CpE 313: Microprocessor Systems Design

Handout 01
Fundamentals of Computer Design

August 24, 2003
Shoukat Ali

shoukat@umr.edu

UNIVERSITY OF MISSOURI-ROLLA
The Name. The Degree. The Difference.

Overview

- administrative matters
- short history of computers
- a definition of computer architecture
- some useful terms about integrated circuits
- technology trends

CpE 313: What Is It About?

- Computer from a programmer's view
  - What the programmer writes
  - How it is converted to something the computer understands
  - How the computer interprets the program
  - How can programs be run faster? What is benchmarking?
- Learn big ideas in computer engineering
  - 5 classic components of a computer
  - Principles/pitfalls of performance measurement
  - Principle of locality, exploited via a memory hierarchy (cache)
  - Greater performance by exploiting parallelism (pipeline)
  - Multiprocessors

What CpE 313 Is Not

- Learning C
- Assembly language programming
  - This is a skill you will pick up, as a side effect of understanding the big Ideas
  - CpE 213 teaches that
- Hardware design
  - Only a little bit of physical logic to give things perspective
  - CpE 111 and 112 teach that

Typical Lecture Format

- 5 minute review of the previous lecture
  - presented by a student
  - will be graded and will count towards the final score
- 35 minute lecture
- 5 minute break (water, stretch)
- 30 minute lecture
- I will come to class early & stay after to answer questions
- in some lectures, slides will have to be filled in by students in class

Will Work in Groups (Leader in Blue)

Acinelli,Todd Michael
Bubenik,David
Chen,Zheng
Wright,Eric M
Smith,Zachary J

Clement,Clifford Chad
Schaefer,Keith J
Freiberger,Thomas Vincent
Gann,Matthew Edward
Kalyani,Radha Padma

Hendren,Matthew Reed
Keeven,Patrick Wayne
Kubilay,Ibrahim Atakan
Langford,Quinton M
Lavallee,Adam Dennis

Lorenz,Joshua Phillip
Lorenz,Timothy Donald
Mansker,Evan R
Middendorf,David James
Muir,Jeffrey D

Ellebrecht,Michael Scott
Schulte,Scott Alan
Seem,Vidhi
Sietins,Alexis Bolster
Soman,Arun

Bilbrey Jr,Randall C
Stallard,William Christopher
Underwood,Ryan Charles
Vernon,Alexander Richard
Wang,Keyou

Course Lecture Outline

- 25 lectures (not including exams and review sessions)
  - 1 week on performance measurement and quantitative principles
  - 1.5 week on design of instruction set architecture
  - 1.5 week on simple pipelining
  - 1 week on simple caches
  - 1 week on precise exceptions and branch prediction
  - 2.5 weeks on dynamic scheduling and superscalar processors
  - 1 week on a case study
  - 2 weeks on advanced issues in caches and memory design
  - 1 week on multiprocessors

Course Exams

- Two midterms and a final
- Will test knowledge versus speed writing
- Can bring 2 letter size pages as crib sheets to all exams

Homework Assignments

- Two types of assignments
  - Exercises from the text book
  - Software assignments (will use SimpleScalar toolset)
  - Collaboration encouraged
- May give a brief (15 minute) quiz on assignment material in lecture
  - Must understand assignment to do quiz
  - 50% credit on assignments 24 hours or less late
    - 0% credit after 24 hours

What You Should Already Know From 111 and 213

- Basic machine structure
  - processor, memory, I/O
- Read and write basic C programs
- Read and write in an assembly language
  - will get familiar with a little bit of MIPS assembly
- Logic design
  - logical equations, schematic diagrams, FSMs, decoders, muxes, etc.

