



# Modern Computer Architecture

## Lecture1 Fundamentals of Quantitative Design and Analysis (I)

**Hongbin Sun**

**国家集成电路人才培养基地**

**Xi'an Jiaotong University**

# Course Administration

- Instructor: Prof. Hongbin Sun ([hsun@mail.xjtu.edu.cn](mailto:hsun@mail.xjtu.edu.cn))
- Office: West 4th Building, Qujiang Campus
- TA:
- Lectures:
- Text Book: Computer Architecture: A Quantitative Approach  
Hennessey and Patterson, 5th Edition (2012)
- Prerequisite: Digital Logic&Computer Organization
- Course Webpage:  
<http://gr.xjtu.edu.cn/web/hsun/3>

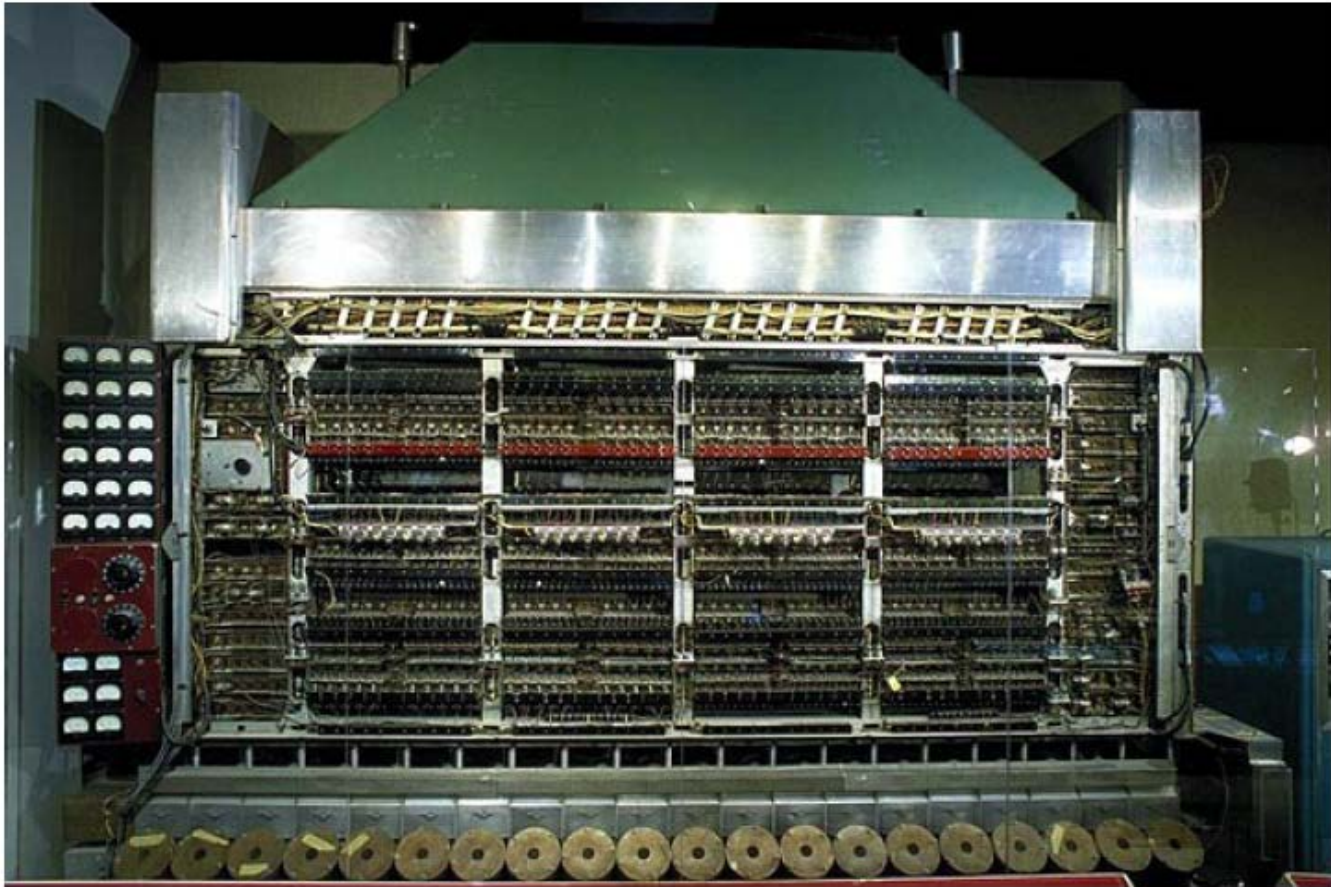


# 1.1 Introduction



- ENIAC was the first electronic general-purpose computer announced in 1946. ENIAC was designed to calculate **artillery firing tables (火炮射击图表)** for the US Army's **Ballistic Research Laboratory (弹道研究实验室, BRL)**.
- Computer technology has made incredible progress in the roughly 65 years since ENIAC was created.

# Computers then



IAS Machine. Design directed by John Von Nuemann.  
First booted in Princeton NJ in 1952



