COMPUTER ARCHITECTURE

Code: CT173

Part I: Introduction and Computer Evolution

MSc. NGUYEN Huu Van LONG

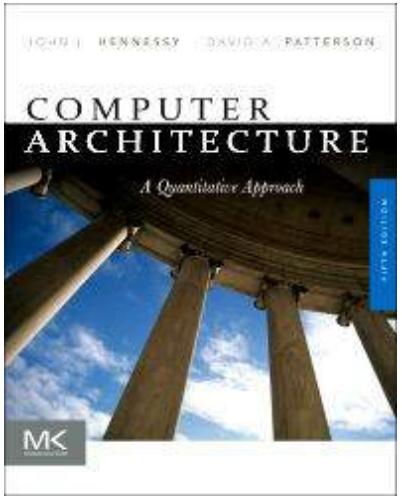
Department of Computer Networking and Communication,

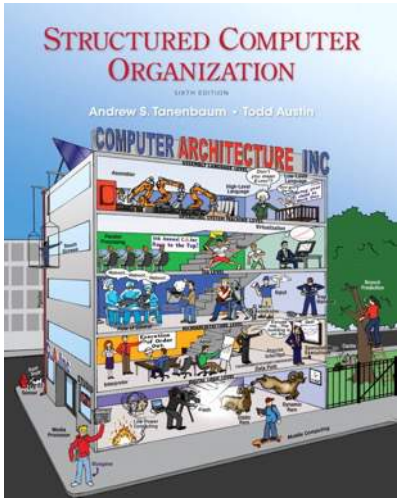
College of Information & Communication Technology,

CanTho University

General Information

- **Instructor:**
  - Name: MSc. **NGUYỄN Hữu Vân LONG**
  - Email: nhvlong@cit.ctu.edu.vn
- **Lectures slides could be downloaded from ELCIT**
  - Old version in Vietnamese
  - New version in English
- **Essential materials:**
  - *Computer Architecture: A Quantitative Approach 5th Edition*
  - *Structured Computer Organization 6th Edition*
  - Lectures slides
- **Evaluation:**
  - *Mid term exam*: 30%
  - *Exercise:* 20%
  - *Final exam*: 50%
  - *Extra score: maximum 1 point*





Computer Classification (Traditional Way)

Super computer CRAY in Swiss

Super computer Blue Gene of IBM

IBM z System z13 and IBM LinuxOne RockHopper

Data General Nova 16 bit computer

LINC mini computer

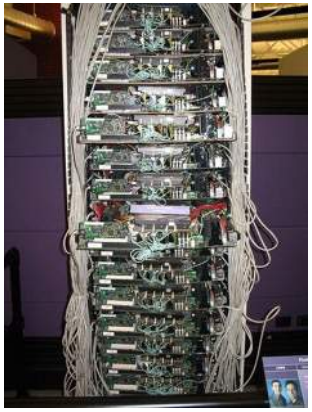
Typically Modern Microcomputer

Computer Classification (Modern Way)

IBM PC 5150 Desktop

Macintosh Motorola 68000

The first laptop Osborne 1



The first Google server

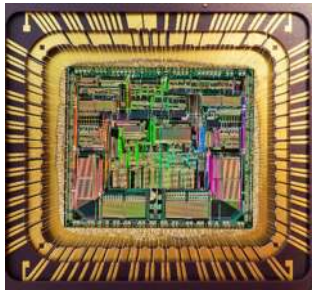
Raspberry Pi Computer

Self Designed Computer

What is Computer Architecture

- **Architecture**
  - Science and art of interconnecting building materials to construct various buildings, subject to constraints
  - Materials: brick, concrete, glass...
  - Buildings: house, office, auditorium...
  - Constraints: cost, safety, time...