A Short History of Computers

- 1960s – era of large mainframes, machines costing millions of dollars, stored in computer rooms, and run by multiple operators
  - typical applications: business data processing and large-scale scientific computing
- 1970s – era of minicomputers, smaller-sized machines mostly used by multiple users in a "time share" fashion
  - each user would access the machine through a dumb terminal
- 1980s – era of desktop computers based on microprocessors
  - personal computers and workstations
  - individually owned, no time-sharing
  - led to servers, computers that provided larger-scale services such as reliable, longterm file storage (file servers) and more computing power (compute servers)
- 1990s – era of the Internet and the World Wide Web

Three Types of Computers Currently Important

- desktop, server, and embedded computers
- embedded computers: computers lodged in other devices where the presence of the computers is not immediately obvious

Feature	Desktop	Server	Embedded
Price of system	\$1000–\$10,000	\$10,000–\$10,000,000	\$10–\$100,000 (including network routers at the high end)
Price of microprocessor module	\$100–\$1000	\$200–\$2000 (per processor)	\$0.20–\$200 (per processor)
Microprocessors sold per year (estimates for 2000)	150,000,000	4,000,000	300,000,000 (32-bit and 64-bit processors only)
Critical system design issues	Price-performance, graphics performance	Throughput, availability, scalability	Price, power consumption, application-specific performance

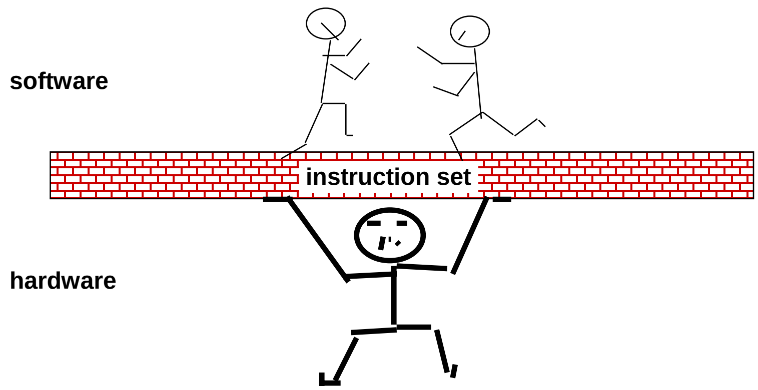
What is "Computer Architecture"

- Computer Architecture =
  - Instruction Set Architecture
    - the attributes of a [computing] system as seen by the programmer, i.e. the conceptual structure and functional behavior

PLUS

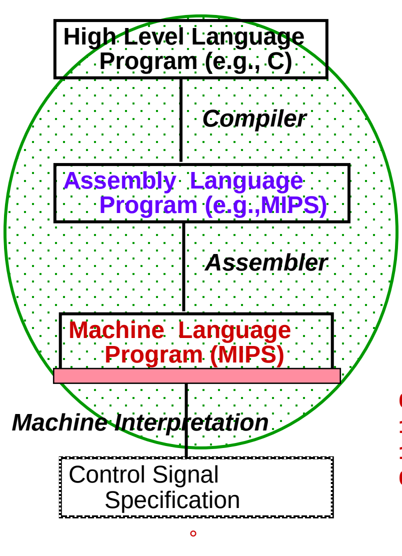
  - Machine Organization
    - functional units and connections, pipeline configuration, location & configuration of caches

Instruction Set Architecture



Some examples of ISA's are:
Digital Alpha, HP PA-RISC, Sun Sparc, SGI MIPS, Intel

