

Name: Drake Bridgewater

CS/ECE 472/572 Midterm 2

ID#: 931-566-291

*All answers should be typed directly below the questions. All work should be shown. Answers with no work shown will be considered cheating. The exam should be typeset via L<sup>A</sup>T<sub>E</sub>X. All questions should be included in the exam.*

*This is an open book exam. As it is a take home exam, you are on the honor code. Use your book, use your notes, don't use your friends.*

*There are 200 points possible on this exam.*

*Be detailed in your answers. Remember, this is a take home exam, so implied answers will not be considered. Take the time and make everything explicit!*

1. (10 points) 6.7: The timing diagram in Figure P6.7 illustrates a system in which operations occur as three consecutive clock cycles. Actions taking place in clock cycle 1 are scalable; that is, if the clock cycle time changes, the actions can be speeded up or slowed down correspondingly. In cycle 2, the action process 1 requires 25 ns and in clock cycle 3 the action process 2 requires 32 ns. If the clock cycle is less than the time required for process 1 or processes 2, then one or more wait cycles have to be inserted for the process to complete.

What is the time to complete an operation if the clock cycle time is:

- (a) a. 50 ns:  $50 * 3 = 150$  ns
- (b) b. 40 ns:  $40 * 3 = 120$  ns
- (c) c. 30 ns:  $30 + 30 + 60 = 120$  ns
- (d) d. 20 ns:  $20 + 40 + 40 = 100$  ns
- (e) e. 10 ns:  $10 + 30 + 40 = 80$  ns

Answer copied from Homework 6 by Drake Bridgewater and Ryan Phillips [2]

2. (10 points) 6.13: A computer has the following parameters:

Operation	Frequency	Cycles
Arithmetic/logical instructions	65%	1
Register load operations	10%	5
Register store operations	5%	2
Conditional branch instructions	20%	8

If the average performance of the computer (in terms of its CPI) is to be increased by 20% while executing the same instruction mix, what target must be achieved for the cycles per conditional branch instruction?

$$.65 * 1 + .1 * 5 + .5 * 2 * .2 * 8 = 2.75(baseCPI) \quad (1)$$

We want to increase this metric by 20%,  
which would correspond to a CPI of 2.2( $2.87 * .8$ )

$$2.2 = .65 * 1 + .1 * 5 + .5 * 2 * .2 * x \quad (2)$$

$$x = 5.25 \quad (3)$$

If cycles per conditional branch instruction could hit 6 instead of 8, we would achieve the performance goal.

Answer copied from Homework 6 by Drake Bridgewater and Ryan Phillips [2]

3. (10 points) 6.30: What are the relative advantages and disadvantages of arithmetic, geometric, and harmonic means as methods of averaging benchmarks?

The arithmetic mean though it is simple calculation a single outliers will lead to heavily screwed data. Whereas using the harmonic mean works best when you have a majority of values with with a few outliers with significantly higher values. As for the geometric mean, SPEC uses it and indicates the central tendency or typical value of a set of numbers.

These answers are summarized from [3, The Economist at Large] and [4, Wikipedia]

4. (10 points) 6.31: For two benchmarks,  $x$  and  $y$ , show that their arithmetic mean is always higher than, or the same as, the geometric mean.

The Arithmetic mean is defined as

$$A_{Arithmetic} = \frac{1}{n} \sum_{i=1}^n X_i \quad (4)$$

The Geometric mean is defined as

$$A_{Geometric} = (\prod_{i=1}^n X_i)^{1/n} \quad (5)$$

To show that the arithmetic mean is always greater we will show that the arithmetic mean is greater than the geometric mean for  $n = 2$

Hypothesis:

$$\frac{x+y}{2} > \sqrt{xy} \quad (6)$$

$$\left(\frac{x+y}{2}\right)^2 - xy = \left(\frac{x+y}{2}\right)^2 > 0 \quad (7)$$

The arithmetic mean would be the largest when ever value,  $\alpha$  is the same

$$\alpha = \frac{\sum \alpha}{n} = \sqrt[n]{\alpha \alpha \dots \alpha} = (\alpha^n)^{1/n} \geq \sqrt[n]{\prod_{i=0}^n x_i} \quad (8)$$

Answer copied from Homework 6 by Drake Bridgewater and Ryan Phillips [2]

5. (10 points) 6.32: The SPEC benchmarks present results with respect to a standard machine by normalizing the benchmarks. That is, a set of benchmarks is run on a reference machine and the times obtained for each for the benchmarks. When a test machine is benchmarked, its times are divided by the results on the reference machine. What are the advantages and disadvantages of giving benchmarks with respect to a reference machine?