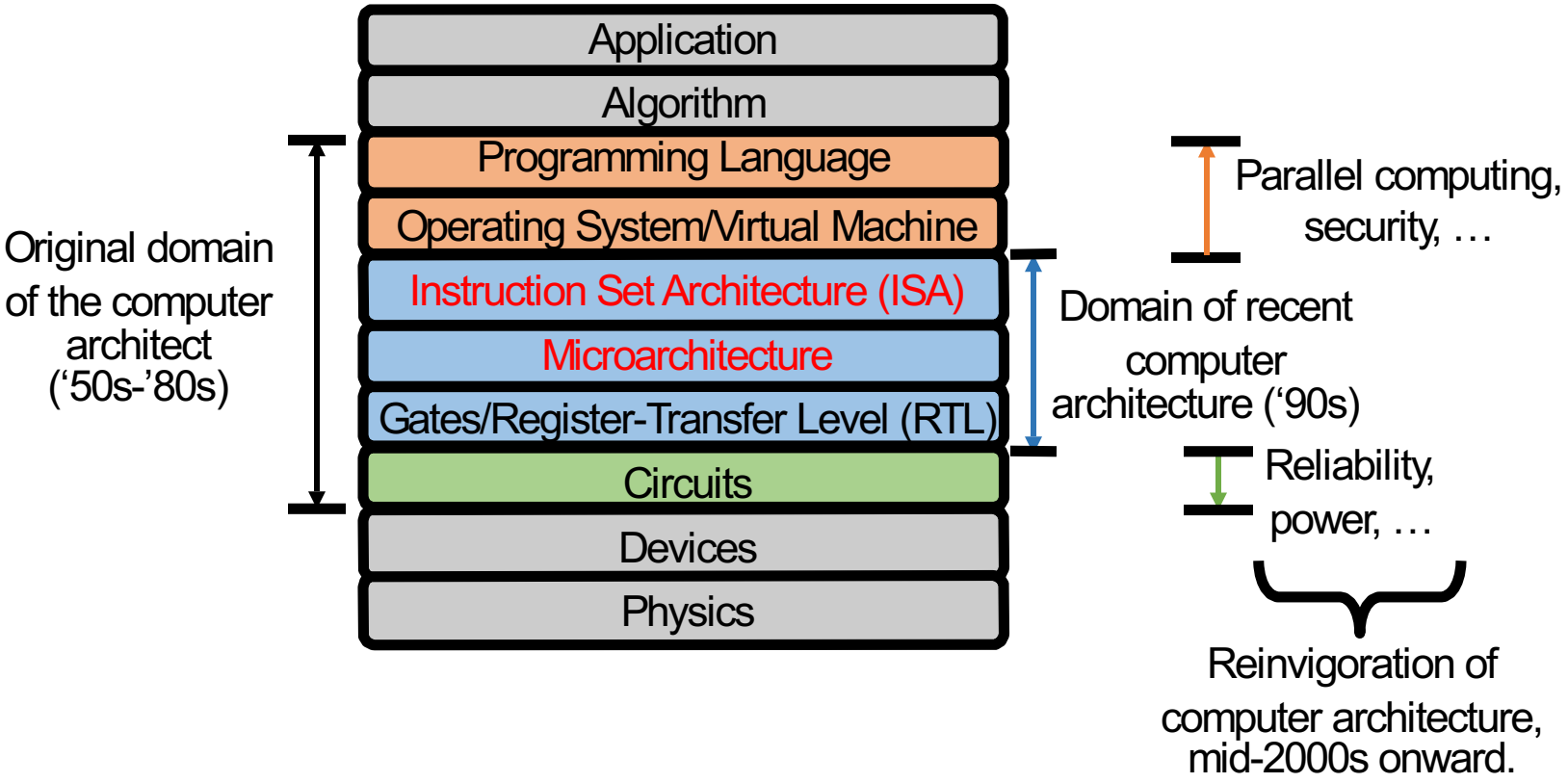
- **Computer Architecture**
  - Science and art of interconnecting hardware components to create computers, subject to constraints
  - Hardware components: circuits, gates, chips...
  - Computers: desktop, server, mobile phone...
  - Constraints: performance, energy, cost, etc...



Role of Computer Architecture knowledge

- **Understand where computers are going**
  - Future capabilities drive the (computing) world
  - Real world-impact: no computer architecture → no computers!
- **Understand high-level design concepts**
  - The best architects understand all the levels
    - Devices, circuits, architecture, compiler, applications
- **Understand computer performance**
  - Learn valid experimental methodologies
- **Write better software**
  - The best software designers also understand hardware
  - Need to understand hardware to write fast software
- **Design hardware**
  - Intel, AMD, IBM, ARM, Qualcomm, NVIDIA, Samsung

Abstraction Layers in Computer Architecture



Critical Aspects of Computer Architecture

- **Instruction Set Architecture (ISA)**
  - Compiler View
  - Instructions visible to the system programmer
  - Opcode, architectural registers, address translation...
- **Microarchitecture**
  - Processor designer View
  - Logical organization that implements the ISA
  - Pipelining, functional units, caches, registers...
- **Circuits**
  - Circuit/chip designer view
  - Detailed logic design and packaging technology
  - Gates, cells, CMOS process...

