

# Computer Architecture

# Introduction



Lynn Choi  
Korea University



高麗大學校

*Computer System Laboratory*



# Class Information

## ☐ Lecturer

- ▶ Prof. Lynn Choi, School of Electrical Eng.
- ▶ Phone: 3290-3249, **공학관** 411, [lchoi@korea.ac.kr](mailto:lchoi@korea.ac.kr),
- ▶ TA: **윤창현/신동욱**, 3290-3896, [{yunch, shindw}@korea.ac.kr](mailto:{yunch, shindw}@korea.ac.kr)

## ☐ Time

- ▶ Tue/Thu 3:30pm – 4:45pm
- ▶ Office Hour: Thu 5:00pm – 6:00pm

## ☐ Place

- ▶ **공학관**466(Tue)/**공학관**250(Thu)

## ☐ Textbook

- ▶ “Computer Systems: A Programmer’s Perspective”, Randal E. Bryant and David O’Hallaron, Prentice Hall, 2<sup>nd</sup> Edition, 2011.

## ☐ References

- ▶ Computer Organization and Design: The Hardware/Software Interface, (MIPS version), D. Patterson, J. Hennessy, 4<sup>th</sup> Edition, Morgan Kaufmann, 2009

## ☐ Class homepage

- ▶ <http://it.korea.ac.kr> : slides, announcements



# Class Information

## Project

- ▶ MIPS/ARM assembly programming
- ▶ iPhone/Android programming

## Evaluation

- ▶ Midterm : 35%
- ▶ Final: 35%
- ▶ Homework and Projects: 25%
- ▶ Class participation: extra 5%
  - ◆ Attendance: no shows of more than 2 will get -5%
  - ◆ Bonus points and Quiz



