Computer Architecture

Lecture – 1

Md Mahadi Hasan Nahid
Assistant Professor,
Department of CSE, SUST
nahid-cse@sust.edu

Outline

- Motivations
- Course Information
- Introduction

Motivations

- How does computing hardware work?
- Where is computing hardware going in the future?
  - And learn how to contribute to this future…
- How does this impact system software and applications?
  - Essential to understand OS/compilers
  - For everyone else, it can help you write better code!
- How are future technologies going to impact computing systems?

Topic of Study

- Focus on what modern computer architects worry about (both academia and industry)
- Get through the basics of modern processor design
- Understand the interfaces between architecture and system software (compilers, OS)
- System architecture and I/O (disks, memory, multiprocessors)
- Look at technology trends, recent research ideas, and the future of computing hardware

Tentative Topics to be covered

- Introduction and Performance Metrics
- ISA/Basic Pipelining Review
- Hardware ILP (Instruction-Level Parallelism)
- Software ILP (Instruction-Level Parallelism)
- Cache/Memory
- Modern Processor Case Studies
- Multiprocessors/Multithreading
- Input/Output and Interconnects
- Research Trends

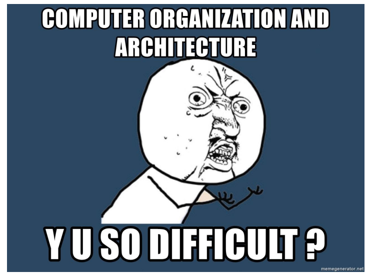
Text Book

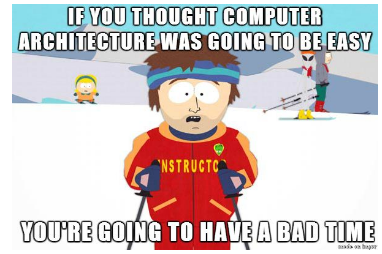
- Computer Architecture – A Quantitative Approach. 6th Edition, by- by- Hennessy & Patterson.
- Computer Organization and Design : The Hardware/Software Interface, 5th Edition, by- Hennessy & Patterson.

Assessment (tentative)

- Class Attendance: 10%
- Home Assignment/Class Test: 20%
- Midterm (Term Test) : 30%
- Final exam: 40%

Warning …





Computer Technology

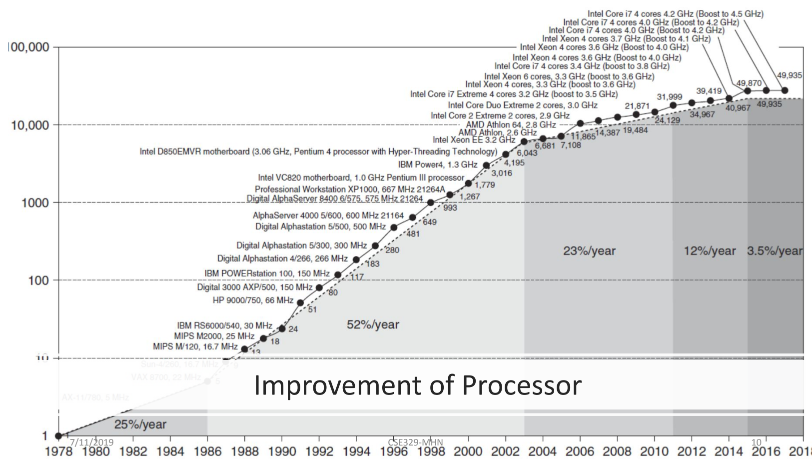
Performance improvements:

- Improvements in semiconductor technology
  - Feature size, clock speed
- Improvements in computer architectures
  - Enabled by HLL compilers, UNIX
  - Lead to RISC architectures

Together have enabled:

- Lightweight computers
- Productivity-based managed/interpreted programming languages

Improvement of Processor



Classes of Computers

Personal Mobile Device (PMD)

- e.g. start phones, tablet computers
- Emphasis on energy efficiency
  - Limited power source
- Real-time performance
  - The running application can have an absolute maximum execution time
  - Consider video, the processor must process two frames within the shortest possible time
- Price must be low,
  - A smart phone in about 4K/5K BDT!

Desktop Computing

- Emphasis on price-performance
- Performance indicate compute as well graphic performance

Classes of Computers

- Internet of Things/Embedded Computers
- Personal Mobile Device
- Desktop Computing
- Servers
- Clusters/Warehouse-Scale Computers

What is Computer Architecture?

Architecture

- Design a building/bridge/… that is well suited for its purpose

Computer Architecture

- Design a computer/computing device that is well suited for its purpose.

Computer Architecture

Application

Long Gap to bridge them in one step

Physics

Computer Architecture

- Application
- Algorithm
- Programming Language
- Operating System
- ISA
- Microarchitecture
- Register-Transfer Level
- Gates
- Circuits
- Devices
- Physics

Computer Architecture

- Instruction Set Architecture (ISA)
- Microarchitecture
- Register-Transfer Level

Computer Architecture is Constantly Changing

Application Requirements:

- Suggest how to improve architecture
- Provide revenue to fund development

Technology Constraints:

- Restrict what can be done efficiently
- New technologies make new arch possible

Architecture provides feedback to guide application and technology research directions

Eight Great Ideas in Computer Architecture

Design for Moore's Law	Use Abstraction to Simplify Design	Make the Common Case Fast	Performance via Parallelism
Performance via Pipelining	Performance via Prediction	Hierarchy of Memories	Dependability via Redundancy

Classes of Parallelism and Parallel Architectures

Data-level parallelism (DLP)

- arises because there are many data items that can be operated on at the same time.