Course focus

Knowledge need to be to achieved

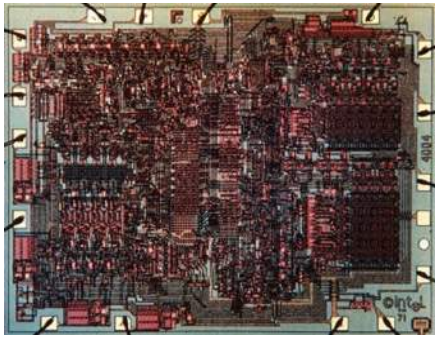
- **The evolution of Computer**
  - Computer generations through history
  - The latest trend computer
- **Data representation in computer system**
  - Number representation: Unsigned Integer, Integer, Floating point number
  - Character representation: ASCII, BCD
- **Instruction Set Architecture – ISA**
  - Essential aspects to develop an ISA
  - Data storage in internal memory
  - Sample ISA for this course

Knowledge need to be to achieved

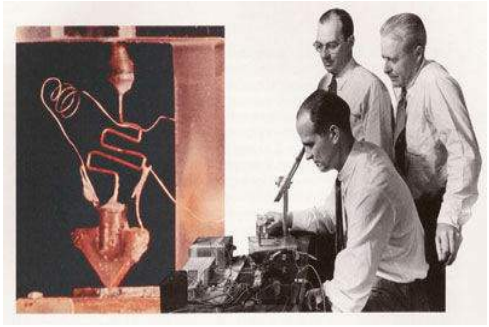
- **Computer operations**
  - Control path and Data path
  - Single cycle vs Multicycle computer
  - Pipeline computer: Advantages and Hazards, Hazards reduce techniques
- **Parallel computers**
  - Instruction Level Parallelism: Superscalar, Very Long Instruction Word - VLIW
  - On chip Parallelism: Multithreading, Multiprocessor
  - Computer networking: Multicomputer
- **Memory system**
  - Fundamental structure of internal memory: SRAM vs DRAM
  - Caches memory
  - Virtual memory

The rise of Electronic Computer

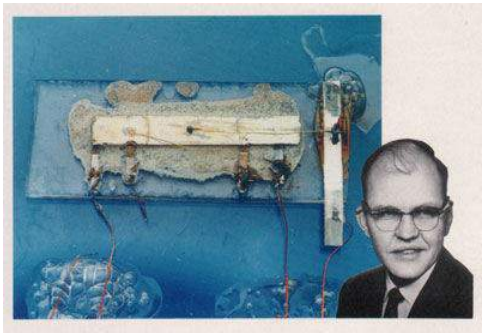
- Since the first electronic computers were created in the late 1940's, performance has increased at a dramatic rate, due to:
  - Integrated circuit technology
  - **Computer Architecture**
  - **Compiling Techniques**
  - Algorithm Developments



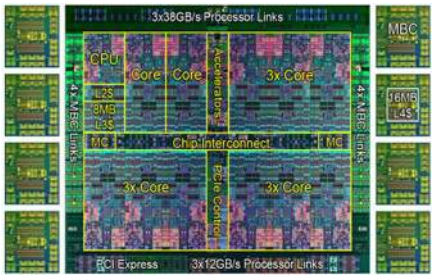
Intel 4004 (1972)



William Shockley, and the first transistor (1947)



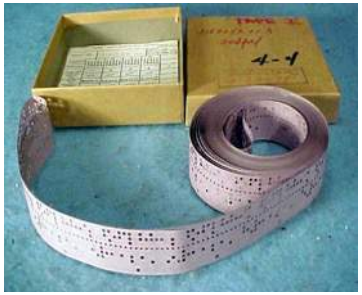
Jack Kilbey with the first IC (1958)

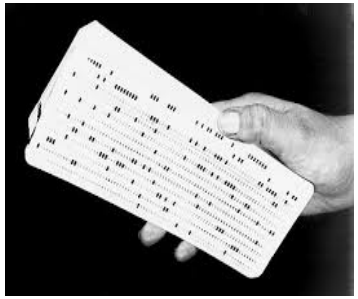


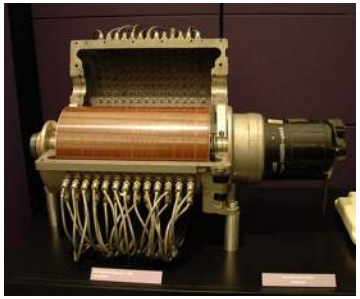
12-core IBM Power 8

First generation 1940 – 1956

- Used vacuum tubes for circuitry, magnetic drums for memory, and were often enormous, taking up entire room!!
- Very expensive, consumed great deal of electricity, generated a lot of heat, which was often the cause of malfunctions.
- Relied on machine language to perform operations, could solve one problem at a given time.
- Input was based on punched cards and paper tape, and output was display on printouts.
- **UNIVAC** and **ENIAC** computer are examples of first generation computing devices.