# Computer

© SAMSUNG. All rights reserved



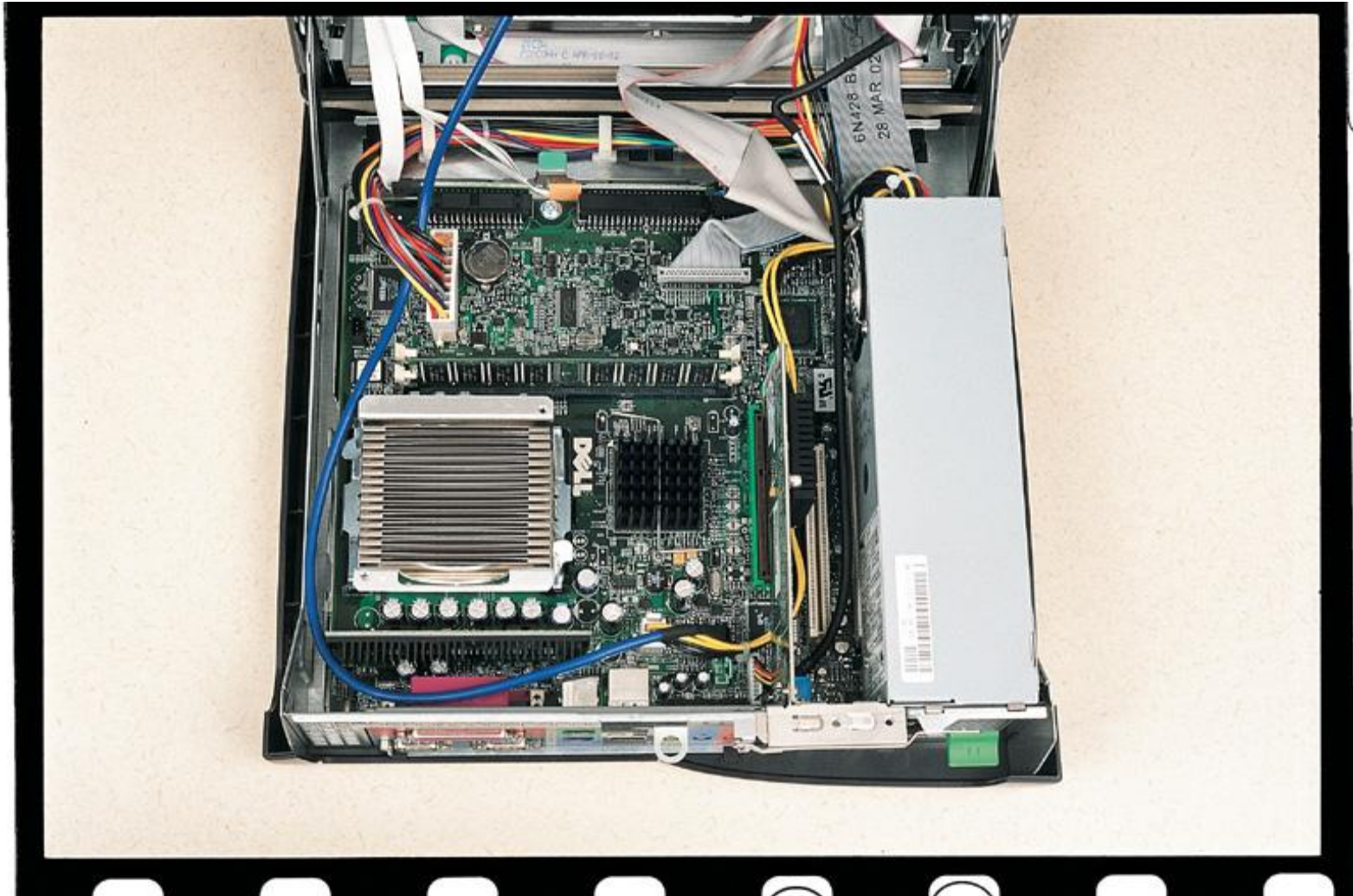
Apple Inc. All rights reserved







# What's Inside a Computer?





# Basics: What is inside Computer?

## **Processor(s)**

- ▶ Also called CPU (Central Processing Unit)
- ▶ Fetches instructions from memory
- ▶ Executes instructions
- ▶ Transfers data from/to memory

## **Memory**

- ▶ Caches, main memory, HDD, CD-ROM, DVD, ROM, FLASH, ..
- ▶ Stores program and data

## **Input devices**

- ▶ Mouse, keyboard, camera, pen, touch screen, barcode reader, scanner, microphone, ...

## **Output devices**

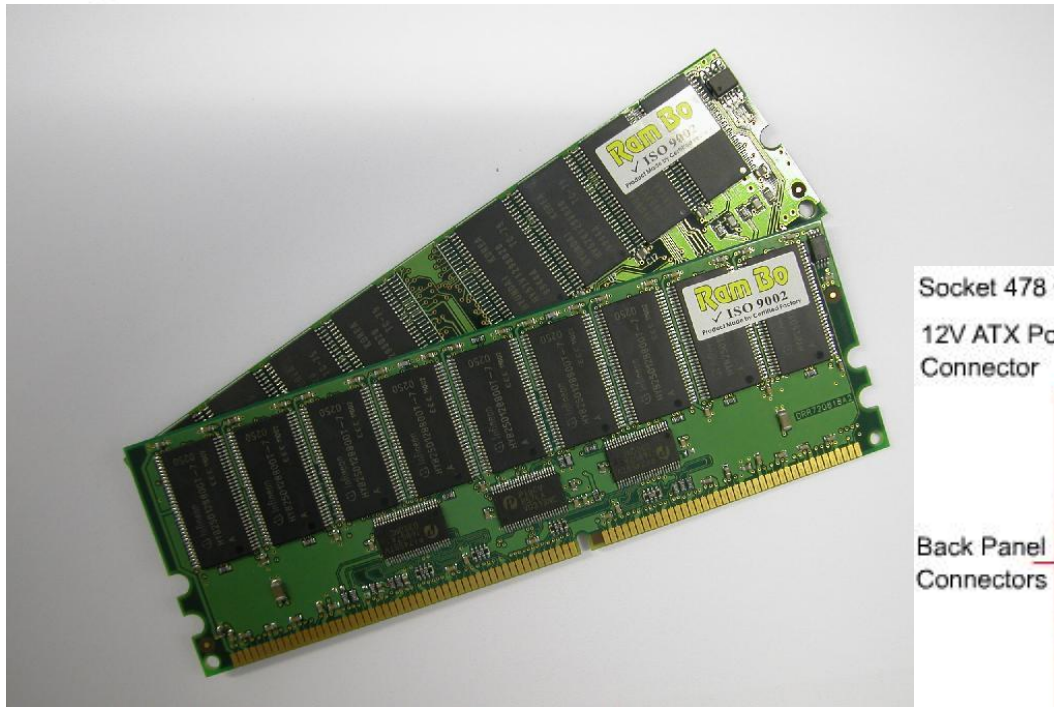
- ▶ Printer, monitor, speaker, beam projector, ..