Task-level parallelism (TLP)

- arises because tasks of work are created that can operate independently and largely in parallel.

Parallelism in four major ways

- Instruction-level parallelism
- Vector architectures, graphic processor units (GPUs), and multimedia instruction sets
- Thread-level parallelism
- Request-level parallelism

Architecture vs. Microarchitecture

"Architecture"/Instruction Set Architecture:

- Programmer visible state (Memory & Register)
- Operations (Instructions and how they work)
- Execution Semantics (interrupts)
- Input/Output
- Data Types/Sizes

Microarchitecture/Organization:

- Tradeoffs on how to implement ISA for some metric (Speed, Energy, Cost)
- Examples: Pipeline depth, number of pipelines, cache size, silicon area, peak power, execution ordering, bus widths, ALU widths

Why we need Computer Architecture?

Improve Performance

- Speed
- Battery Life
- Size
- Weight
- Energy Efficiency

Improve Abilities

- 3D Graphics Support
- Debugging Support
- Security

Application Areas

General-Purpose Laptop/Desktop

- Productivity, interactive graphics, video, audio
- Optimize price-performance
- Examples: Intel Pentium 4, AMD Athlon XP

Embedded Computers

- PDAs, cell-phones, sensors => Price, Energy efficiency
- Examples: Intel XScale, StrongARM (SA-110)
- Game Machines, Network uPs => Price-Performance
- Examples: Sony Emotion Engine, IBM 750FX

Application Areas

Commercial Servers

- Database, transaction processing, search engines
- Performance, Availability, Scalability
- Server downtime could cost a brokerage company more
- than $6M/hour
- Examples: Sun Fire 15K, IBM p690, Google Cluster

Scientific Applications

- Protein Folding, Weather Modeling, CompBio, Defense
- Floating-point arithmetic, Huge Memories
- Examples: IBM DeepBlue, BlueGene, Cray T3E, etc.

Computer Architecture and Technology Design Trends

Design with current Technology and Parts
□ Obsolete Computer

Anticipate Future Technology
□ Future Computer

Technology Trends … is very important…

Software Trends

- NO LONGER JUST EXECUTING C/FORTRAN CODE
- OBJECT ORIENTED PROGRAMMING
- JAVA
- ARCHITECTURAL FEATURES TO ASSIST SECURITY
- MIDDLEWARE

Moore's Law

Every 18-24 months

- Doubles memory capacity
- Energy/operation □ 1/2x
- Number of transistors per die increases by □ 2x
- Doubles processor speed

But we are starting to hit some roadblocks…

Also, what to do with all these transistors???

How have we used these transistors?

More functionality on one chip

- Early 1980s – 32-bit microprocessors
- Late 1980s – On Chip Level 1 Caches
- Early/Mid 1990s – 64-bit microprocessors, superscalar (ILP)
- Late 1990s – On Chip Level 2 Caches
- Early 2000s – Chip Multiprocessors, On Chip Level 3 Caches

What is next?

- How much more cache can we put on a chip? (IRAM)
- How many more cores can we put on a chip? (Piranha, etc)
- What else can we put on chips?

Performance vs. Technology Scaling

- Architectural Innovations
  - Massive pipelining (good and bad!)
  - Huge caches
  - Branch Prediction, Register Renaming, OOO-issue/execution,
- Speculation (hardware/software versions of all of these)
- Circuit Innovations
  - New logic circuit families (dynamic logic)
  - Better CAD tools
  - Advanced computer arithmetic

As an architect, our main job is to deal with tradeoffs

- Performance, Power, Die Size, Complexity, Applications

Support, Functionality, Compatibility, Reliability, etc.

- Technology trends, applications… How do we deal

with all of this to make real tradeoffs?

- Abstractions allow this to happen
- Focus is on metrics of these abstractions
- Performance, Cost, Availability, Power

Dealing with Complexity: Abstractions

Metrics: Availability

Availability: Fraction of Time a system is available

For servers, may be as important as performance

Mean Time Between Failures (MTBF)

- Period that a system is usable
- Typically 500,000 hours for a PC Harddrive

Mean Time to Repair (MTRR)

- Recovery time from a failure
- Should approach 0 for a big server (redundancy

Metrics: Power Dissipation

- We all understand why energy is important for
- embedded CPUs…
- High-end Server: ~130-170W
- High-end Desktop: ~70-100W
- High-end Laptop: ~30W
- Battery-Optimized Laptop: ~3-10W
- Embedded CPUs: ~.5-1W
- DSP: ~100mW
- Microcontrollers: ~10mW

Metrics: Performance

Metrics: Cost

Summary : Computer Architecture is very IMPORTANT !!!

Merci Beaucoup