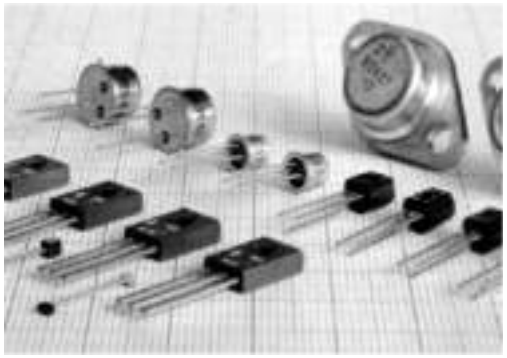






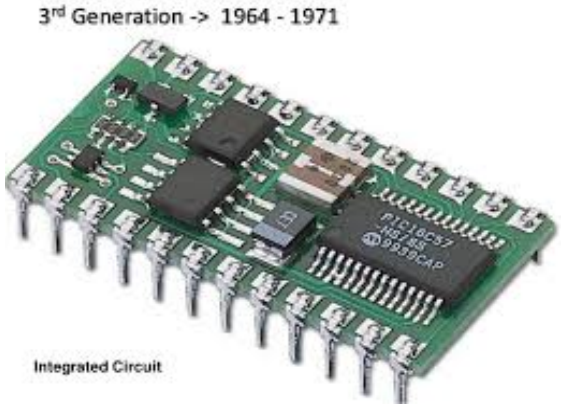
Second generation 1956 – 1963

- Transistors replaced vacuum tubes allowing computers to become smaller, faster, cheaper, more energy efficient and more reliable than their predecessors.
- Still replied on punched card for input and printouts for output
- 2nd generation computers moved from cryptic binary machine language to symbolic or assembly, languages which allowed programmers to specify instructions in words.
- High level programming languages like **COBOL** and **FORTRAN** were used



Third generation 1964 – 1971

- Integrated circuit was used
- Transistors were miniaturized and placed on silicon chips, called semiconductors, which increased the speed and efficiency of computers.
- Instead of punched cards and printouts, users interacted through keyboards and monitors and interfaced with an operating system which allowed the device to run many different applications at one time with a central program that monitored the memory.
- Computers for the first time became accessible to a mas audience because they were smaller and cheaper than their predecessors.