## **Interconnects: buses**

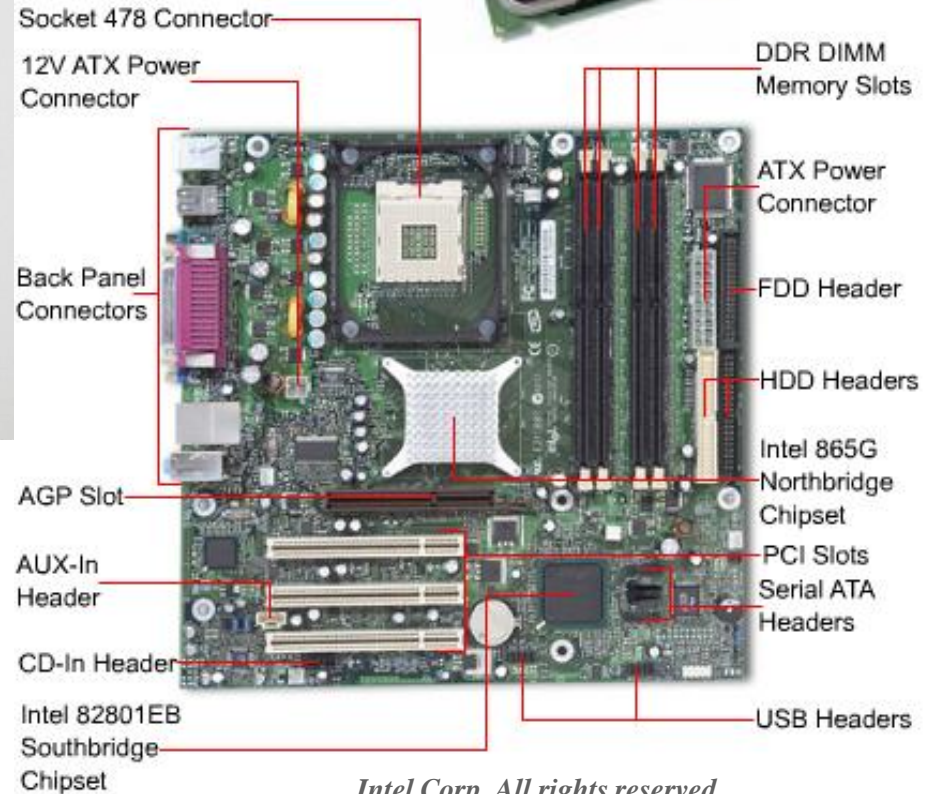
## **Motherboards, chipsets, ...**



# Computer Components



*Intel Corp. All rights reserved*



*Intel Corp. All rights reserved*

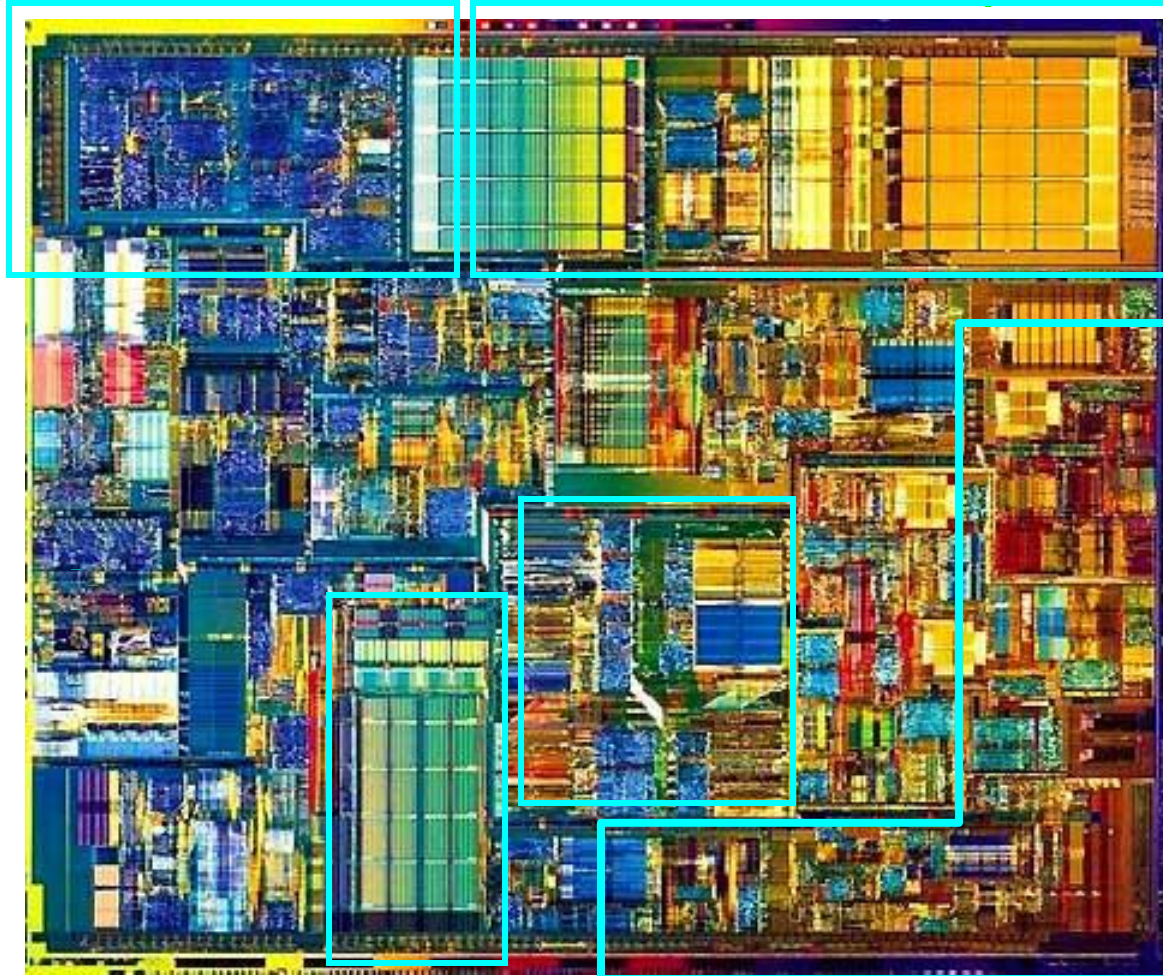




# What's inside a CPU?

## Pentium 4 Processor Die on 0.18 micron (42M transistors)

400MHz  
system bus



Advanced  
Transfer Cache

Pipeline

Trace cache    FP/MMX

*Intel Corp. All rights reserved*





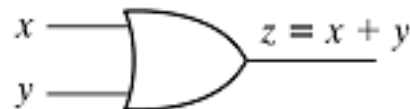
# Logic Level : Gates

**Table 1-8**  
*Truth Tables of Logical Operations*

AND			OR			NOT	
$x$	$y$	$x \cdot y$	$x$	$y$	$x + y$	$x$	$x'$
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		