# Computers now



**Sensor Nets**



**Cameras**



**Set-top boxes**



**Games**



**Media Players**



**Laptops**



**Servers**



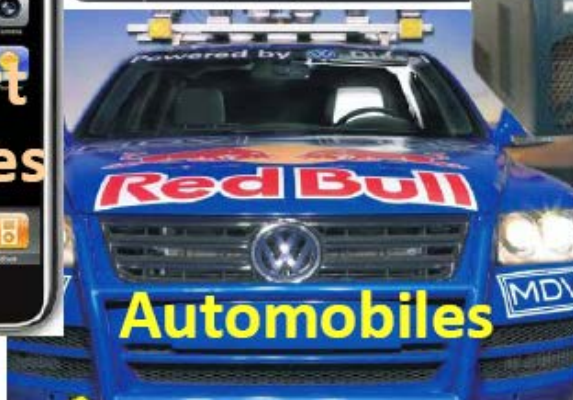
**Routers**



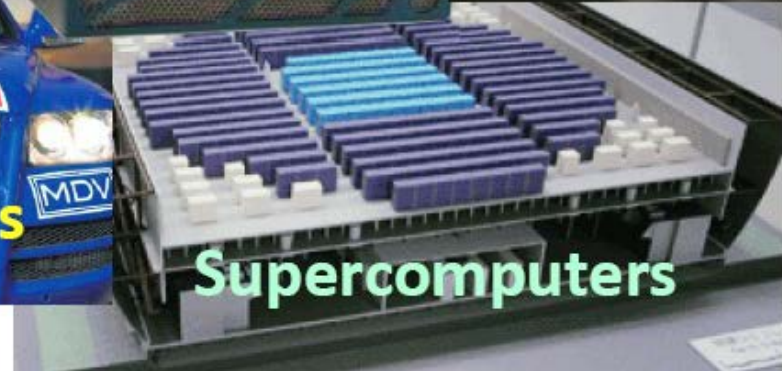
**Robots**



**Smart phones**

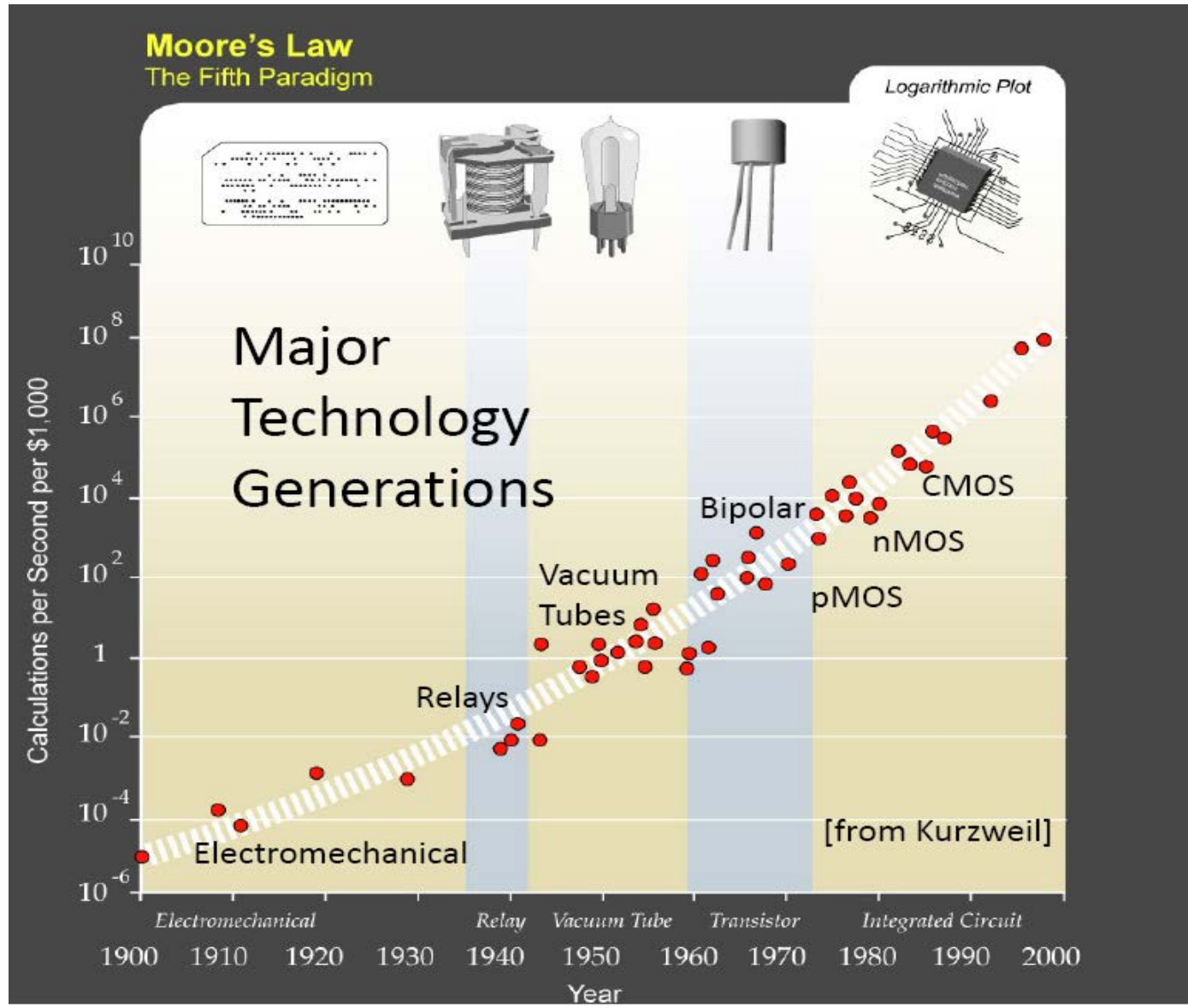


**Automobiles**

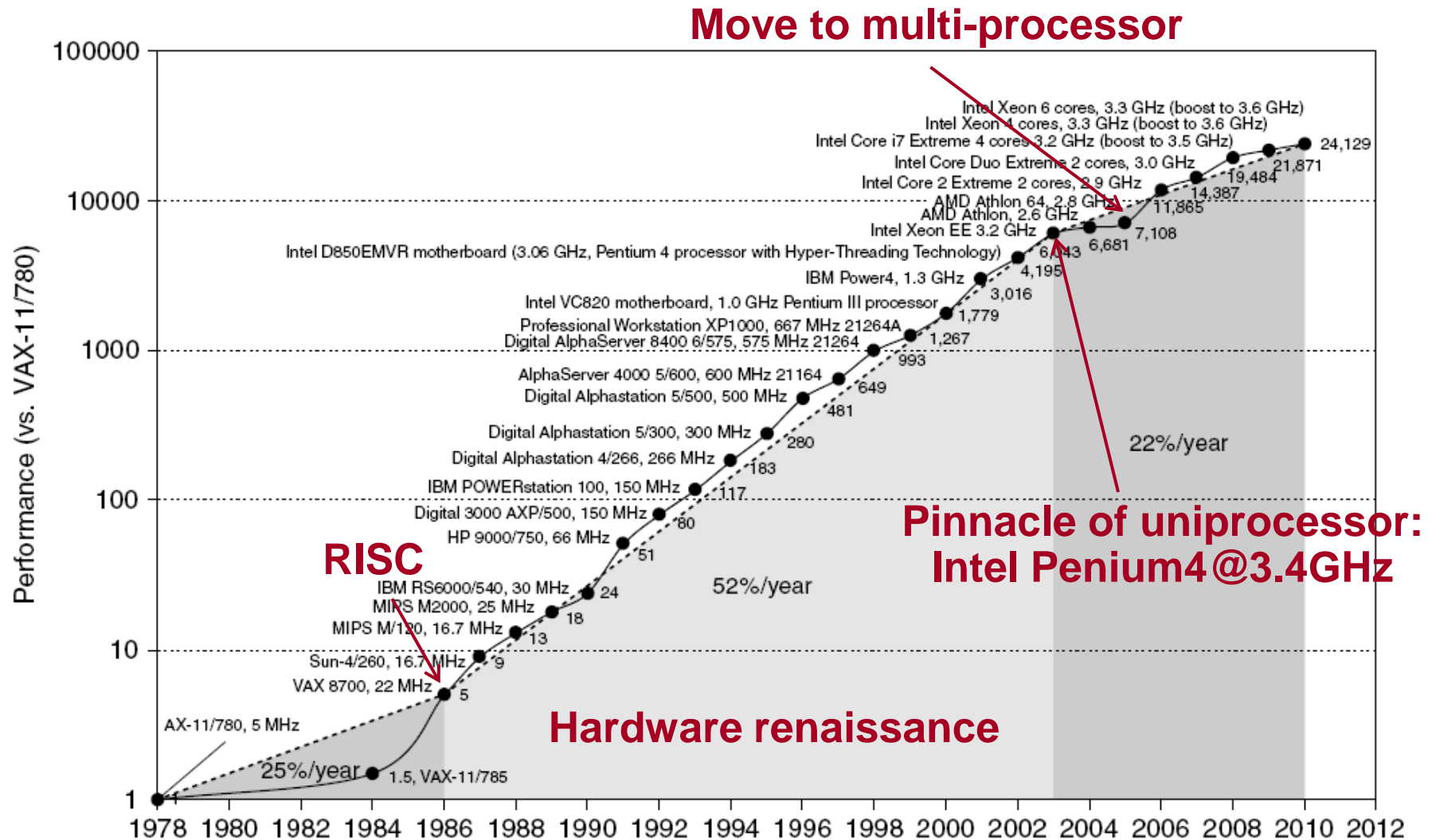


**Supercomputers**

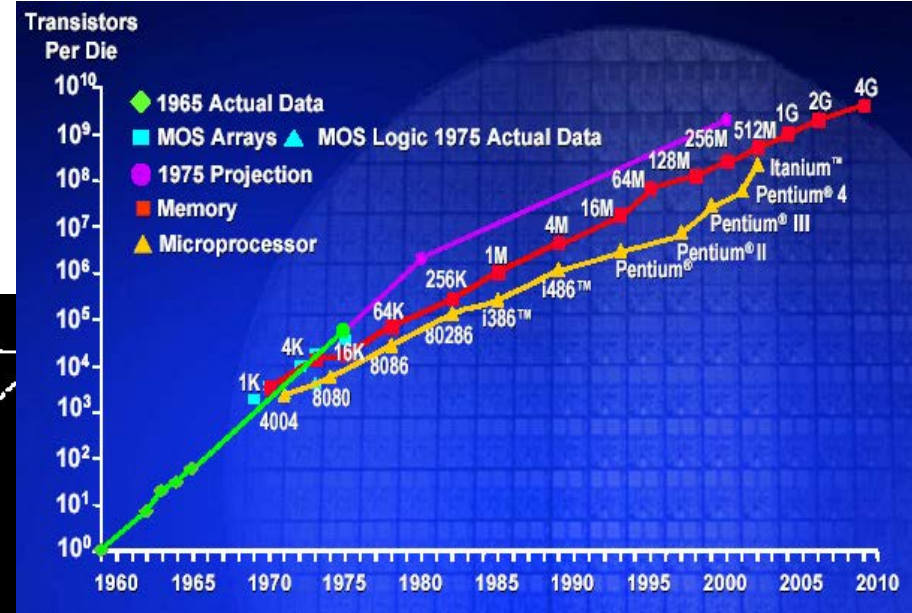
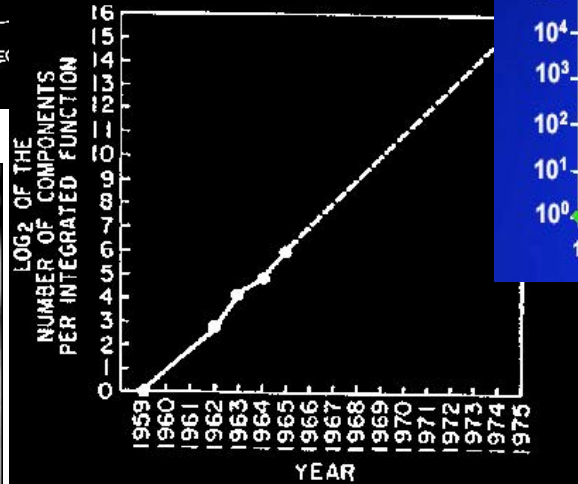
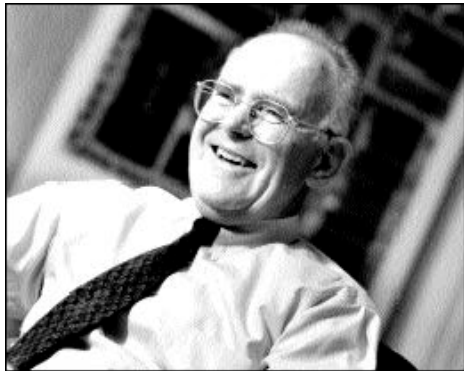
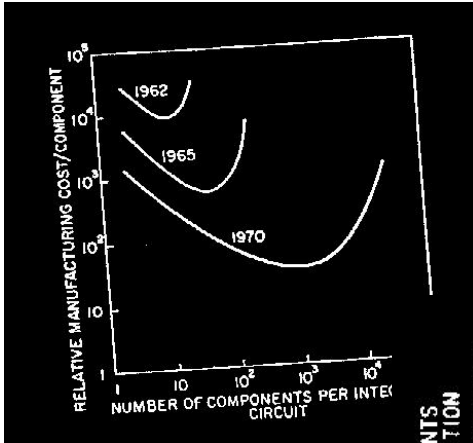
# Major Technology Generations