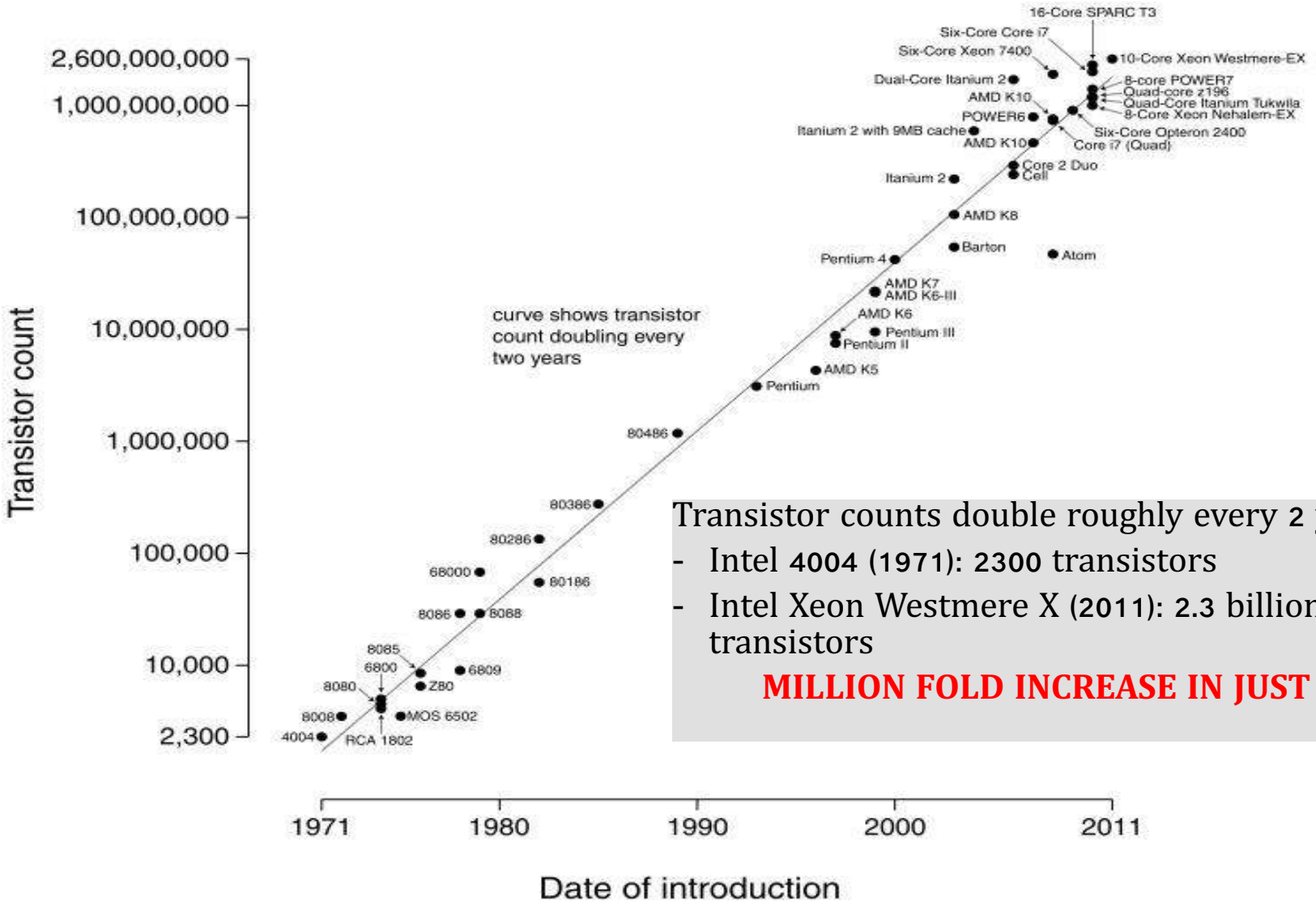
Fourth generation 1971 – Present

- Microprocessor were used
- Could do anything (business, scientific, analysis...) in the palm of the hand.
- In 1981, the first computer for home user is presented by IBM.
- The development of GUI - Graphical User Interface
- Multi tasks could be perform concurrently

Fifth generation (Recently Trends)

- Artificial intelligence are still development, such as face detection, emotion reaction, self learning and experimenting...
- The use of parallel processing and superconductors is helping to make artificial intelligence a reality
- Quantum computation and molecular and nanotechnology will radically change the face of computers in years to come
- The recently achievements: computer could responds to natural input (movements, languages, emotions...) and are capable of learning and self organization

Transistors Evolution

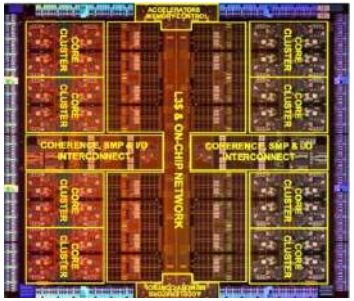


Transistor counts double roughly every 2 years

- Intel 4004 (1971): 2300 transistors
- Intel Xeon Westmere X (2011): 2.3 billion transistors

MILLION FOLD INCREASE IN JUST 40 YEARS

Transistor in latest computers

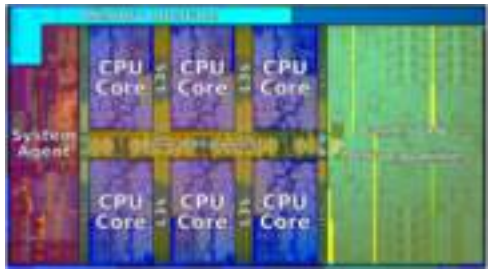


Oracle Sparc M7

- 32 V9 cores grouped to 8 core clusters
- 10B transistors, 700mm2 on die chip
- 20 nm transistor technology
- ISA: Sparc

Apple A11

- 6 cores + 3 Apple designed GPU cores
- 4.3B transistors, 89.23 mm^2
- 10 nm transistors technology
- ISA: ARMv8