(a) Two-input AND gate



(b) Two-input OR gate

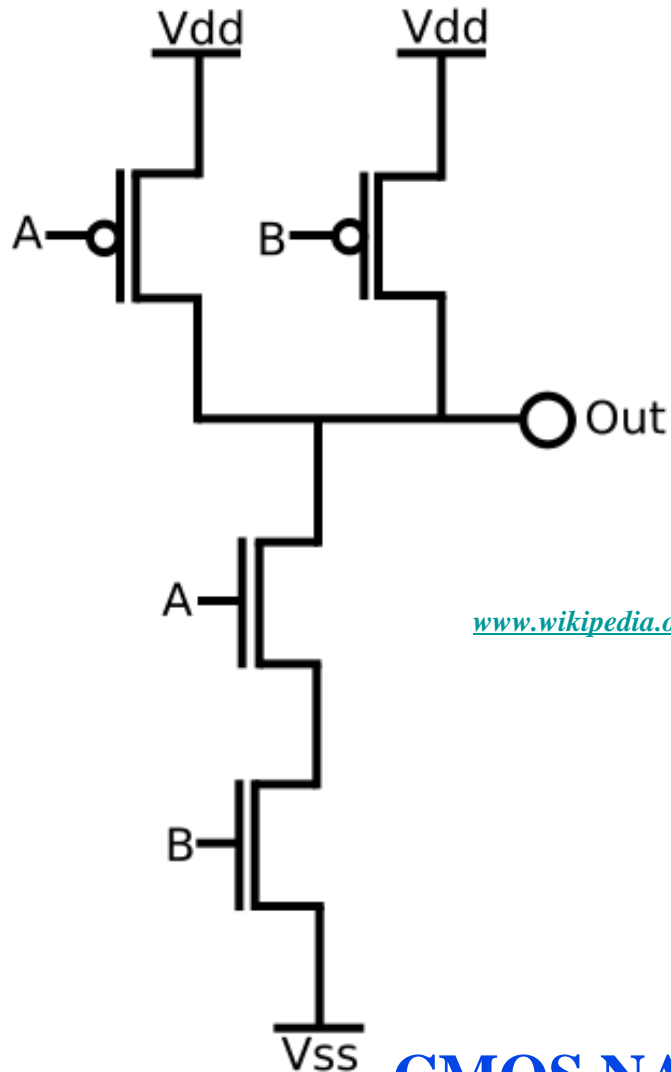


(c) NOT gate or inverter

*Prentice Hall Inc. All rights reserved*

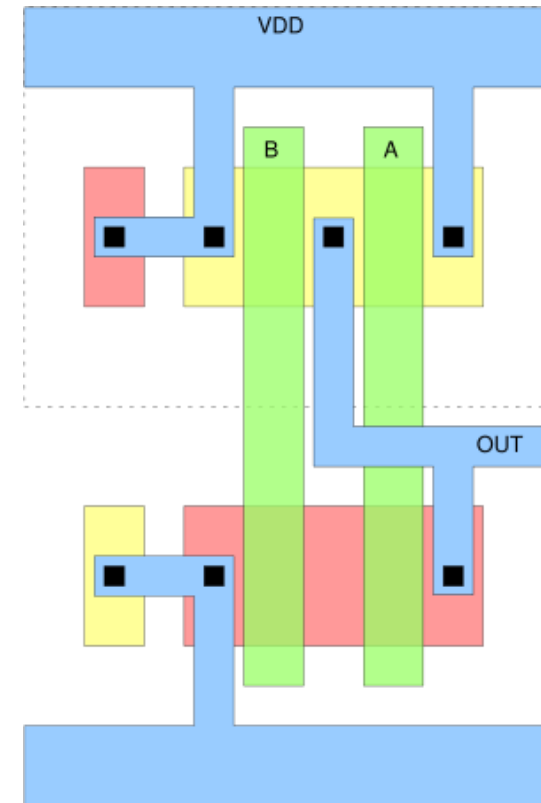


# Circuit Level: Transistors



[www.wikipedia.org](http://www.wikipedia.org) All rights reserved