# Growth in Processor Performance



# The Moore's Law



- “Cramming More Components onto Integrated Circuits”
  - Gordon Moore, Electronics, 1965
- # of transistors on cost-effective integrated circuit double every 18 months



# CISC vs. RISC

- **CISC (Complex Instruction Set Computer), 1970s**
- **Two significant changes in the computer marketplace**
  - The emergence of high level language
  - The creation of standardized, vendor-independent operating systems, such as UNIX and Linux
- **RISC (Reduced Instruction Set Computer), 1980s**
  - The exploitation of *instruction-level parallelism* (pipelining and multiple instruction issue)
  - The use of cache
- **The RISC-based computers raised the performance bar, forcing prior architectures to keep up or disappear**
  - Digital Equipment Vax
  - Intel x86

# RISC Architecture

- **RISC**: A fixed (32-bit) instruction size with few format;
- **CISC**: typically had variable length instruction sets with many format.
- **RISC**: A load-store arch. where data processing instructions operate only on registers, separate from MA instruction;
- **CISC**: typically allowed values in memory to be used as operands in data processing instructions.
- **RISC**: A large register bank of thirty two 32-bit registers, all of which could be used for any purpose, to allow the load-store architecture to operate efficiently;
- **CISC**: not as large as RISC, and most had different registers for different purpose.

# RISC Organization

- **RISC**: hard-wired instruction decode logic;
- **CISC**: used large microcode ROMs to decode their instructions.
- **RISC**: pipelined execution;
- **CISC**: allowed little, if any, overlap between consecutive instruction (though they do now)
- **RISC**: single-cycle execution;
- **CISC**: typically took many clock cycles to complete a single instruction.
  
- **RISC**: MIPS, ARM
- **CISC**: x86

# The Growth Effect in 20th Century

- 1. It has significantly enhanced the capability available to computer users.**
- 2. This dramatic improvement in cost-performance leads to new classes of computers.**
  - Personal Computer (PC)**
  - Mobile Client Devices**
  - Warehouse-scale computer**
- 3. Continuing improvement of semiconductor manufacturing has led to the dominance of microprocessor-based computers across the entire range of computer design.**
- 4. Software development, allowed programmers today to trade performance for productivity.**



# SaaS & Cloud Computing



- **Software as a Service (SaaS) used over the Internet is replacing shrink-wrapped software that must be installed and run on a local computer.**

# The Diversified Applications



**Google's  
Goggles**

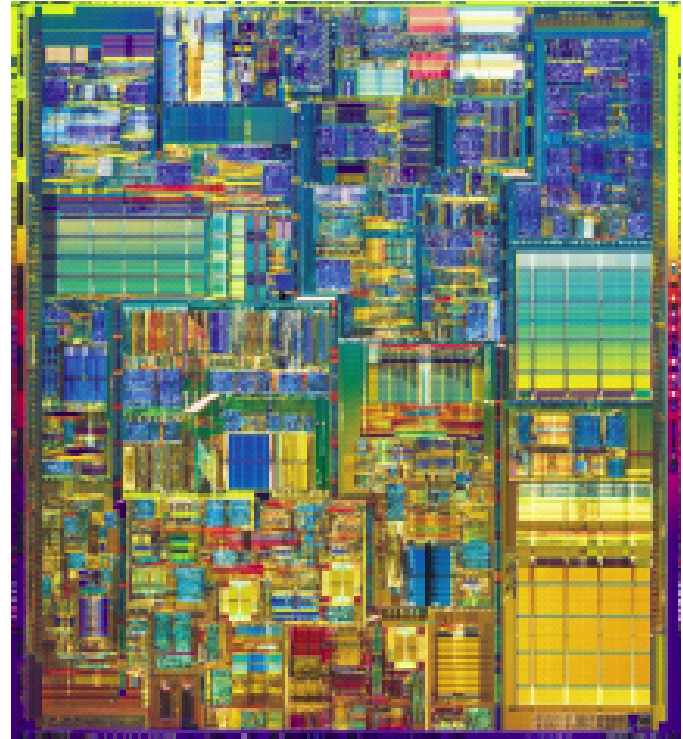
**The nature of applications also changed!**

# The Growth after 2003

- **Since 2003, uniprocessor performance improvement has dropped to less than 22% per year.**
  - Maximum power dissipation of air-cooled chip
  - The lack of more instruction-level parallelism to exploit efficiently
- **The milestone signal in 2004: Intel canceled its uniprocessor project.**
  - Instruction-Level Parallelism (ILP)
  - Data-Level Parallelism (DLP)
  - Thread-Level Parallelism (TLP)
  - Request-Level Parallelism (RLP)

# Pinnacle of Single-Core MP

- Intel Pentium4 (2003)
  - Application: desktop/server
  - Technology: 90nm (1/100x)
  - 55M transistors (20,000x)
  - 101 mm<sup>2</sup> (10x)
  - 3.4 GHz (10,000x)
  - 1.2 Volts (1/10x)
  - 32/64-bit data (16x)
  - 22-stage pipelined datapath
  - 3 instructions per cycle (superscalar)
  - Two levels of on-chip cache
  - data-parallel vector (SIMD) instructions, hyperthreading





# The Future: Processor becomes a transistor?



Intel 48 cores single chip cloud computing

# 1.2 Classes of Computers

- **Personal Mobile Device (PMD)**
- **Desktop Computing**
- **Servers**
- **Clusters/Warehouse-Scale Computers**
- **Embedded Computers**

