## Amer Din / aad180004

## CS 2340 Computer Architecture

## Homework 8: Computer Architecture Lecture by Yale Patt

1. List and briefly describe the 7 “layers” involved in using computers to solve problems.

The 7 layers involved in using computers to solve problems are comprised of the problem, algorithm, program, ISA, microarchitecture, circuits and electrons. The problem is essentially the layer that leverages a proper use case to combat a real-world issue in a user’s life. For example, browsing the web requires specific programs such as Google Chrome or Safari to access sites and data. The algorithm layer makes the best optimal use of the underlying functions involved in delivering the programs and its dedicated purpose. The program layer allows diversification to the user in how they want to access and manipulate data; For example, a calculator is a prime example of a program because it allows for a user to access real time operations to numbers. ISA layers essentially remark how the instructions set architecture is designed; For example, the intel x86 processor would be a prime ISA. The microarchitecture is essentially the way of which the internal hardware is organized. Circuits coincide with the electron layer in which how the power is allocated among the transistors and effect the heating of the computer. The last layer of the electron is the raw horsepower of powering the computer utilizing transistors.

1. Briefly describe microprocessor evolution from 1971 to today in terms of number of transistors and processor speed.

Beginning in 1971, transistors emerged as small and compact on a chip. The intel 4004 utilized 2300 transistors and ran at 106 KHz. 21 years later, the Pentium chip utilized 3.1 million transistors and ran at 66MHz This timeline followed the implementation of Moore’s law, the notion that the number of transistors on each chip will double every 2/3 years. In terms of processing speed,

1. As time went on, did more and more transistors go towards processing or memory?

As more time went on, the transistors emphasized on chip memory, cache at a rate that is exponential. However, the processing has increased linearly as time has gone by.

1. What significant change in processor design caused a more balanced use of transistors between cache and processing?

The significant change that enabled balance uses of transistors between cache and processing was the addition of multi-core processors. This enabled proportional growth but a setback of power.

1. What is cache?

Cache is essentially is a hardware cache that enables the computer to reduce the average cost to access data from the main memory.

1. Now that we have multi-core processors, have we learned how to program them? Why or why not?

With the addition of multi-core processors, we have not learned how to program them. The reason being is because of the setback between power consumption and also due to the intense speed of adding cores without optimization; With more cores, the optimization of each core and its programming is not applicable currently. In addition, there is the issue of limited availability of bandwidth.

1. Describe Professor Patt’s predictions for the next phase (Phase 3) of microprocessor design.

Professor Patt’s prediction consists of variable cores, instead of identical ones. He also elicits that some parts of the chip will be reconfigurable and power allocation will be much more sustainable.

1. What is Step 1 of Professor Patt’s vision of the future of microprocessors?

Professors Patt’s predictions for the next phase is beginning with breaking the layers. Instead of 8 individual layers, he insists they should be interconnected with each other, allowing more engineers to be more conscious and aware of setbacks in both hardware and software components.

1. What good things does Professor Patt believe will follow step 1?

Professor Patt believes following step 1 will lead to good things such as having more than one interface, “Organic” run-time systems and ILP cores. The idea having more than one interface will allow programmers to understand the underlying hardware configurations.

1. Programs can be written on a platform that is easy to use but may not have great performance, or they can be written in a more challenging way for optimal performance. Discuss the trade-offs of each approach.

In regard to the first methodology, programs that are easy to use but that do not have great performance can lead to more usage and ease of use in the implementation. However, with this idea, the tradeoff is that the efficiency will be lagging and there will be less optimization that makes the best use of power and processing. The goal in the future would be to bridge the gap between ease of use and hardware performance by bridging the 2 interfaces. As for the second methodology, when programs are written in a more challenging way with optimal performance, the benefits can be that the program is very strong in delivering capabilities, while also accounting extensively for power and processing cohesively. The program most likely makes the best usage of algorithms and all other 7 layers so that the program can rarely crash. The trade-off is that the ease of use may be unapparent, and the processing may require a heavier engine