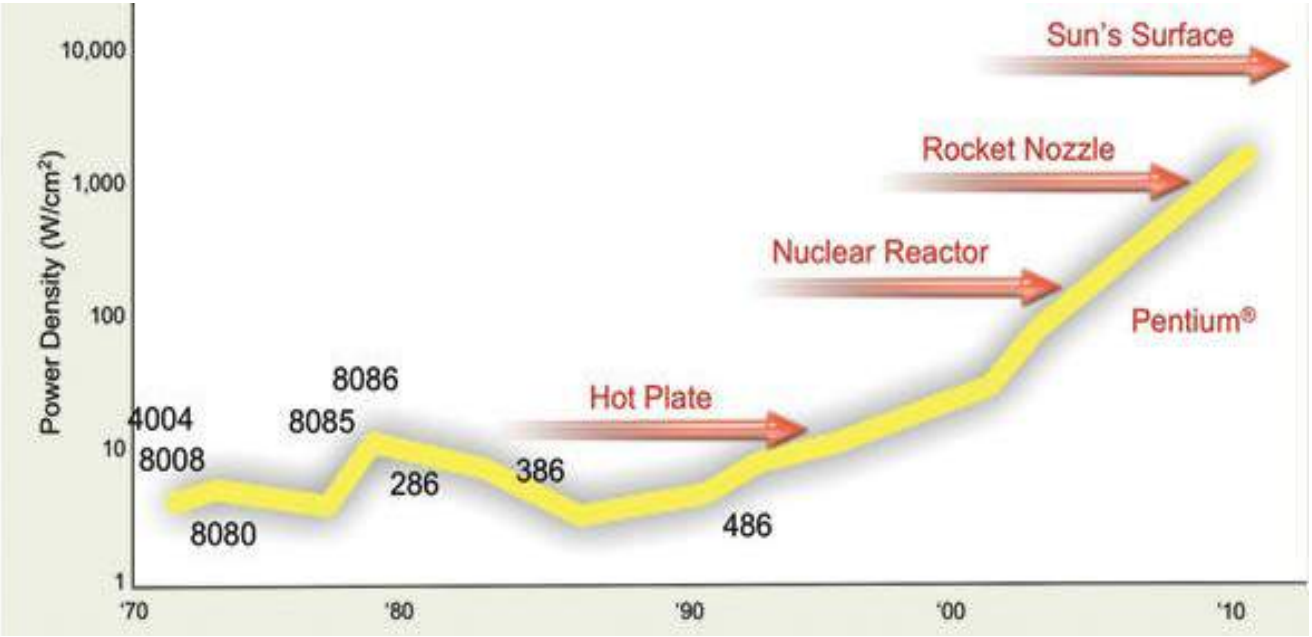
Intel 8th Coffee Lake

- 6 cores + 24 GPU cores
- 2.2 B transistors on 149.6 mm^2 on die-chip
- 14nm tri-gates transistor technology
- ISA: x86-64

Transistor evolution challenges

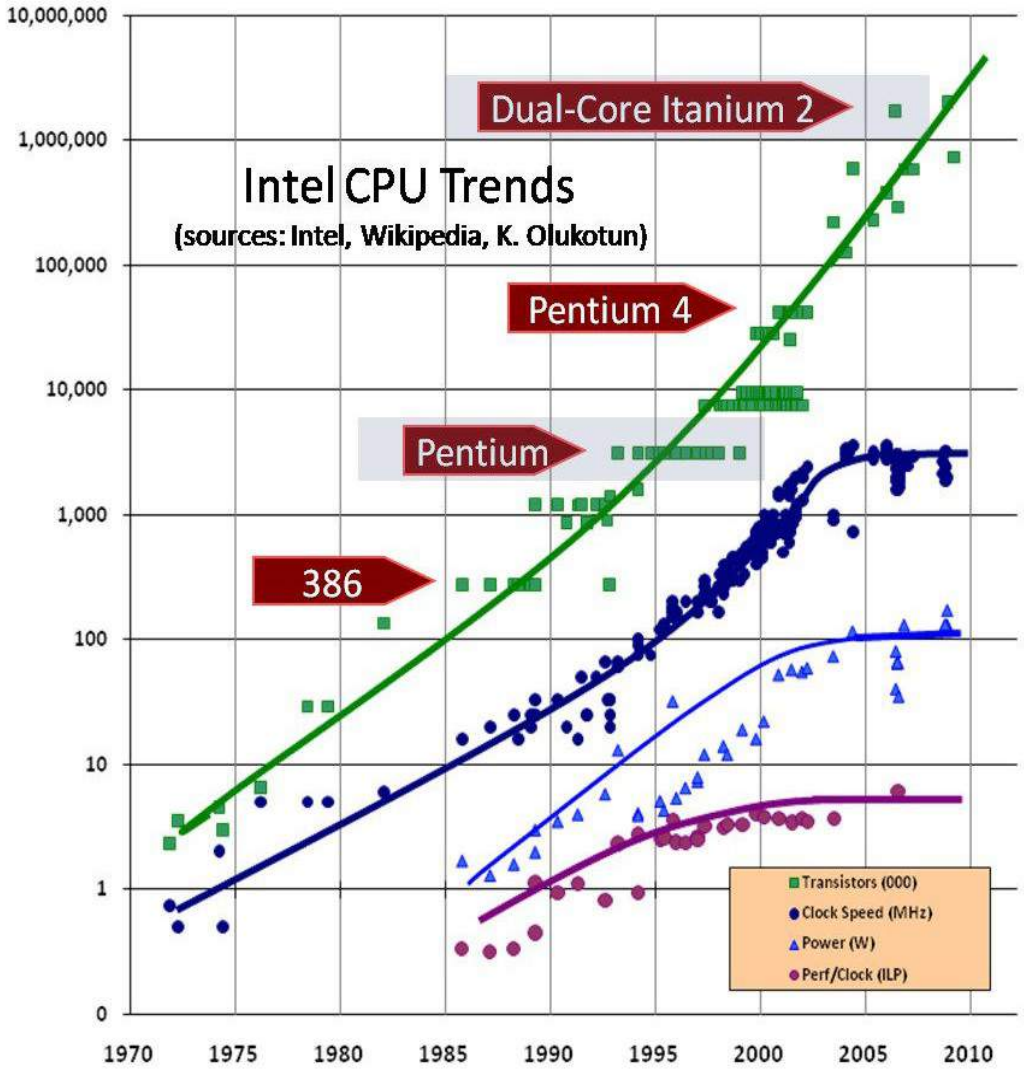
- Number of transistors increase and Size of them decrease
- Integrated density of transistors on chip increase (More transistors on a specified dead area)
- High switching frequency