Which Instructions are We Talking About?




```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
lw  $to,   0($2)
lw  $t1,   4($2)
sw  $t1,   0($2)
sw  $t0,   4($2)
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

Why Do ISA's Differ?

- different companies have different customers, and are trying to satisfy them

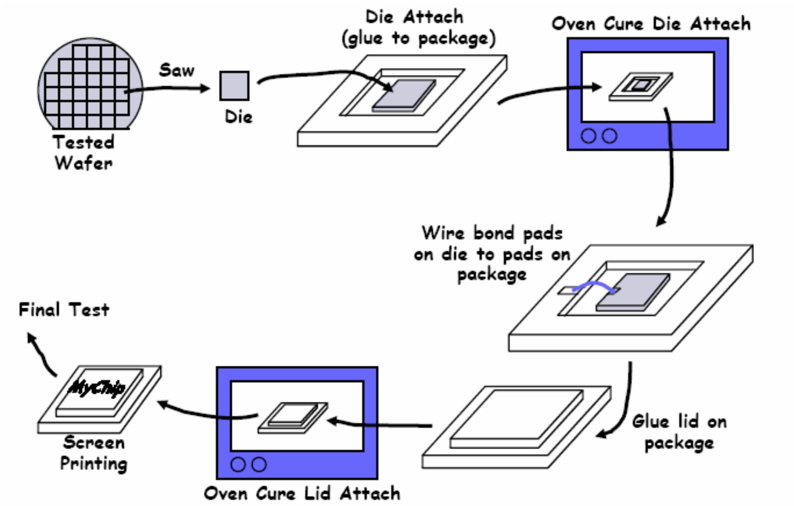
ISA 1	ISA 2

temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
	temp = v[k];
	v[k] = v[k+1];
	v[k+1] = temp;

lw $reg0,mem1
lw $reg1,mem2
sw $reg1,mem1
sw $reg0,mem2
	xchg mem1, mem2

Transition: Some Useful Terms About ICs

Some Useful Terms About ICs – Wafer, Die, Chip

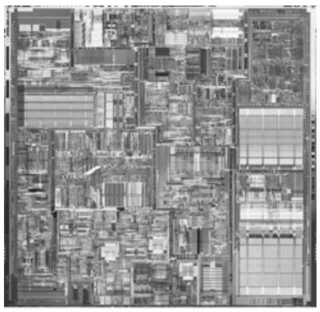


Some Useful Terms About ICs – Feature Size

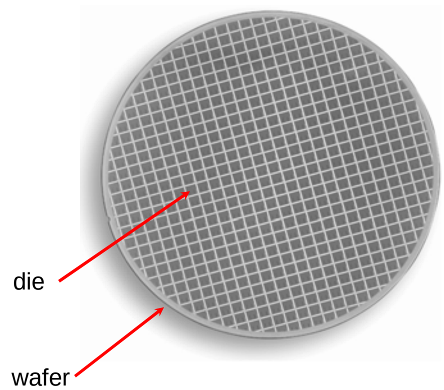
- the size of a transistor
  or
  the width of the wires connecting transistors on the chip
- state of the art chips are using sub-micron feature sizes from 0.25 micron (1997) to 0.13 micron (2003)
- the smaller the feature size, the more transistors there are available on a given chip area
  - Pentium 3, 018 micron, 28 million transistors
  - Pentium 4, 0.13 micron, 55 million transistors
- also means that a smaller-feature-size microprocessor
  - will be smaller,
  - run faster, and
  - use less power
- because more of these smaller chips can be obtained from a single wafer, each chip will cost less

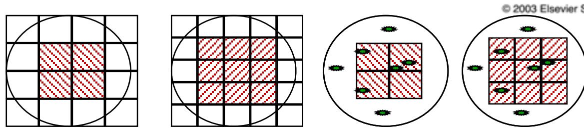
Some Useful Terms About ICs – Die Size

- why not increase the number of transistors on a chip by increasing the die size, i.e., the area of silicon used for each chip?



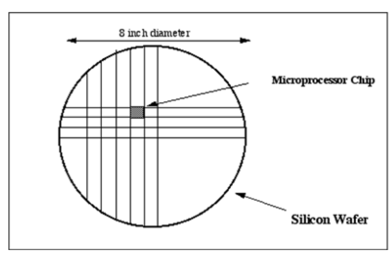
Pentium 4 Processor





Some Useful Terms About ICs – Die Size