# Classes of Parallelism and Arch.

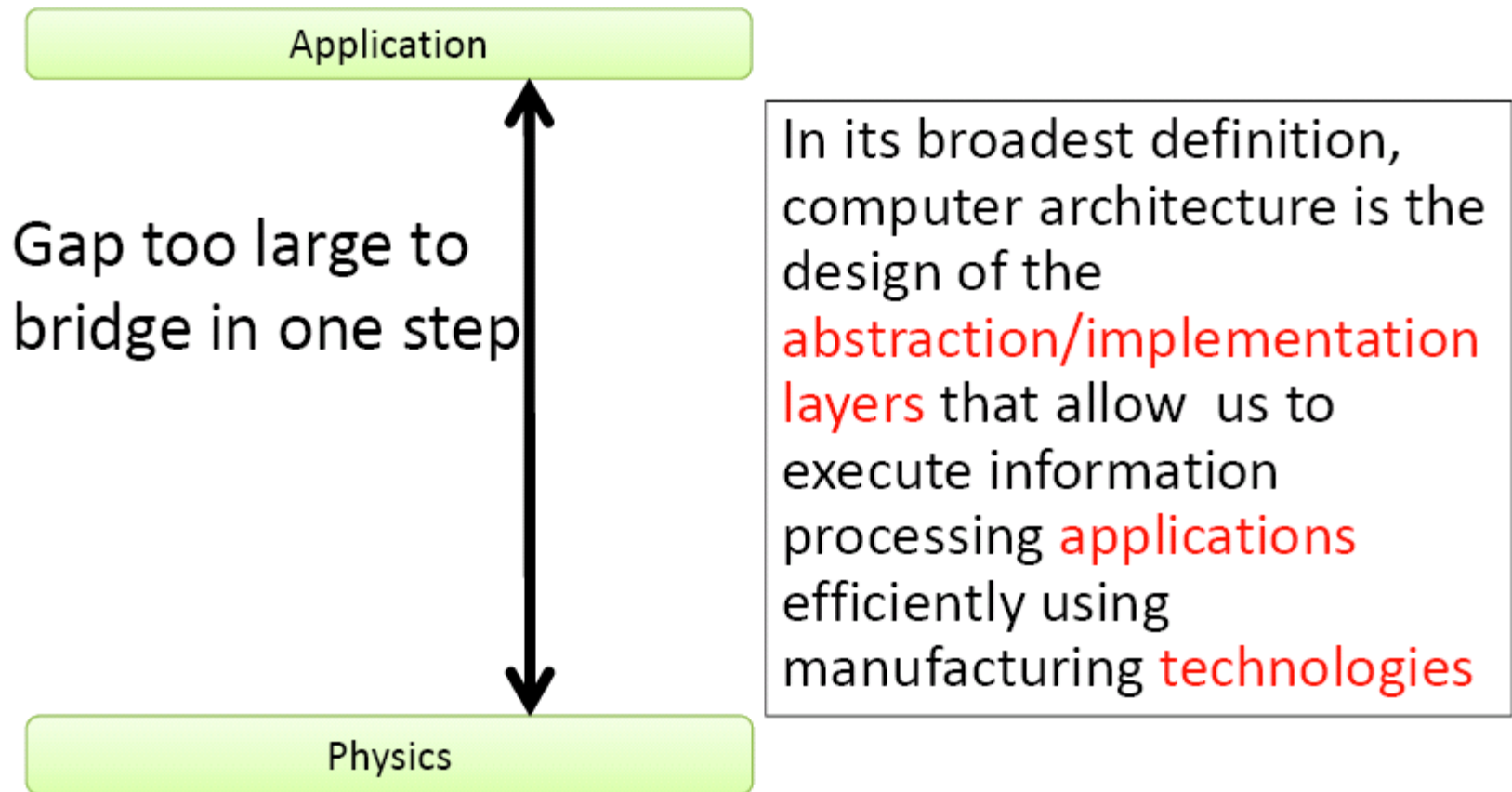
- There are basically two kinds of parallelism in applications:
  - Data-Level Parallelism (DLP)
  - Task-Level Parallelism (TLP)
- Computer hardware in turn can exploit these two kinds of application parallelism in four major ways:
  - Instruction-Level Parallelism: pipelining & speculative execution
  - Vector Architecture and Graphic Processor Units (GPU)
  - Thread-Level Parallelism
  - Request-Level Parallelism

# Flynn's Taxonomy of CA

- **Single instruction stream, single data stream (SISD)**
  - Uniprocessor
  - Instruction Level Parallelism, ILP
- **Single instruction stream, multiple data stream (SIMD)**
  - Vector architecture, multimedia extensions and GPUs
  - Data-Level Parallelism, DLP
- **Multiple instruction streams, single data stream (MISD)**
  - No commercial multiprocessor of this type right now
- **Multiple instruction streams, multiple data stream (MIMD)**
  - Multiprocessor, Thread-Level Parallelism, TLP
  - Cluster and ware-house scale computers, RLP



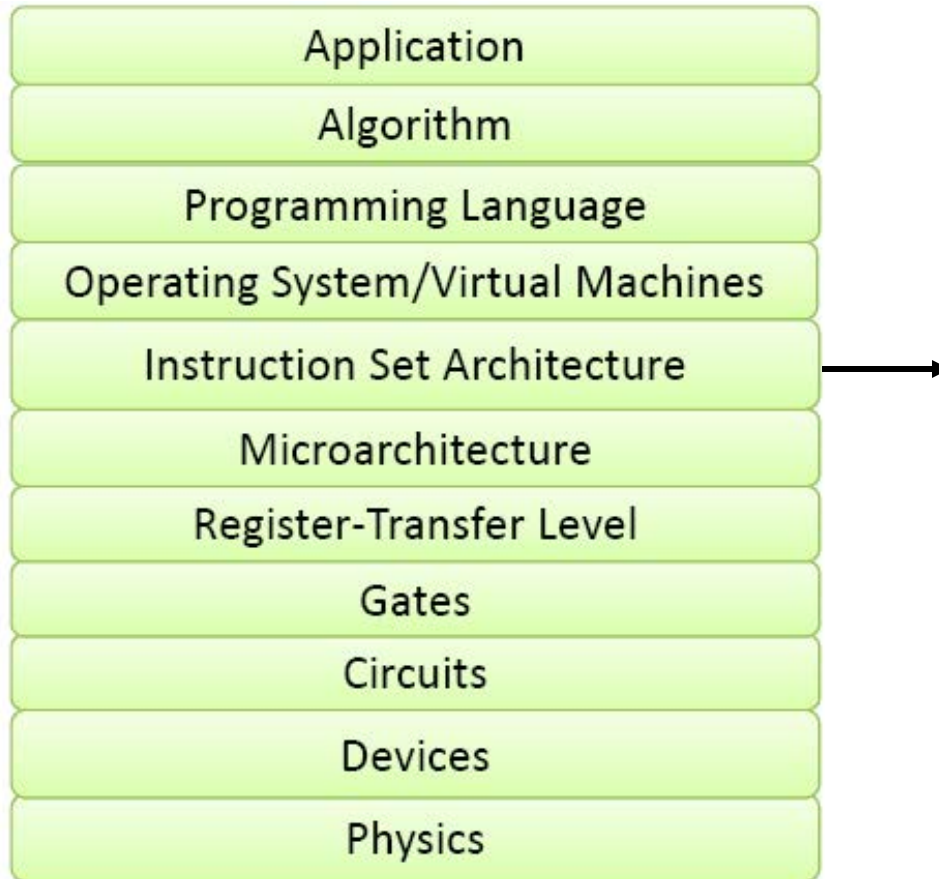
# 1.3 What is Computer Architecture?