➔ Power wall or Heat Dissipation Challenge



Pat Gelsinger, Intel Developer Forum, 2004

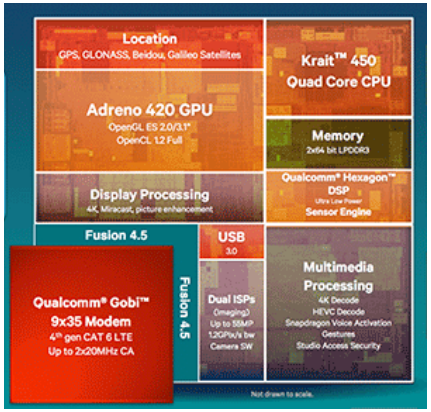
Computer evolution in many ways



- Transistor is not the unique technique and also not the measurement unit to improve the performance of computer.
- Other factors should be considered:
  - Clock speed
  - Power
  - Throughput

New trend for Computer Architecture

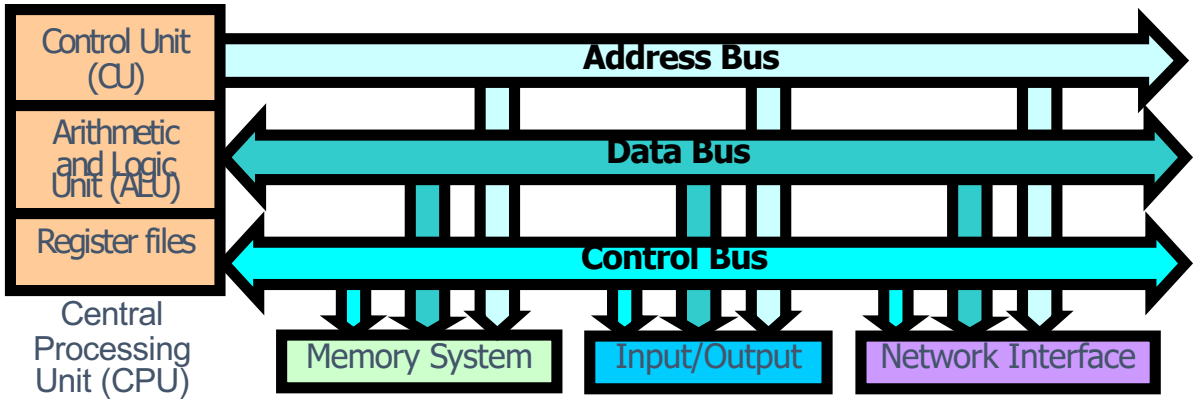
- **Very complex processor design**
  - Out of order execution
  - Control and data speculation: Sophistication branch & Memory dependence predictors
  - Deep cache hierarchies
  - Multiple highly specialized pre-fetchers for both instructions and data
- **Parallelism and integration at chip level**
  - Chip multiprocessors (CMPs)
  - Multi-threading
  - Systems on Chip (CPU, GPU, DSP, accelerators,
  - sensors, memory & storage interfaces…)
- **Power conscious designs**



Qualcomm SnapDragon 805

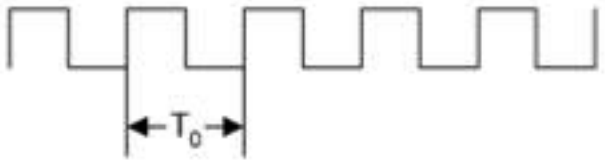
Computer Components

- **Central Processing Unit**: data processing, operates based on the program inside main memory system.
  - Control Unit - CU
  - Arithmetic and Logic Unit - ALU
  - Register files - RF
- **Memory System**: stores data and programs; supplies Write and Read function;
  - Internal Memory
  - External Memory
- **Input – Output System**: support the computer system to communicate with user
  - Peripheral devices
  - IO Modules
- **System Interconnection**
- **Network Interface**