- why not increase the number of transistors on a chip by increasing the die size, i.e., the area of silicon used for each chip?
- reason: it will decrease the yield of “good” chips
- silicon is converted into circular “wafers,” and then chips are cut off that wafer
- a fixed number of faults occur randomly on a wafer of given **area**
- a single fault will render an individual chip useless



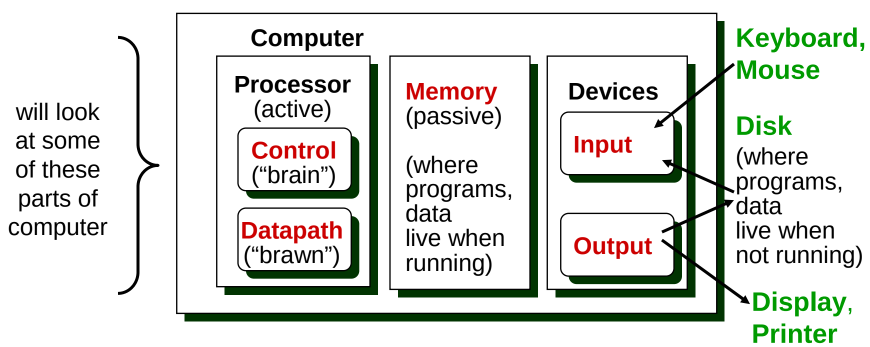
- example: if a wafer were to contain 40 chips and ten faults occur randomly, then up to 10 of the 40 chips may be useless giving up to 25% wastage
- with 200 (smaller) chips, there would still be 10 faulty chips, i.e., 5% wastage
- *there is a trade-off between die size and yield, i.e. a larger die size leads to a decrease in yield*

Transition: Technology Trends

Tell Me How to Design a New Computer!

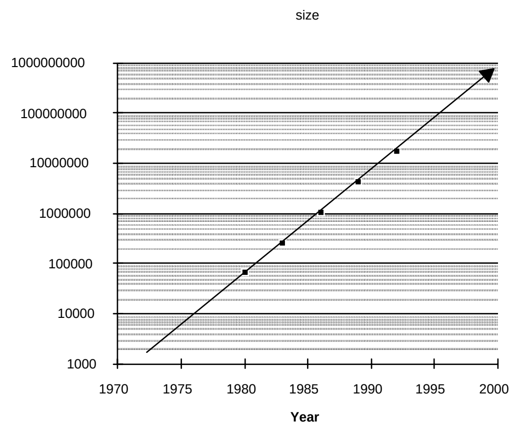
- ok, I am interested, now tell me how to design a new computer

- if a new design has to be successful, it must survive changes in computer technology
- an architect must plan for technology changes
- let us look at past technology trends to plan the future



Technology Trends: Semiconductor DRAM

- density (memory cells/area) increases 40%--60% every year
- access time reduced by 1/3 in last ten years (slow!)
- bandwidth/chip increased by about 2/3 in last ten years

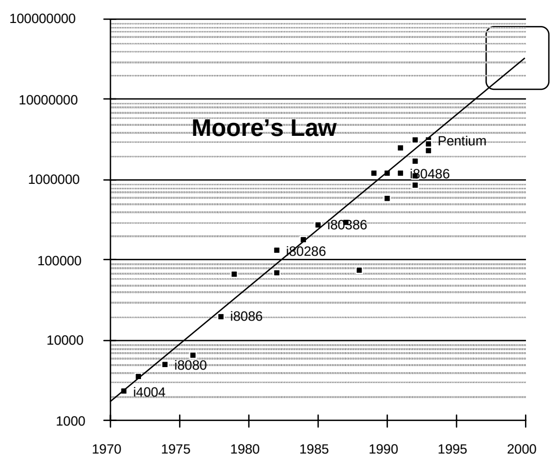


year	size(Megabit)
1980	0.0625
1983	0.25
1986	1
1989	4
1992	16
1996	64
1999	256

if density increases 1.6 X/yr, it doubles every 1.5 years:

Technology Trends: IC Logic Technology

- density (trans./area) increases by 35% every year
- die size increases by 10%--20% every year (why not by 200%?)
- combined effect: trans. count/chip increases by 55%



