtualization, task-level parallelism, and dependability via redundancy.

(i) Compare and contrast (i.e., similarities and dissimilarities) thread-level vs. task-level parallelism. Consider at least 4 factors. [4]

(ii) Computing equipment's contribute to ~30% of the total power consumption of a typical WSC. However, only 33% of this 30% is used by the CPUs. Is it worthwhile to make the CPUs used in WSCs even more power efficient? Briefly discuss. [3]

(iii) Consider a WSC with 25,000 nodes. Each node has 2 hard disks. If the Mean Time To Failure (MTTF) of a hard disk is 3 years, how many disk failures should be expected per day? [3]

(iv) To increase the performance of a WSC, multiple levels of architectural changes are required at different levels. Briefly describe one type of architectural change that can be introduced in each of application, storage and network levels. [9]

(v) WSCs co-locate Virtual Machines (VMs) to increase the resource utilization while reducing the power consumption. However, co-location also affects the Quality of Service (QoS) experienced by an application. Briefly explain how carefully designed stress tests, past statistics, and predictions can better schedule VMs while preserving the QoS? [6]

------------------------- END OF THE PAPER -------------------------