**Fall 2013: COMP 7300 Advanced Computer Architecture**

**Test 1 (100 pts) (1h:15)**

**LAST PAGE EXERCISE IS BONUS**

Grading policy:

¼ Credit for correct answer

**¾ Credit for well written and solid justification/facts/arguments.**

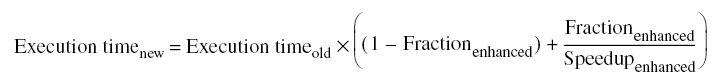
**A) Introduction (15 points)**

1. (**3 points**) This semester, we stressed that modern computer architecture focuses on meeting specific requirements of the target machine and aims to maximize performance within the followingkey constraints: \_\_\_\_\_\_\_\_\_\_\_\_, \_\_\_\_\_\_\_\_\_\_\_\_, and \_\_\_\_\_\_\_\_\_\_\_\_\_ [**Fill in the blank**]
2. **(12 points) Exercise**:

We consider a program ***P*** executed on an imaginary processor with a clock rate of **4 GHz**. This table provides the number of executions for each type of instruction and the number of cycles. Answer the questions based on this table.

|  |  |  |
| --- | --- | --- |
|  | Number of instructions | (CPI) Clock cycle Per Instruction |
| Load/Store | 400 | 6 |
| Arithmetic | 500 | 4 |
| Logical | 300 | 2 |

1. **(1.5 point)** How many clock cycles in total are needed to execute Program ***P***?
2. **(1.5 point)** What is the average CPI (average number of clock cycles per instruction) for Problem ***P***?
3. **(1 point)** What is the latency (in ns) of **one** *Load/Store* instruction?
4. **(1 point)** How long does Program ***P*** take to execute (in **microseconds**)?
5. **(3 points)** To speed up program P, **John** proposes to speed up **only** Load/Store instructions by a factor of 6.



Using Amdahl’s formula, what is the new execution time for Program P?

1. **(3 points)** To speed up program P, **Paul** prefers to p to speed up **all instructions OTHER** than Load/Store instructions by a factor of 2. Using Amdahl’s formula, what is the new execution time for Program P?
2. **(1 point)** Which improvement (John’s or Paul’s) yields the best overall speed-up?

**B) (65 points) Memory Concepts**

1. **(3 points)** Two key **requirements** lead to build a memory as an **hierarchical** memory system. **Cite** these two requirements?
2. **(3 points)** Two key **facts** lead to build a memory as an **hierarchical** memory system. **Cite** these two facts?
3. **(20 points)** We consider a **2-way** associative cache with size 64 bytes and a block size of 8 bytes. Assume 16 bit addresses.
4. **(2.5 points)** How many bits does the offset have?
5. **(2.5 points)** How many sets does the cache have?
6. **(2.5 points)** How many bits does the index have?
7. **(2.5 points)** How many bits does the tag have?
8. **(10 points)** The first column on the table below shows the addresses (to bytes) that a CPU generates. You may omit leading 0s. First, complete the table and then tell whether each address is a hit or a miss assuming that the LRU replacement policy is used.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Address (Hexa)** | **Address (Binary)** | **Tag (binary)** | **Index** | **offset** | **Miss/Hit** |
| 0x98 |  |  |  |  |  |
| 0xBF |  |  |  |  |  |
| 0x78 |  |  |  |  |  |
| 0xAF |  |  |  |  |  |
| 0x88 |  |  |  |  |  |
| 0xA8 |  |  |  |  |  |
| 0xB8 |  |  |  |  |  |
| 0xCF |  |  |  |  |  |
| 0xAC |  |  |  |  |  |
| 0x9F |  |  |  |  |  |

1. **(20 points)** Below is a *questionable* (i.e., maybe not true) table summarizing the characteristics memory chips:

|  |  |  |
| --- | --- | --- |
| **Memory Type** | **Access Time** | **Cost per GB** |
| DSRAM | 50 ns – 70 ns | ~$50 |
| Flash | 20 ns – 35 ns | ~$2 |
| SRAM | 0.5ns – 2.5 ns | $2,000 - $5,000 |
| Magnetic Disk | 5 ms – 20 ms | $3 |

1. **(4 points)** This table may contain some *questionable* (i.e, doubtful, not true) facts. Which cells/facts are questionable? Why?
2. **(5 points) Assume for now that the table is accurate.** Based on the table above, which memory type would most likely be used for L1 caches (closest to CPU)?
3. **(5 points) Assume that the table above is accurate (true)**. Based on the table above, which type of memory should be used for the main memory?
4. **(6 points) Assume that the table above is accurate (true),** how many levels will this memory system have? Cite each memory level and the memory type used at each level
5. **(17 points)** Paul and John are arguing about the size of a cache block. Paul wants a larger block size.
   1. **(3 points)** John states that a larger block size leads to a higher miss penalty. Is John right?
   2. **(4 points)** John states that a larger block size will **always** decrease the miss rate. Is John right?
   3. **(4 points)** John argues that *Early Restart* will halve (divide by 2) on average the miss penalty regardless of the block size.
   4. **(3 points)** Paul argues that if *Critical-Word-First* is used, the miss penalty will not increase with a larger block. Explain what *Critical-Word-First* is. Is Paul right?
   5. **(3 points)** John argues that if *Critical-Word-First* is used, a larger block will decrease the miss rate for program with high special locality. Is John right?

**Exercises (30 points)**

1. **(22 points) Exercise 1**: **In addition** to the data block storage, a cache must store a validity bit and a tag for each data block. The validity bit and the tag incur a ***storage overhead***. The *storage overhead* is expressed as:

This exercise explores the storage overhead for direct-mapped and n-way associative caches.

We assume 16-bit addresses with a cache ***C*** that accommodates 32 KB of data. The block size is16-words (one word is four bytes).

* 1. **(8 points)** How many total bits are required if the cache ***C*** is a direct-mapped cache?
  2. **(2 point)** What is the storage overhead for the above direct-mapped cache?
  3. **(8 points)** How many total bits are required if the cache ***C*** is a 4-way associative cache?
  4. **(2 points)** What is the storage overhead for the above 4-way associative cache?
  5. **(2 points)** Which method (direct-mapped or 4-way associative) offers the best overhead?

1. **(10 points) BONUS Exercise 2:** The objective of this exercise is to derive the formula for the average number ***AvgCPI*** of clock cycles : ***AvgCPI*** ***= CPI + mr \* mp*** where *mr* is the miss rate, *mp* is the miss penalty, and *CPI* is the number of clock cycles per instruction for an **ideal** cache (*mr* = 0).
2. **(1 point)** Suppose that the cache is **ideal**, express ***AvgCPI*** as a function of CPI.
3. **(1 point)** For **all** **the following** questions, we assume that the cache is not ideal and that a program has ***n*** instructions. Let us compute the number ***h*** of instructions that experience a cache hit. Express ***h*** as a function ***n*** and the miss rate ***mr***.
4. **(1 point)** Express the number of clock cycles that ***h*** instructions (hits) take.
5. **(1 point)** Let us compute the number ***m*** of instructions that have a cache miss. Express ***m*** as a function ***n*** and the miss rate ***mr***.
6. **(1 point)** Express the number of clock cycles that ***m*** instructions (hits) take.
7. **(3 points)** Express the average CPI as a function of ***h***, ***m***, ***n***, ***mp***, and ***CPI.***
8. **(2 points)** Derive from Question f) the formula ***Average CPI = CPI + mr \* mp***