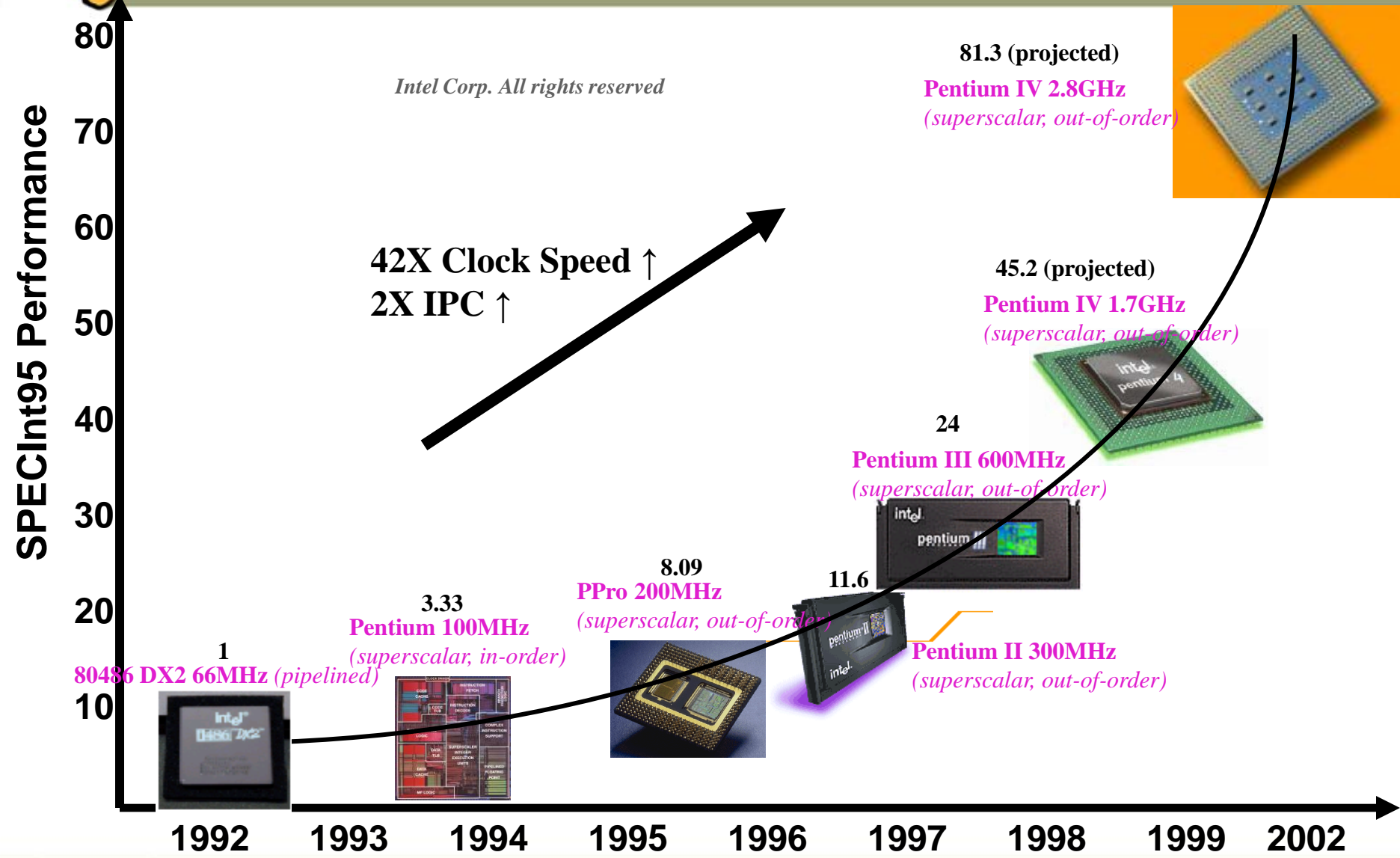
## CMOS NAND Gate



- METAL1
- POLY
- CONTACT
- N DIFFUSION
- P DIFFUSION
- N-WELL



# Advances in Intel Microprocessors







# Terminology

## ❏ **Microprocessor: a single chip processor**

- ◆ Pentium IV, AMD Athlon, SUN Ultrasparc, ARM, MIPS, ..