The disadvantages of using a reference machine for calculating benchmarks is that the reference machine is not always comparable to the system you are running and

6. (10 points) 7.13: What modifications would have to be made to the architecture for the computer in figure P7.12 to implement predicated execution like the ARM?

The computer would need a flag from the Z bit of the register file to determine if it needs branch predication which would be best determined after the instruction is read therefore you want this Z bit to trigger the branch

7. (10 points) 7.19

- (a) **Is it true that a larger number of register in any architecture is always better than a smaller number?** It is not true it just defines difference architecture that is being used for Intel processors they only use a single register while other processors such as ARM have multiple. If you create an architecture that does not require registers then adding register will do nothing for it, but if you do use registers then adding register will increase performance to a certain degree. There is the possibility that you can have too many registers; one downside will be when you have too many register to transfer on a single bus line causing the need for additional cycles for executions. Each architecture is designed differently which is a large factor as to the difficulties of benchmarking.
- (b) **What limits the number of register that can be implemented by any ISA?** The number of register is limited by price, bus size, and instruction size.
- (c) **What are the relative advantages and disadvantages of dedicated registers like the IA32 architecture compared to general purpose registers like ARM and MIPS?** As Swanson Technologies [5] states, when Intel was originally creating there processors they had general purpose registers because you can reduce the physical size and each register was being used for a specific purpose. This is great for software engineers if they take advantage of the optimizations, but as he states may software engineers are unaware of these optimizations therefore they are not used properly. This article also states that having consistent register use brings better compression.
- (d) **If you have an m-bit register select field in an instruction, you can have more then  $2^m$  registers. There are in fact, ways round this restriction. Suggest ways of increasing the number of register beyond  $2^m$  while keeping an m-bit register select field.**

8. (10 points) 7.20

9. (10 points) 9.12: How is data in main store mapped on to each of the following: a direct-mapped cache, a fully associative cache, and a set-associative cache?

**Direct Mapped:** Each location in main memory goes with one entry in the cache. This is one of the simplest types of caches to design.

**Fully Associative:** Has no restrictions on where data can be located. It uses a tag and a valid bit to check and retrieve the data; done in parallel.

**Set-Associative:** Set-associative is a combination of fully associative and direct mapped. Lines are grouped into sets. A given address is mapped into a set (like in direct-mapped), and within the the set the lines are organized like that of a fully

associative scheme.

Answer copied from Homework 5 by Drake Bridgewater and Ryan Phillips [1]

10. (15 points) 9.21: The cache system can be located between the CPU and the MMU or between the MMU and the system random access memory. what factors determine the optimum location of cache memory?

A system that access has the cache between the MMU and Main memory allows a program to treat its memory space as a single contiguous block and the MMU takes care of the mapping of virtual memory to physical address. Whereas if the cache is between the processor and MMU the processor addressees cache directly without needing to go through the MMU meaning that address don't have to be translated by the MMU saving precious time, but the one downside is that all the processes have the same virtual address space. Overall to determine the optimal location of the cache you would want to know the weather typical executions need to access main memory very often and how much memory a process is requesting.

11. (10 points) 9.23: When a CPU writes to the cache, both the item in the cache and the corresponding item in the memory must be updated. If data is not in the cache, it must be fetched from memory and loaded in the cache. If  $t_1$  is the time taken to reload the cache on a miss, show that the effective average time of the memory system is given by  $t_{avg} = ht_c + (1 - h)t_m + (1 - h)t_1$

$$\begin{aligned}t_{avg} &= hits + misses \\ hits &= ht_c \\ misses &= miss\ time + time\ fetch\ memory \\ misses &= (1 - h)t_m + (1 - h)t_1 \\ t_{avg} &= ht_c + (1 - h)t_m + (1 - h)t_1\end{aligned}$$

Answer copied from Homework 5 by Drake Bridgewater and Ryan Phillips [1]

12. (10 points) 9.26: A system has a level 1 cache and a level 2 cache. the hit rate of the level 1 cache is 90% and the hit rate of the level 2 cache is 80%. An access to level 1 cache requires one cycle, an access to level 2 cache requires four cycles, and an access to main memory requires 50 cycles. What is the average access time?