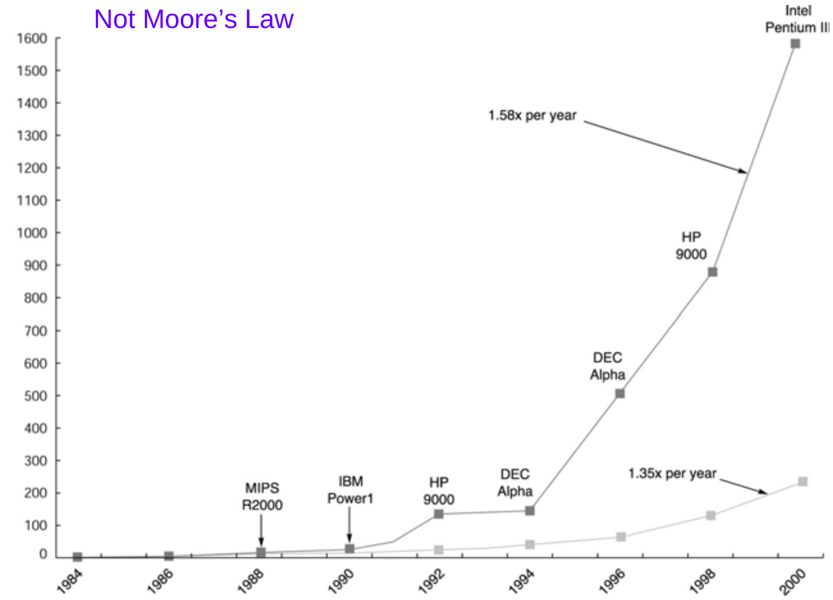
Alpha 21264: 15 million
Pentium Pro: 5.5 million
PowerPC 620: 6.9 million
Alpha 21164: 9.3 million
Sparc Ultra: 5.2 million

apprx 2X transistors/chip Every 1.5 years

Called "Moore's Law":

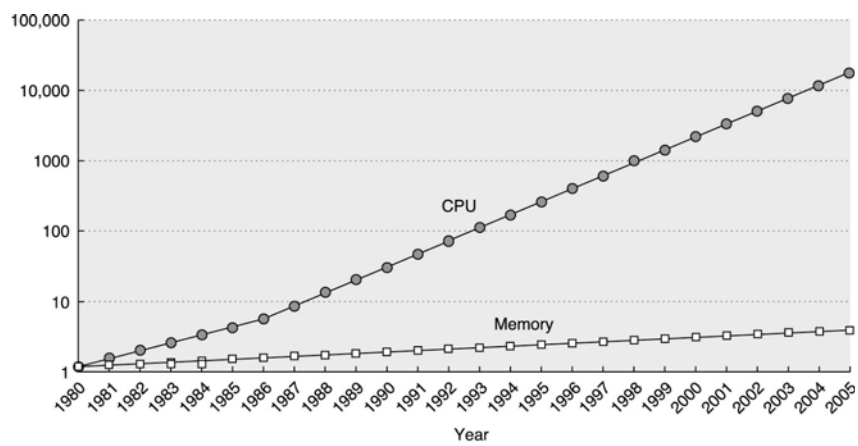
Technology Trends: Processor Speed

Not Moore's Law



Technology Trends: Memory Speeds

- main memory speeds have not kept up with the processor speeds



Technology Trends: Magnetic Disk

- disk density improving by 100% every year since 1990
- access time improved by 1/3 in past ten years (slow!)

Summary: Dramatic Changes in Computer Technology

- Processor
  - 2X in speed every 1.5 years
  - 2X in transistor count every 1.5 years
- Memory
  - DRAM capacity: 2X / 1.5 years
  - access time reduced by 1/3 is last ten years
  - bandwidth/chip increased by about 2/3 in last ten years
- Disk
  - density: 100% every year since 1990
  - access time improved by 1/3 in past ten years

Application Trends

- Numerical, scientific → commercial, entertainment
- Few, large → ubiquitous, small
  - mainframes → minis → microprocessors → handheld→embedded
- Increasing memory requirements – for both main memory and disk storage
- Standalone →networked

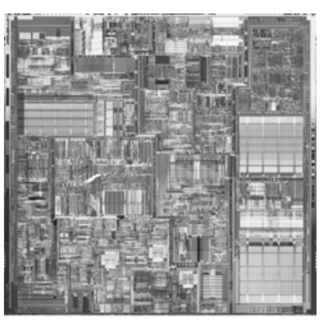
Example Applications

- Scientific/Numerical
  - Computational Fluid Dynamics, Weather Prediction, ECAD
    - Long word length, floating point arithmetic
- Commercial
  - inventory control, billing, payroll, decision support
    - byte oriented, fixed point, high I/O, large secondary storage
- Home Computing
  - multimedia, entertainment
    - high bandwidth data movement, graphics,
    - encryption, compression/decompression

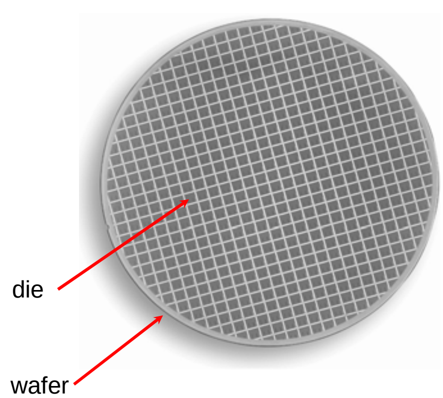