## ❏ **ISA (Instruction Set Architecture)**

- ▶ Defines machine instructions and visible machine states such as registers and memory
- ▶ Examples
  - ◆ X86(IA32): 386 ~ Pentium III, Pentium IV
  - ◆ IA64: Itanium, Itanium2
  - ◆ Others: PowerPC, SPARC, MIPS, ARM

## ❏ **Microarchitecture**

- ▶ Implementation: implement according to the ISA
  - ◆ Pipelining, caches, branch prediction, buffers
- ▶ Invisible to programmers



# Terminology

## ❏ CISC (Complex Instruction Set Computer)

- ▶ Each instruction is complex
  - ◆ Instructions of different sizes, many instruction formats, allow computations on memory data, ...
- ▶ A large number of instructions in ISA
- ▶ Architectures until mid 80's
  - ◆ Examples: x86, VAX

## ❏ RISC (Reduced Instruction Set Computer)

- ▶ Each instruction is simple
  - ◆ Fixed size instructions, only a few instruction formats
- ▶ A small number of instructions in ISA
- ▶ Load-store architectures
  - ◆ Computations are allowed only on registers
    - ✦ Data must be transferred to registers before computation
- ▶ Most architectures built since 80's
  - ◆ Examples: MIPS, ARM, PowerPC, Alpha, SPARC, IA64, PA-RISC, etc.



# Terminology

## Word

- ▶ Default data size for computation
  - ◆ Size of a GPR & ALU data path depends on the word size
- ▶ The word size determines if a processor is a 8b, 16b, 32b, or 64b processor

## Address (or pointer)

- ▶ Points to a location in memory
- ▶ Each address points to a byte (*byte addressable*)
- ▶ If you have a 32b address, you can address  $2^{32}$  bytes = 4GB
- ▶ If you have a 256MB memory, you need at least 28 bit address since  $2^{28} = 256\text{MB}$

## Caches

- ▶ Faster but smaller memory close to processor
  - ◆ Fast since they are built using SRAMs, but more expensive





# Pentium 4 Microprocessor

## Intel Pentium IV Processor

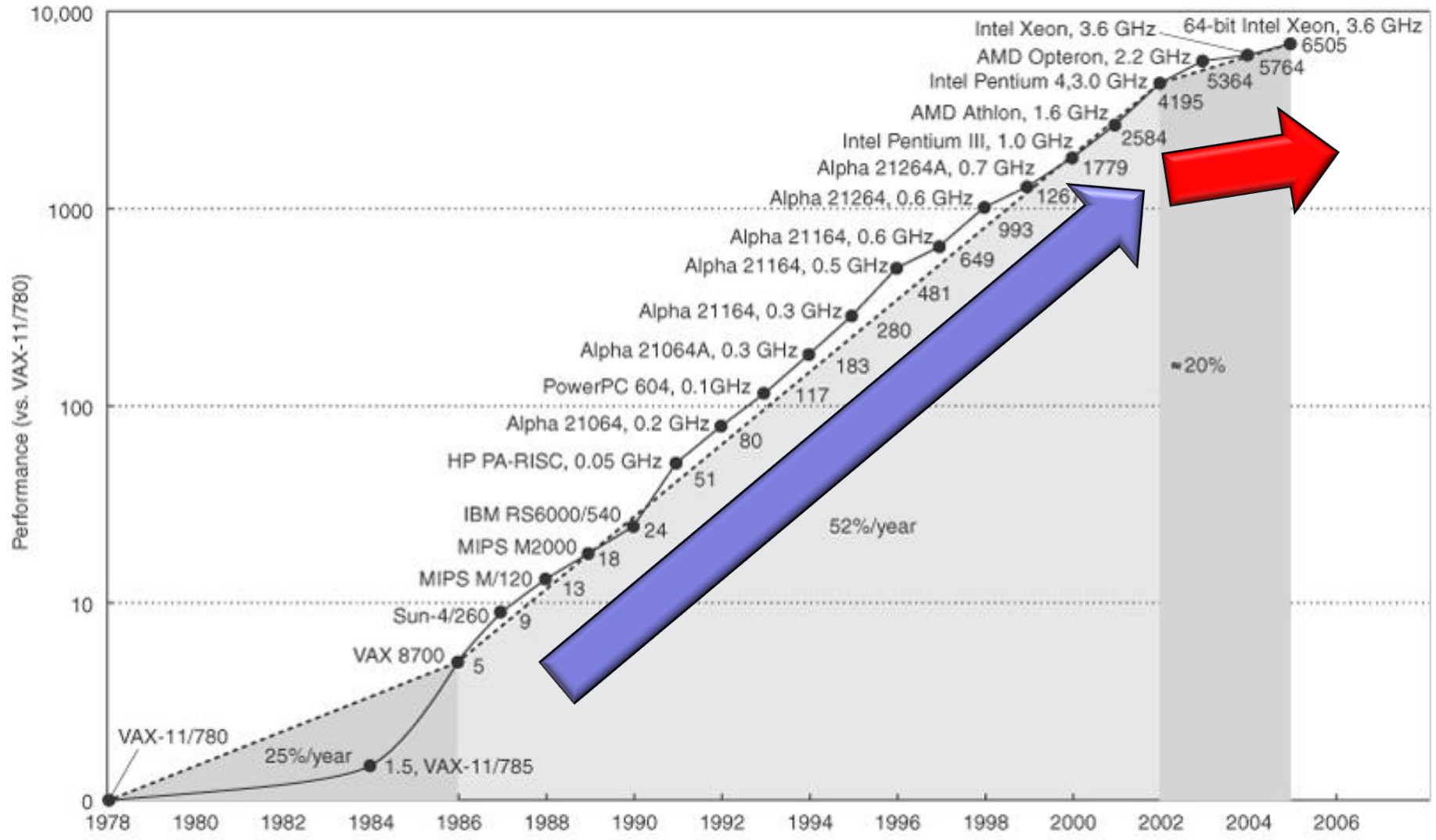
- ▶ Technology
  - ◆ 0.13 $\mu$  process, 55M transistors, 82W
  - ◆ 3.2 GHz, 478pin Flip-Chip PGA2
- ▶ Performance
  - ◆ 1221 Ispec, 1252 Fspec on SPEC 2000
    - ✦ Relative performance to SUN 300MHz Ultrasparc (100)
  - ◆ 40% higher clock rate, 10~20% lower IPC compared to P III
- ▶ Pipeline
  - ◆ 20-stage out-of-order (OOO) pipeline, hyperthreading
- ▶ Cache hierarchy
  - ◆ 12K micro-op trace cache/8 KB on-chip D cache
  - ◆ On-chip 512KB L2 ATC (Advanced Transfer Cache)
  - ◆ Optional on-die 2MB L3 Cache
- ▶ 800MHz system bus, 6.4GB/s bandwidth
  - ◆ Compared with 1.06GB/s on P III 133MHz bus
  - ◆ Implemented by quad-pumping on 200MHz system bus