# Abstraction in modern computer system

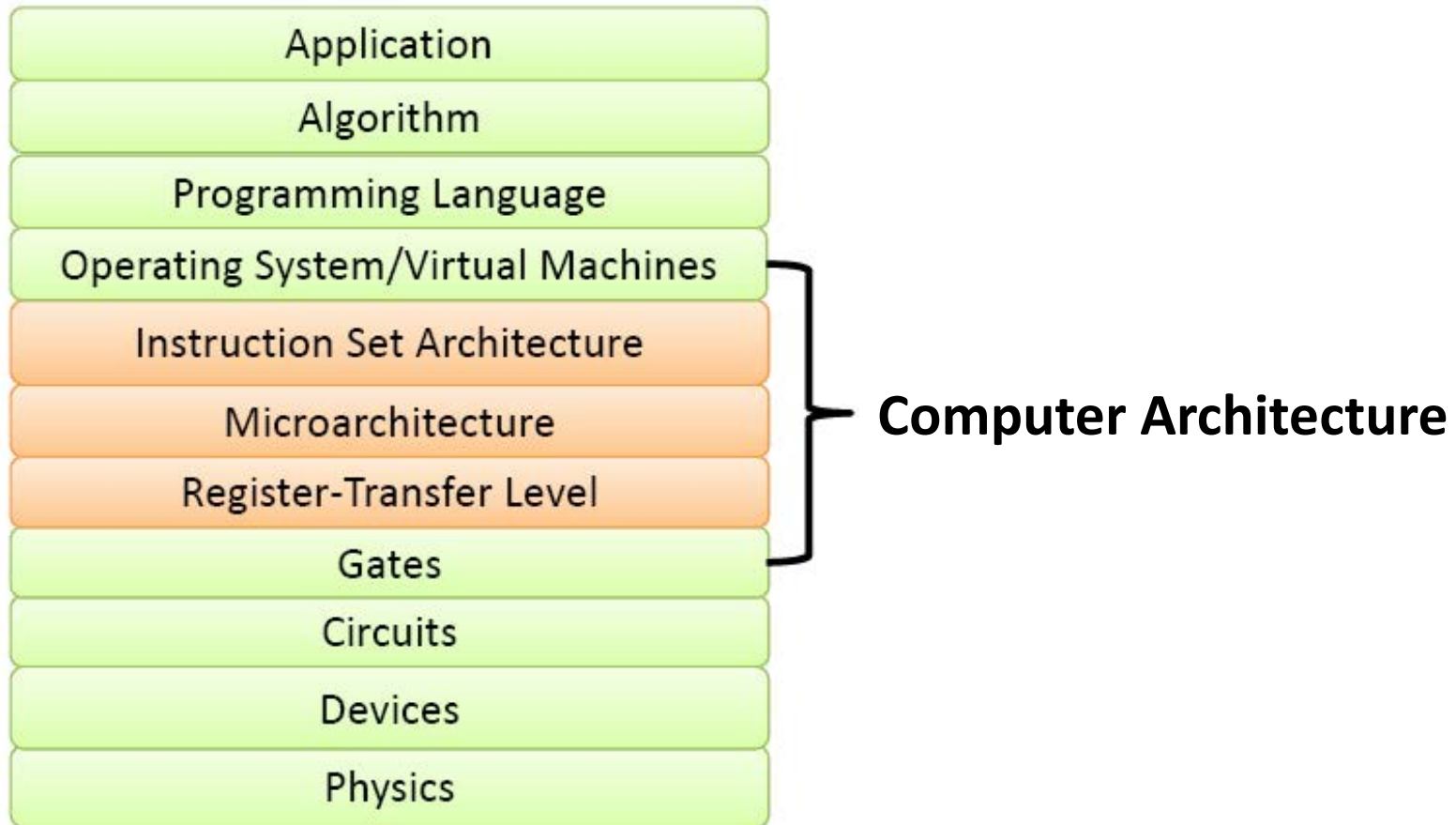


# Abstraction in modern computer system



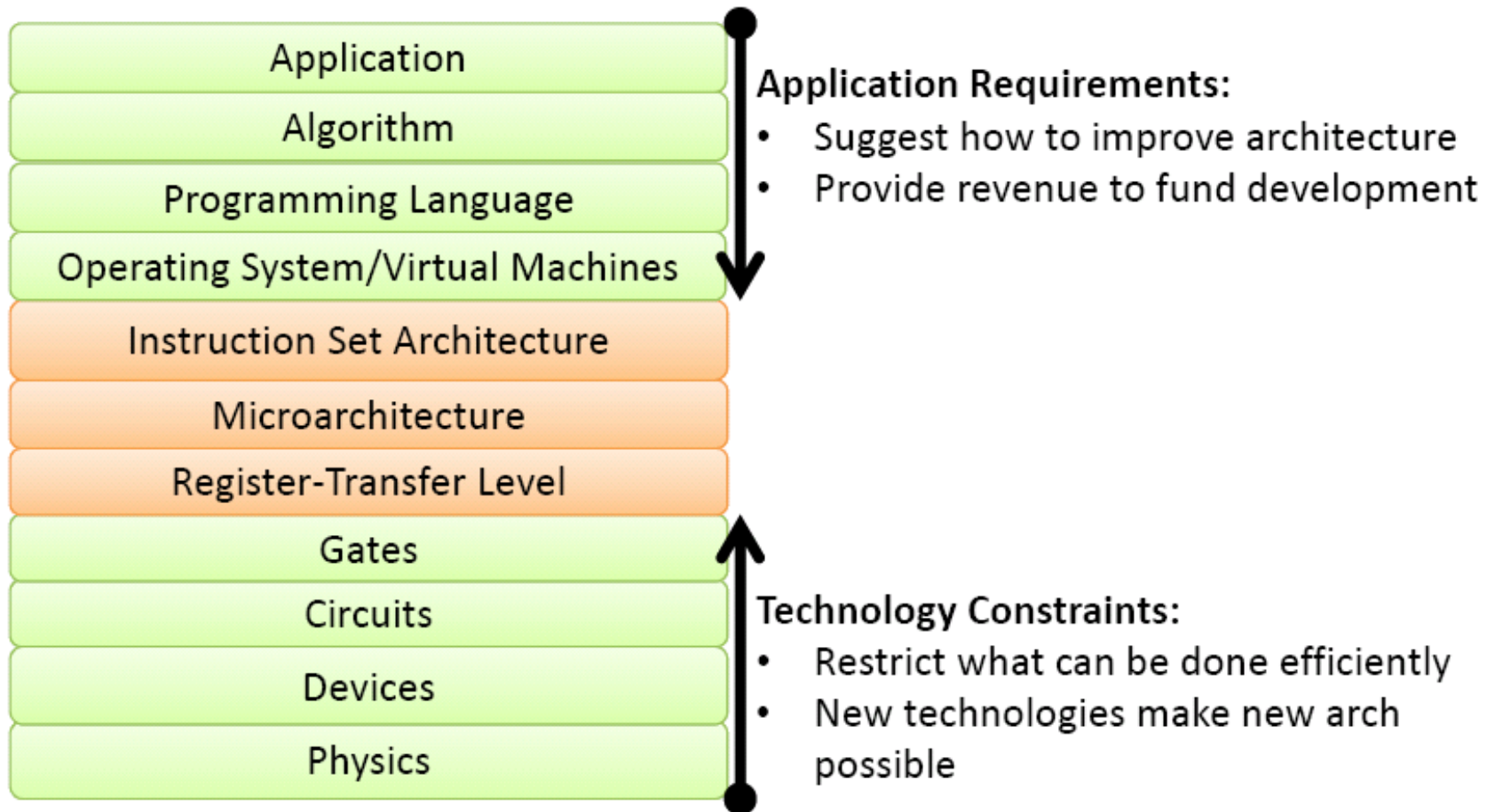
The myopic view of computer architecture: **instruction set architecture**, the interface between software and hardware.