Cost Trends

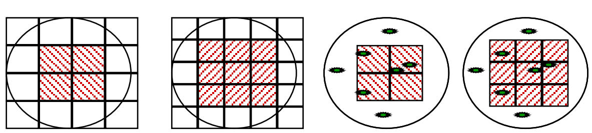
- cost of a computer component decreases over time
- factors
  - learning curve – cost to manufacture decreases over time
  - volume – increasing volume of production decreases the time to learn, cost of purchasing raw materials, etc.
  - commodities – essentially identical products sold by multiple vendors in large volumes, e.g., DRAMs, disks, keyboards
    - competition drives the cost down

Reminder: Useful Terms for Integrated Circuits



Pentium 4 Processor





Integrated Circuit Cost

$$\text{IC Cost} = \frac{\text{Die Cost + Cost of Testing die + Cost of Packaging \& Final Test}}{\text{Final Test Yield}}$$

$$\text{Die Cost} = \frac{\text{Wafer Cost}}{\text{Dies/Wafer * Die Yield}}$$

$$\text{Dies per Wafer} = \frac{\pi\ (\text{Wafer\_diameter}/2)^2}{\text{Die Area}} - \frac{\pi\ (\text{Wafer\_diameter})}{\left(2 * \text{Die Area}\right)^{\frac{1}{2}}}$$

$$\text{Die Yield} = \text{Wafer Yield} * \left(\frac{\text{Defects per unit area} * \text{Die\_Area}}{\alpha}\right)^{-\alpha}$$

Where α is a parameter inversely proportional to the number of mask levels, which is a measure of the manufacturing complexity.
For today's 6-metal CMOS process, good estimate is $\alpha = 4$

Real World Examples

Chip	Metal Layers	Wafer Cost ($)	Defects/ cm^2	Area (mm^2)	Dies/ Wafer	Yield	Die Cost ($)
386DX	2	900	1.0	43	360	71	4
486DX2	3	1200	1.0	81	181	54	12
PowerPC 601	4	1700	1.3	121	115	28	53
HP PA 7100	3	1300	1.0	196	66	27	73
DEC Alpha	3	1500	1.2	234	53	19	149
SuperSPARC	3	1700	1.6	256	48	13	272
Pentium	3	1500	1.5	296	40	9	417

From "Estimating IC Manufacturing Costs," by Linley Gwennap, *Microprocessor Report*, August 2, 1993, p. 15