*Intel Corp. All rights reserved*



# Microprocessor Performance Curve



© 2007 Elsevier, Inc. All rights reserved.



# Today's Microprocessor

## Intel i7 Processor

Intel Corp. All rights reserved

- ▶ Technology
  - ◆ 32nm process, 130W, 239 mm<sup>2</sup> die
  - ◆ 3.46 GHz, 64-bit 6-core 12-thread processor
  - ◆ 159 Ispec, 103 Fspec on SPEC CPU 2006 (296MHz UltraSparc II processor as a referencen machine)
- ▶ Core microarchitecture
  - ◆ Next generation *multi-core* microarchitecture introduced in Q1 2006 (Derived from P6 microarchitecture)
  - ◆ Optimized for multi-cores and lower power consumption
    - 14-stage 4-issue out-of-order (OOO) pipeline
  - ◆ 64bit Intel architecture (x86-64)
  - ◆ Core i3 (entry-level), Core i5 (mainstream consumer), Core i7 (high-end consumer), Xeon (server)
- ▶ 256KB L2 cache/core, 12MB L3 Caches
- ▶ Integrated memory controller
  - ◆ 36bit physical address, 3 channel, 3.2GH clock, 25.6 GB/s memory bandwidth (memory up to 24GB DDR3 SDRAM)