If we take a sample of 100 accesses to memory/cache. 90% of these call will be caught by L1 cache and will require 1 cycle to complete, leaving 10 accesses which 80% will be caught by L2 cache requiring 4 cycles, lastly if the CPU misses on L1 and L2 then it will have to fetch from main memory requiring 50 cycles.

$$100_{total\_access} = (100 * 90\%_{hits} * 1_{cycles}) + (10 * 80\%_{hits} * 4_{cycles}) + (2 * 100\%_{hits} * 50_{cycles})$$

$$(90 + 22 + 100)/100 = 2.3 \text{ Avg. Cycles}$$

Answer copied from Homework 5 by Drake Bridgewater and Ryan Phillips [1]

13. (10 points) 9.35: A 64-bit processor has a 8-MB, four-way set-associative cache with 32-byte lines. How is the address arranged in terms of set, line and offset bits?

offset = 5 bits since each line is 32 bytes, line = 16 bits, set = 43 bits.

14. (10 points) Assume a 64-bit virtual address and a 64-bit physical address. The page size is 4KB. How many total entries are there in the page table? Express your answer in powers of 2.

A 64 bit address can address  $2^{64}$  bytes in a byte addressable machine. Since the size of a page is 4KB ( $2^{12}$ ) the number of addressable pages is  $\frac{2^{64}}{2^{12}} = 2^{52}$

15. (15 points) Describe the ARM pipeline. What benefit does this serve over a single cycle flow through processor?

16. (15 points) Describe a method of programmatically determining cache line size as well as cache sizes.

Write to an array of increasingly larger sizes each time the array size increases time the write cycle of all the elements. You will want to start with small sizes such as 1KB and slowly increment up to possible cache sizes. When you see a change in the time it takes to write to the array you know that the cache is between the current size and the previous size.

17. (25 points) On modern CPUs, there are often multiple cores. Describe ways in which you believe cache design had to change to deal with this situation.

CPUs in general need a cache to enable quicker operations, but when you add an additional core how is that cache to be used. Modern CPUs have multiple cores and to accommodate this movement CPU manufactures are giving each core it own cache which is exceptional fast then they will have an L2 cache that is connected to multiple cores.

The cache design changed by giving each core there own cache and when there is a missed request L2 cache is checked. This L2 cache is located between and connected

to the other cores. The reason they have to have a connected cache is each core is performing operations on data that is most likely related to what the other processors is running.

## References

- [1] Drake Bridgewater and Ryan Phillips. Homework 5. [https://github.com/drakebridgewater/cs472\\_Computer\\_Architecture/raw/master/Assignment5/CS472\\_HW5\\_bridgewd\\_philliry.pdf](https://github.com/drakebridgewater/cs472_Computer_Architecture/raw/master/Assignment5/CS472_HW5_bridgewd_philliry.pdf).
- [2] Drake Bridgewater and Ryan Phillips. Homework 6. [https://github.com/drakebridgewater/cs472\\_Computer\\_Architecture/raw/master/Assignment6/CS472\\_HW6\\_bridgewd\\_philliry.pdf](https://github.com/drakebridgewater/cs472_Computer_Architecture/raw/master/Assignment6/CS472_HW6_bridgewd_philliry.pdf).
- [3] Andrew Matuszak. Differences between arithmetic, geometric, and harmonic means, 2010. <http://economistatlarge.com/finance/applied-finance/differences-arithmetic-geometric-harmonic-means>.
- [4] Multiple. Means, 2014. <https://en.wikipedia.org/wiki/Mean>.
- [5] William Swanson. The art of picking intel registers, 2003. <http://www.swansontec.com/sregisters.html>.