CSE471—Computer Design and Organization

Prof. Steven Burns *Autumn 1994*

Lecture Notes: First Day

Why Learn About Computer Architecture?

- Successful software development requires knowledge of computer organization
  - Explosive growth of computing power and sales
  - Successful software must be fast (as well as correct, etc.)
  - Obtaining maximum speed requires knowledge of computer organization
- As CE's, you might also want to take part in designing the latest machine. For a blow by blow, read *The Soul of a New Machine*, Tracy Kidder, 1981.

Abstraction

- Our ability to design and use computers is based on a number of crucial abstractions:
  - high-level language (e.g. C) — assembly language (machine code)
  - instruction set architecture (ISA) — microarchitecture
  - function blocks and logic gates — networks of transistors
  - digital signals — analog signals
- Chapter 1 gives a good introduction.

Classic Components of a Computer

- Datapath — performs arithmetic and logic operations
- Control — directs operation of other components
- Memory — stores instructions and data; may be hierarchical, including caches and disks (virtual memory)

- I/O — communication and long-term storage
- **Compiler** — schedules instructions for high performance
- **Operating System** — partitions resources to different tasks

Improving Technology

- VLSI chip capacity is doubling every 1.5 years
- This really is amazing.
- The challenge is to make use of this increased capacity.
  - Develop software to solve larger, harder problems
  - Develop hardware that uses more sophisticated organization, e.g., pipelines and caches

What determines the cost of chips?

- Three simple equations:

$$\text{cost per die} = \frac{\text{cost per wafer}}{\text{dies per wafer} \times \text{yield}}$$

$$\text{dies per wafer} = \frac{\text{wafer area}}{\text{die area}}$$

$$\text{yield} = \frac{1}{(1 + \text{defects per area} \times \text{die area}/2)^2}$$

- According to the formula, how can you improve yield?
  - Decrease area.
  - Decrease defects per area.

History of Computers

- For something that started in the 1940's, computer architecture has a deep and interesting history.
- Read the Historical Perspective sections at the end of each chapter.
- There is a lot of insight to be gained.
- Things become more and less important as time marches on.
- Certain things will undoubtedly come back into importance.

Pitfalls and Fallacies

- *Fallacy: Computers have been built in the same, old-fashion way for far too long, and this antiquated model of computation is running out of steam.*
- *Pitfall: Ignoring the inexorable progress of hardware when planning a new machine.*
- *Pitfall: Trying to predict price or performance more than five years into the future in this rapidly moving field.*

What to Get Out of this Class

- How to compare performance of processors and systems.
- The different ways to implement arithmetic and be able to compare the performance vs. cost.
- The basic structure of a RISC processor.
- Standard pipelining techniques. Superscalar Ideas.
- Memory system design. Including caches and virtual memory.
- I/O system design.

What to Get Out of this Class (cont.)

- Experience with Verilog-XL.
- Enough of an understanding of the insides of a processor to appreciate new product descriptions in the popular press (e.g., Byte Magazine.)