# Today's Microprocessor

## Apple A6 Chip

Apple Inc. All rights reserved

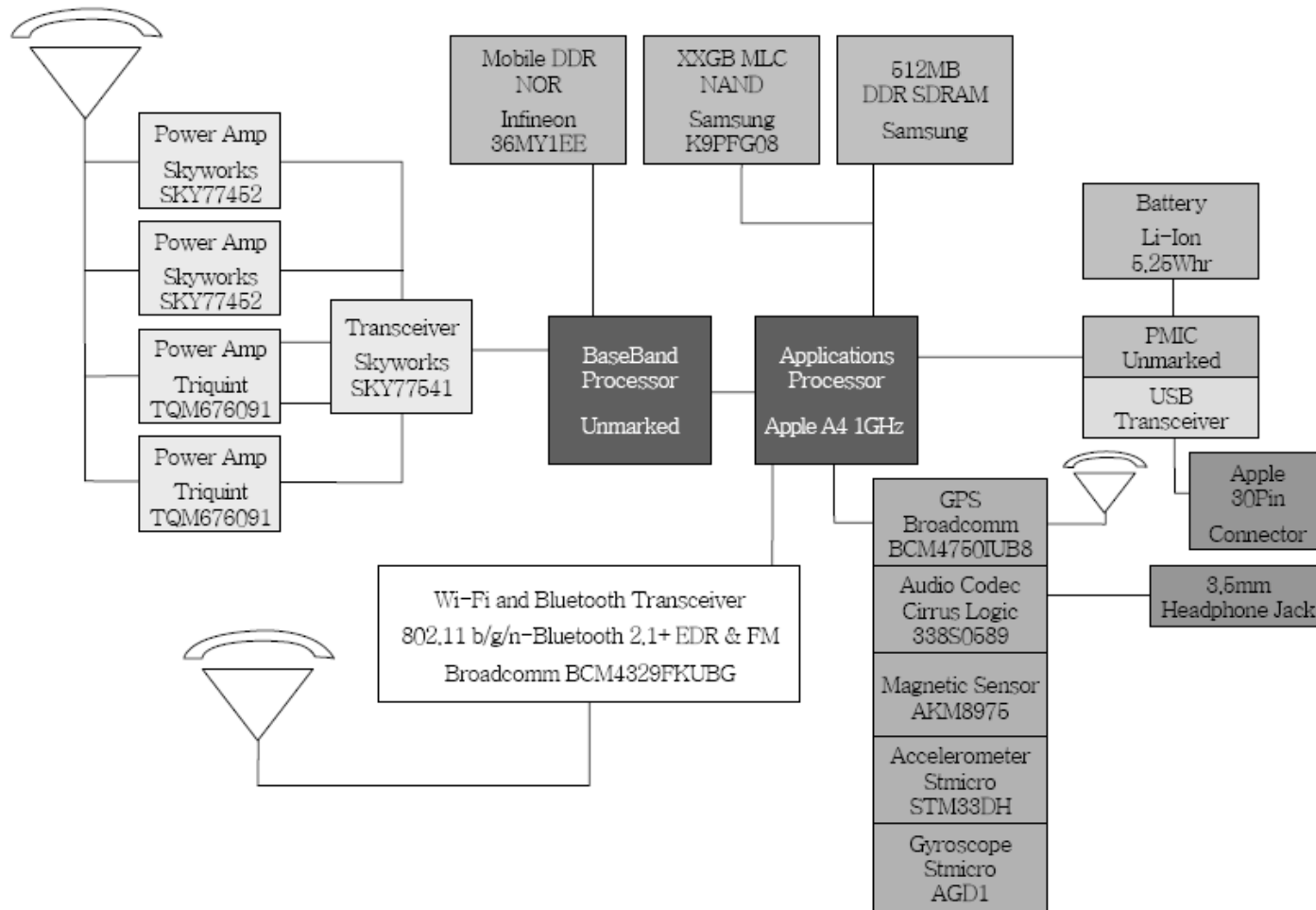
### Technology

- ◆ SoC designed by Apple and manufactured by Samsung
- ◆ 32nm process, 96.71mm<sup>2</sup>, 500m~800mW
- ◆ 1.3 GHz ARMv7 based dual-core CPU with integrated triple-core PowerVR GPU
  - Custom-designed dual-core CPU called *swift*
    - » Rather than ARM-licensed CPU (ARM Cortex A9MPcore in A5 and ARM Cortex A8 in A4)
  - Triple-issue superscalar pipeline
  - 2X speedup in both CPU and GPU performance compared to A5
- ◆ Used for iPhone 5 and 4<sup>th</sup> generation IPAD (contains 1.4GHz A6X)
- ◆ 1GB RAM





# iPhone 4 System Architecture



<자료>: [www.ifixit.com](http://www.ifixit.com)

(그림 3) 스마트폰(아이폰4)의 구조



# Processor Performance Equation

■  $T_{\text{exe}}$  (Execution time per program)

$$= NI * CPI_{\text{execution}} * T_{\text{cycle}}$$

- ▶ NI: # of instructions / program (program size)
  - ◆ Small program is better
- ▶ CPI: clock cycles / instruction
  - ◆ Small CPI is better. In other words, higher IPC is better
- ▶  $T_{\text{cycle}}$  = clock cycle time
  - ◆ Small clock cycle time is better. In other words, higher clock speed is better





# Class Information

## ☐ **Class content**

- ▶ Introduction (Chapter 1)
- ▶ Instruction Set Architecture (Chapter 2)
- ▶ *Linking (Ch. 7)*
- ▶ Computer Arithmetic (Chapter 3)
- ▶ Pipelining (Chapter 4)
- ▶ Caches and Memory Hierarchy (Chapter 5)
- ▶ *Virtual Memory (Ch. 9)*
- ▶ *Exceptions and Signals (Ch. 8)*
- ▶ *System-Level IO (Ch. 10)*
  - ◆ Input and Output (Chapter 6)



# Homework 1

☐ **Read Chapter 1 and Chapter 2**

☐ **Exercise**

- ▶ 1.1
- ▶ 1.4
- ▶ 1.6
- ▶ 1.7
- ▶ 1.10