# Abstraction in modern computer system

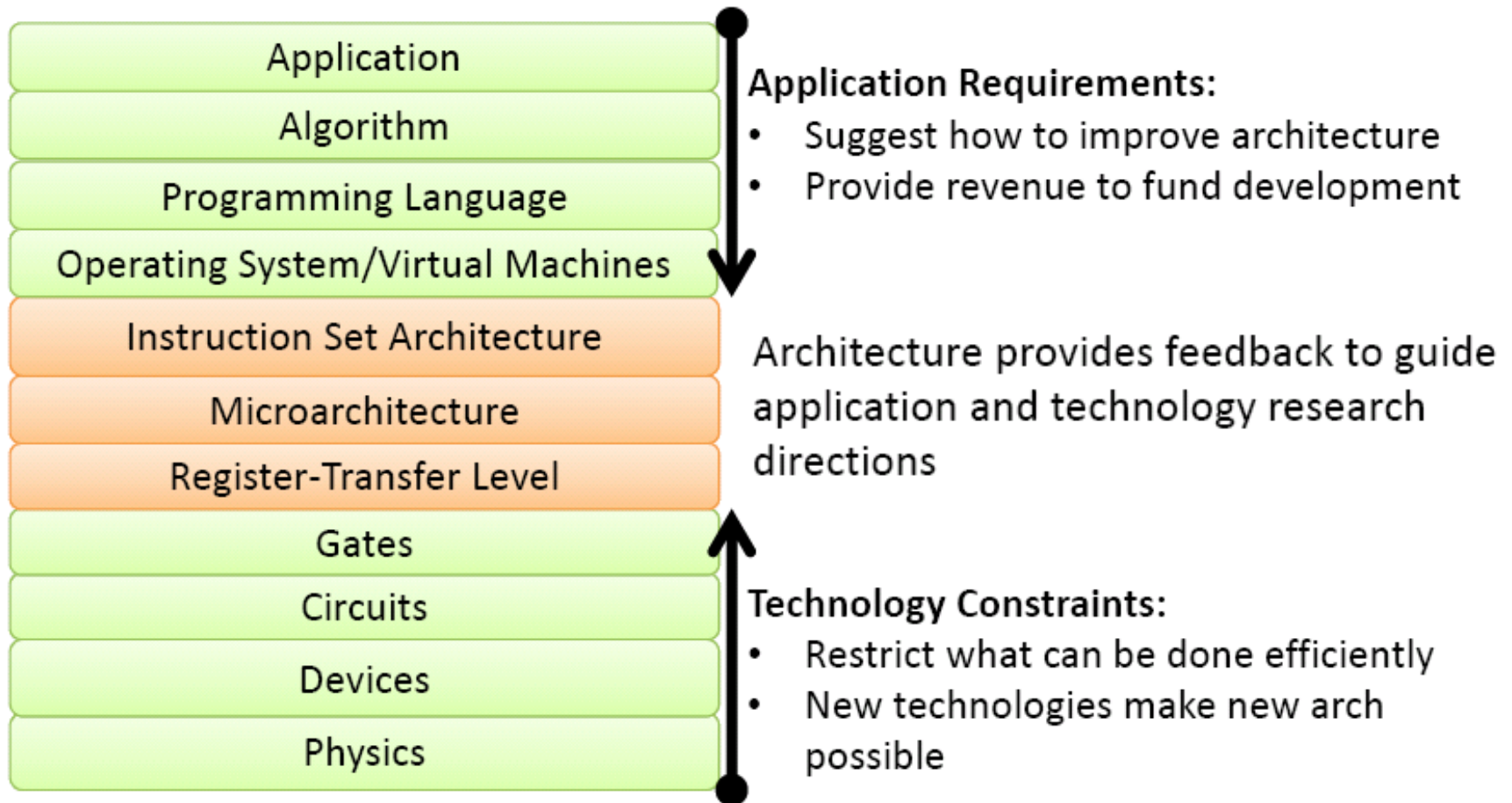




# Computer Architecture is Constantly Changing



# Computer Architecture is Constantly Changing

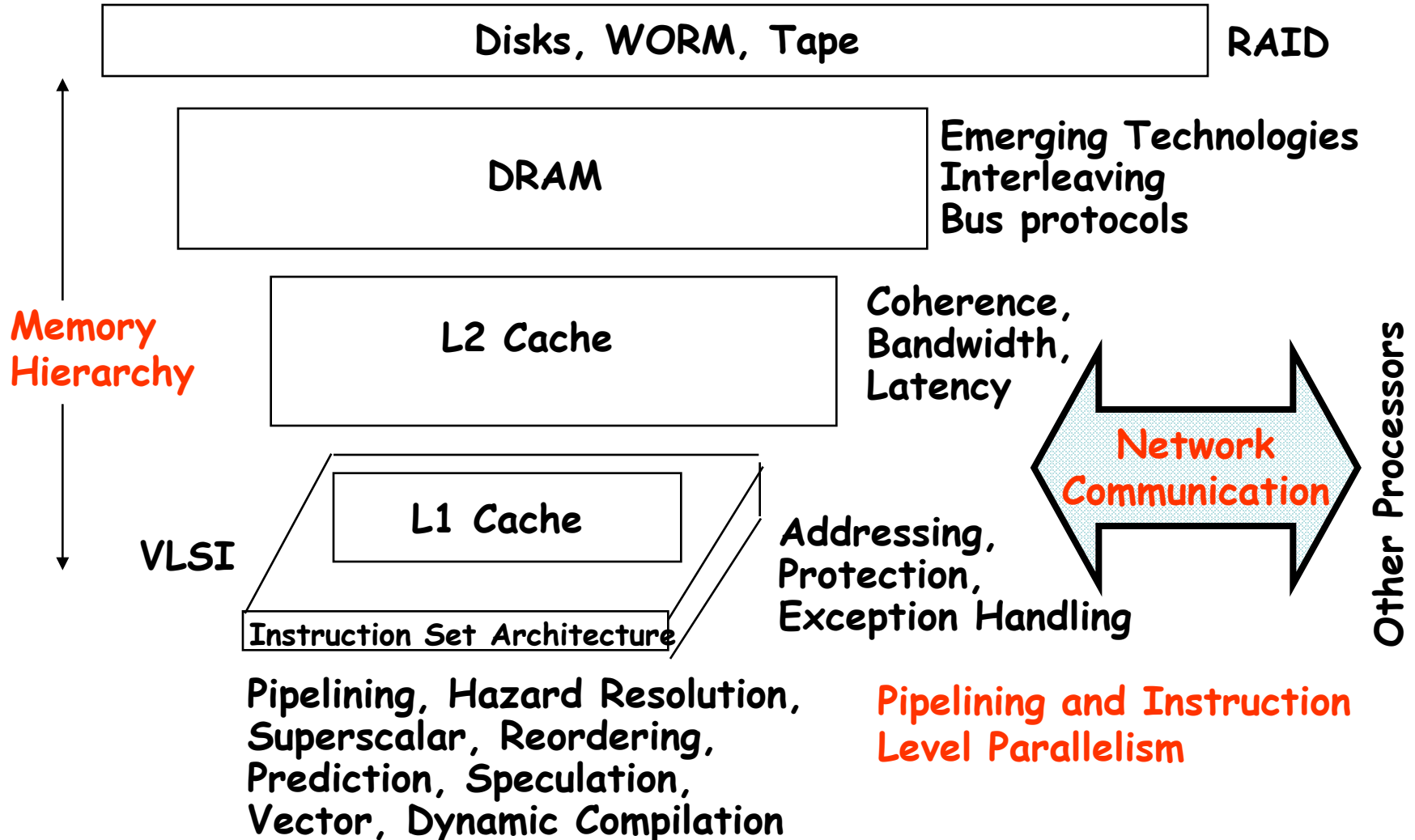


# Computer Architecture's Changing Definition

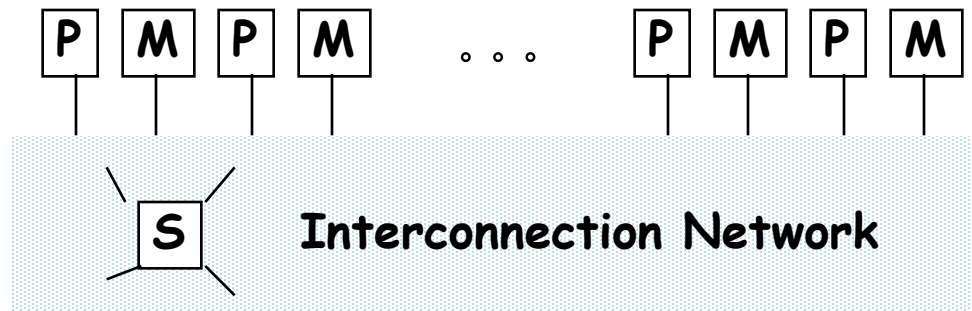
- **1950s to 1960s: Computer Architecture Course: Computer Arithmetic**
- **1970s to mid 1980s: Computer Architecture Course: Instruction Set Design, especially ISA appropriate for compilers**
- **1990s: Computer Architecture Course: Design of CPU, memory system, I/O system, Multiprocessors, Networks**
- **2000s: Computer Architecture Course: Non Von-Neumann architectures, Reconfiguration, Focused MIPs**

# Computer architecture topics

## Input/Output and Storage



# Computer architecture topics



Processor-Memory-Switch

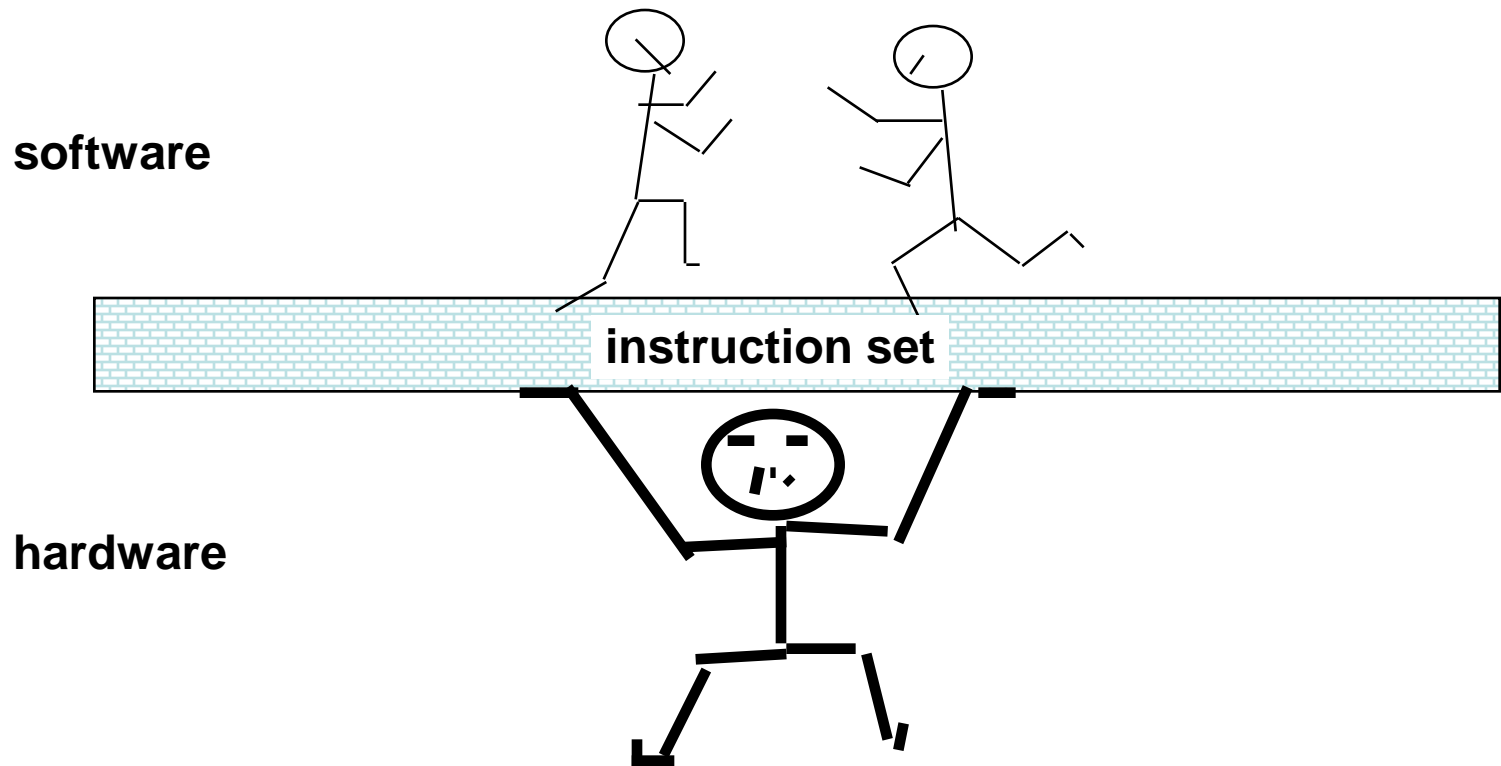
**Multiprocessors**  
**Networks and Interconnections**

Shared Memory,  
Message Passing,  
Data Parallelism

Network Interfaces

Topologies,  
Routing,  
Bandwidth,  
Latency,  
Reliability

# Instruction Set





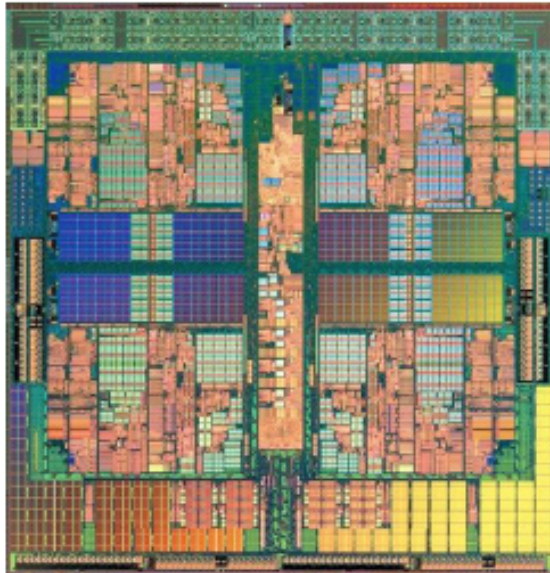
# ISA vs. Microarchitecture

- **Architecture covers all three aspects of computer design – instruction set architecture (ISA), microarchitecture or organization and hardware.**
- **Instruction Set Architecture:**
  - Programmer visible state (register and memory)
  - Operations (Instructions and how they work)
  - Execution semantics (Interrupts)
  - Input/Output
  - Data types/sizes
- **Microarchitecture:**
  - Trade-offs on how to implement ISA for some metrics (speed, energy and cost).
  - Examples: pipeline depths, cache size, execution order, bus widths and ALU widths.

# Same ISA, Different Microarchitecture

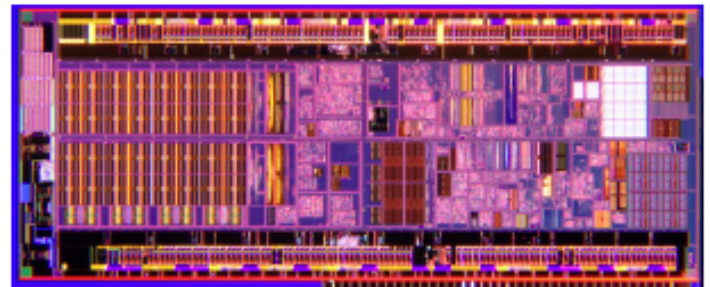
## AMD Phenom X4

- X86 Instruction Set
- Quad Core
- 125W
- Decode 3 Instructions/Cycle/Core
- 64KB L1 I Cache, 64KB L1 D Cache
- 512KB L2 Cache
- Out-of-order
- 2.6GHz



## Intel Atom

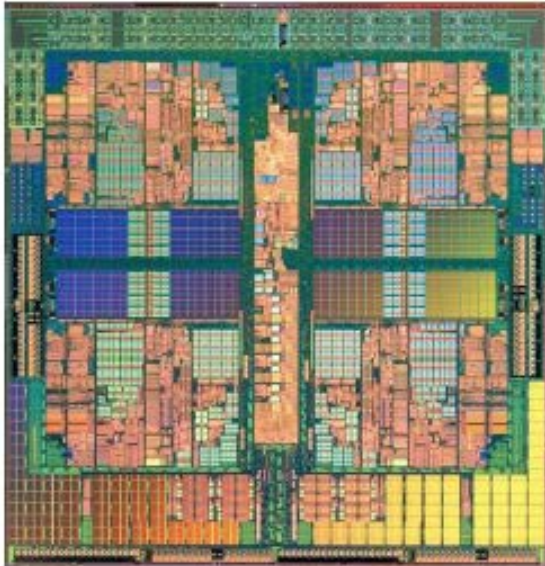
- X86 Instruction Set
- Single Core
- 2W
- Decode 2 Instructions/Cycle/Core
- 32KB L1 I Cache, 24KB L1 D Cache
- 512KB L2 Cache
- In-order
- 1.6GHz



# Diff. ISA, Diff. Microarchitecture

## AMD Phenom X4

- X86 Instruction Set
- Quad Core
- 125W
- Decode 3 Instructions/Cycle/Core
- 64KB L1 I Cache, 64KB L1 D Cache
- 512KB L2 Cache
- Out-of-order
- 2.6GHz



## IBM POWER7

- Power Instruction Set
- Eight Core
- 200W
- Decode 6 Instructions/Cycle/Core
- 32KB L1 I Cache, 32KB L1 D Cache
- 256KB L2 Cache
- Out-of-order
- 4.25GHz

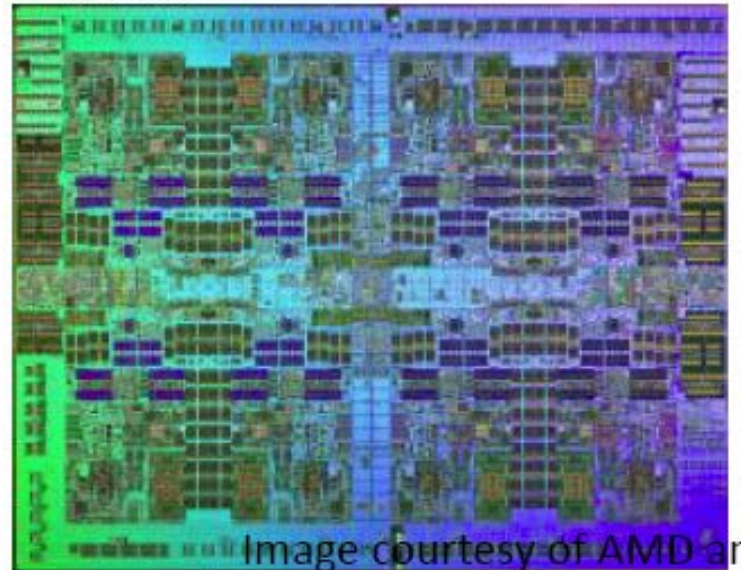
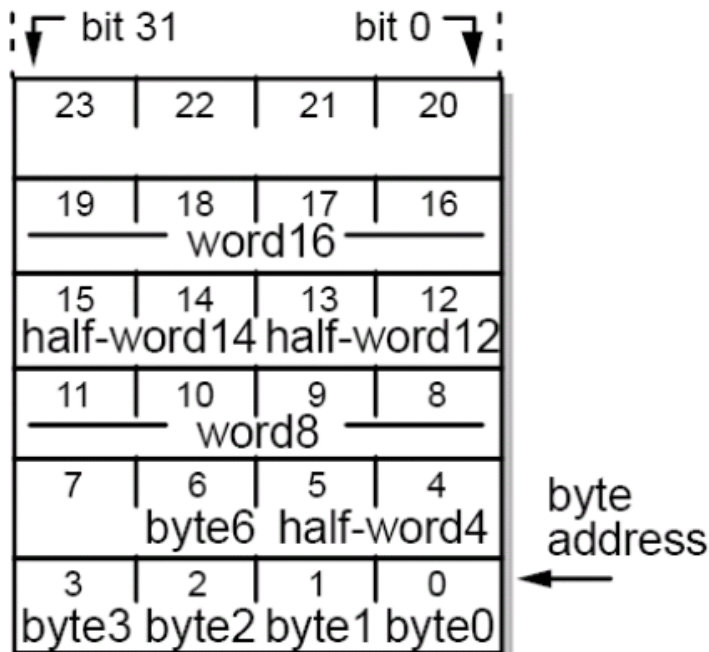


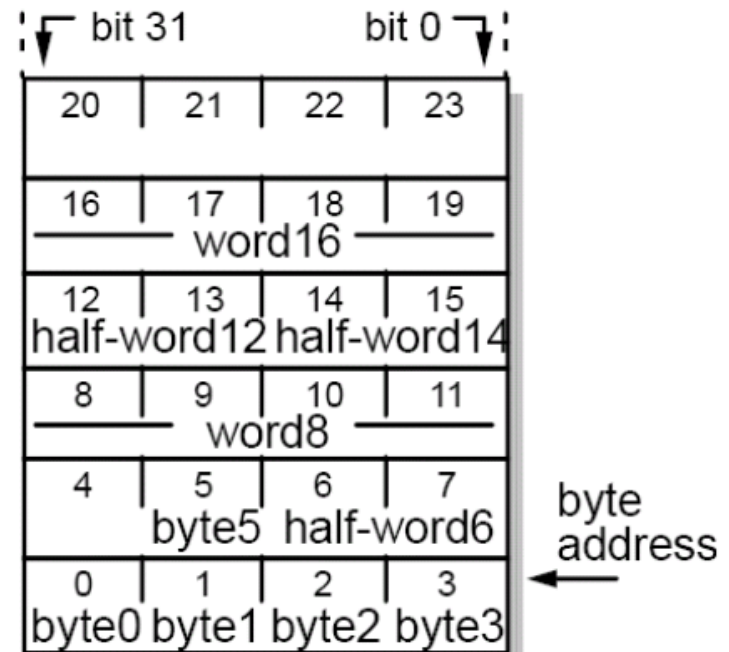
Image courtesy of AMD and IBM

# A Review of ISA (1)

- **Class of ISA: General-purpose register architectures**
  - Register-memory ISAs: 80x86
  - Load-store ISAs: ARM and MIPS
- **Memory address: byte addressing, aligned**



**Little endian**



**Big endian**



# A Review of ISA (2)

Addressing Mode	Instruction	Function
Register	Add R4, R3, R2	Regs[R4] <- Regs[R3] + Regs[R2] **
Immediate	Add R4, R3, #5	Regs[R4] <- Regs[R3] + 5 **
Displacement	Add R4, R3, 100(R1)	Regs[R4] <- Regs[R3] + Mem[100 + Regs[R1]]
Register Indirect	Add R4, R3, (R1)	Regs[R4] <- Regs[R3] + Mem[Regs[R1]]
Absolute	Add R4, R3, (0x475)	Regs[R4] <- Regs[R3] + Mem[0x475]
Memory Indirect	Add R4, R3, @(R1)	Regs[R4] <- Regs[R3] + Mem[Mem[R1]]
PC relative	Add R4, R3, 100(PC)	Regs[R4] <- Regs[R3] + Mem[100 + PC]
Scaled	Add R4, R3, 100(R1)[R5]	Regs[R4] <- Regs[R3] + Mem[100 + Regs[R1] + Regs[R5] * 4]

# A Review of ISA (3)

- **Type and size of operands**
  - 8-bit (ASCII character)
  - 16-bit (Unicode character or half word)
  - 32-bit (Integer or word)
  - 64-bit (double word or long integer)
  - IEEE 754 floating point in 32-bit (single precision) and 64-bit (double precision)
- **Operations**
  - Data transfer
  - Arithmetic/logical
  - Control
  - Floating point



# A Review of ISA (4)

- **Control flow instructions: PC relative addressing**
  - Conditional branches
  - Unconditional jumps
  - Procedure calls and returns
- **Encoding on ISA**
  - Fixed length: easy to decode, RISC arch, eg. ARM, MIPS, PowerPC
  - Variable length: less space in memory and caches, CISC arch
    - eg. 80x86 (1 byte up to 17 bytes)
  - Mostly fixed or compressed:
    - eg. MIPS16, Thumb
    - eg. PowerPC and some VLIW (store instructions compressed and, decompress into instruction cache)
  - Very Long Instruction Word (VLIW): multiple instructions in a fixed length bundle
    - eg. TI C6000

# A Review of ISA (5)

## X86 (IA-32) Instruction Encoding

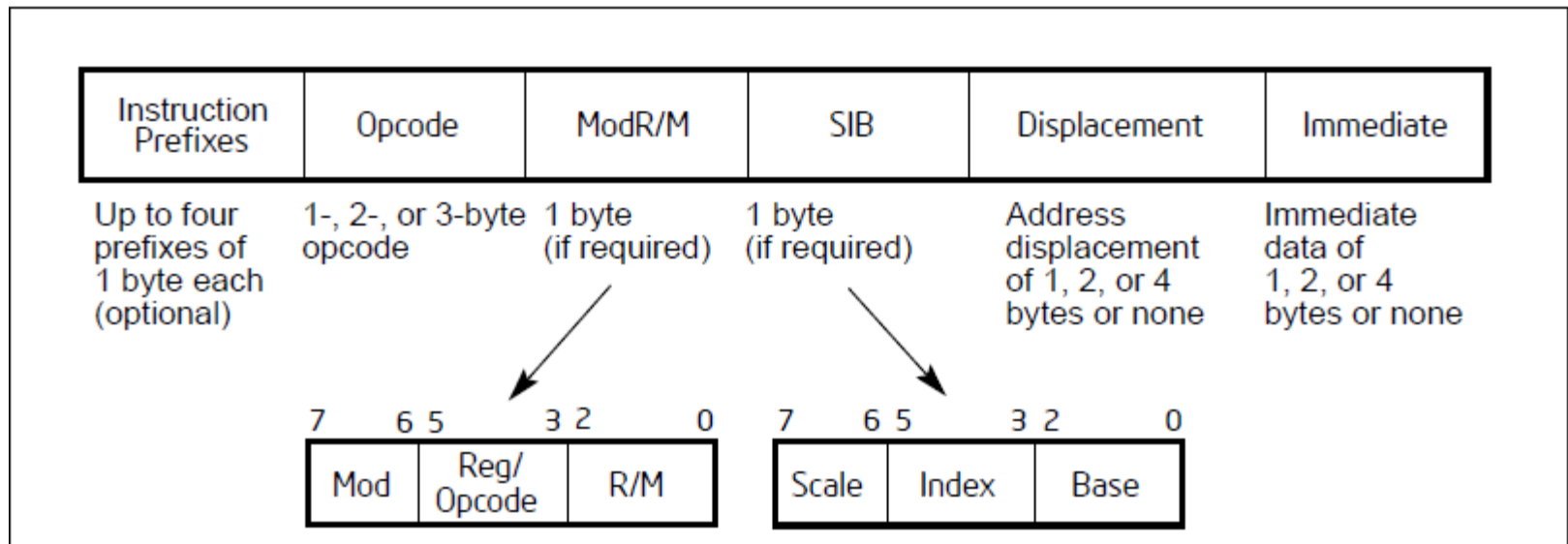


Figure 2-1. Intel 64 and IA-32 Architectures Instruction Format

# A Review of ISA (6)

## MIPS Instruction Encoding

	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits
<b>R:</b>	op	rs	rt	rd	shamt	funct
<b>I:</b>	op	rs	rt	address / immediate		
<b>J:</b>	op	target address				

op: basic operation of the instruction (opcode)

rs: first source operand register

rt: second source operand register

rd: destination operand register

shamt: shift amount

funct: selects the specific variant of the opcode (function code)

address: offset for load/store instructions ( $\pm 2^{15}$ )

immediate: constants for immediate instructions

# A Review of ISA (7)

Arch	Type	# Oper	# Mem	Data Size	# Regs	Addr Size	Use
Alpha	Reg-Reg	3	0	64-bit	32	64-bit	Workstation
ARM	Reg-Reg	3	0	32/64-bit	16	32/64-bit	Cell Phones, Embedded
MIPS	Reg-Reg	3	0	32/64-bit	32	32/64-bit	Workstation, Embedded
SPARC	Reg-Reg	3	0	32/64-bit	24-32	32/64-bit	Workstation
TI C6000	Reg-Reg	3	0	32-bit	32	32-bit	DSP
IBM 360	Reg-Mem	2	1	32-bit	16	24/31/64	Mainframe
x86	Reg-Mem	2	1	8/16/32/ 64-bit	4/8/24	16/32/64	Personal Computers
VAX	Mem-Mem	3	3	32-bit	16	32-bit	Minicomputer
Mot. 6800	Accum.	1	1/2	8-bit	0	16-bit	Microcontroler

# A Review of ISA (8)

- **Technology influenced ISA**
  - Storage is expensive, tight encoding important
  - Reduced Instruction Set Computer
  - Multicore/Manycore: Transistors not turning into sequential performance
- **Application influenced ISA**
  - Instructions for applications: embedded, DSP
  - Compiler technology has improved
    - SPARC register windows no longer needed
    - Compiler can do register allocation efficiently
- **The other challenges beyond ISA design are particularly acute at the present, when differences among ISAs are small and when there are distinct application areas.**