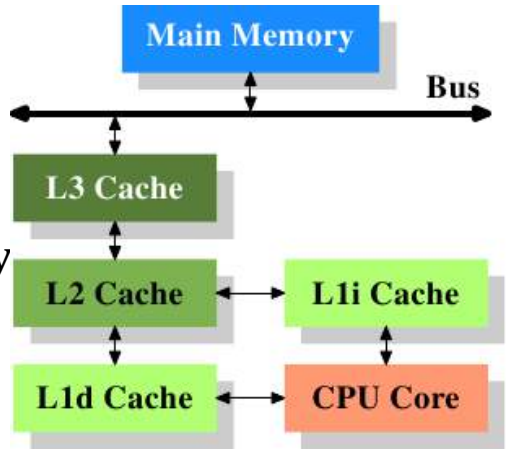
CPU Fundamental Concepts

- **CPU fundamental components**
  - **Control Unit - CU**: control the computer operations based on the dedicated program inside main memory.
  - **Arithmetic and Logic Unit - ALU**: support the arithmetic calculations and logic calculations on CPU.
  - **Register File - RF**: temporary store the data used in CPU's operations.
- **Clock frequency of CPU = Speed of CPU** instead of using Number of instruction completed in 1 second
  - T_0: clock period of CPU
  - $f_0 = 1/T_0$: clock frequency of CPU
  - Each CPU's operation should be done in kT_0
  - T_0 is shorter in time, CPU is faster in operation.
  - Example: 2GHz CPU → $f_0 = 2 \times 10^9$ Hz → $T_0 = 1/f_0 = 1/2\times10^9 =$ 0.5 ns



Memory System

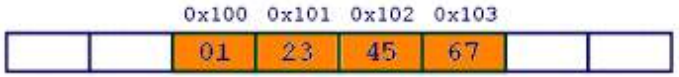
- **Main memory**
  - Store data and programs which are using by CPU
  - Organize as an array or table from 0 to C, with C is the capacity of main memory (1GB, 2GB, 4GB...)
  - Each element or cell in array has an index or address for CPU to locate when needed
  - The content inside cell could be changed regularly meanwhile the cell's physical address is sustainable.
- **Cache memory**
  - Placed between CPU and main memory to enhance the speed in data processing of CPU.
  - Using same structure liked main memory
  - Normally, cache is impressive fast compared to main memory although its capacity is very limited.
  - Cache could be organized unique or spitted.
- **External memory**
  - Store permanently data like software, applications, tools...
  - Integrate with computer system through I/O devices
  - Big size but cost by speed
  - Several type of external memory: floppy disk, hard disk, CD, DVD, USB, memory card...



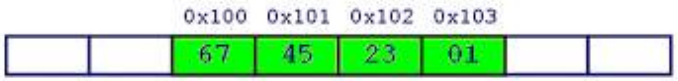
Address	Value
0x00	01001010
0x01	10111010
0x02	01011111
0x03	00100100
0x04	01000100
0x05	10100000
0x06	01110100
0x07	01101111
0x08	10111011
...	...
0xFE	11011110
0xFF	10111011

Memory Addressing

- In today's machine, memory is generally byte addressed - an address refers to the number of bytes counted from the beginning of memory.
- Memory provides access object (data) for bytes (8 bits), half words (16 bits), words (32 bits), and double words(64 bits)
- The type of access object is implied as Opcode in instruction format, i.e LDB - load byte; LDW - load word;
- Byte Ordering
  - **Little Endian**: puts the byte whose address is xx00 at the least significant position in the word. (7,6,5,4,3,2,1,0)
  - **Big Endian**: puts the byte whose address is xx00 at the most significant position in the word. (0,1,2,3,4,5,6,7)
  - Problem: Exchanging data among machines with different orderings



Big Endian

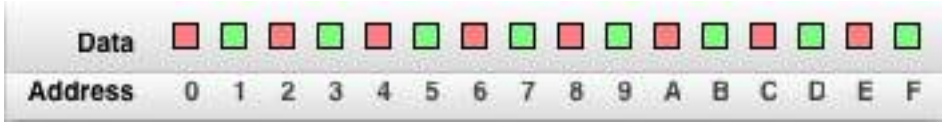


Little Endian

Alignment Issue in Memory

Alignment issue: accesses to objects larger than a byte must be ***aligned***. An access to an object of size s bytes at byte address A is aligned if $A \bmod s = 0$.

- Bytes always aligned or never misaligned
- Half words (16 bits) aligned at byte offsets 0,2,4,6,...
- Words (32 bits) aligned at byte offsets 0,4,8,12,...
- Double words (64 bits) aligned at byte offsets 0,8,16,24,...



How programmers see memory

How processors see memory

Single byte memory access granularity

Issues gaps between programmers and processors

- Your software will run slower.
- Your application will lock up.
- Your operating system will crash.
- Your software will silently fail, yielding incorrect results.

Double byte memory access granularity

Quad byte memory access granularity

Memory Addressing

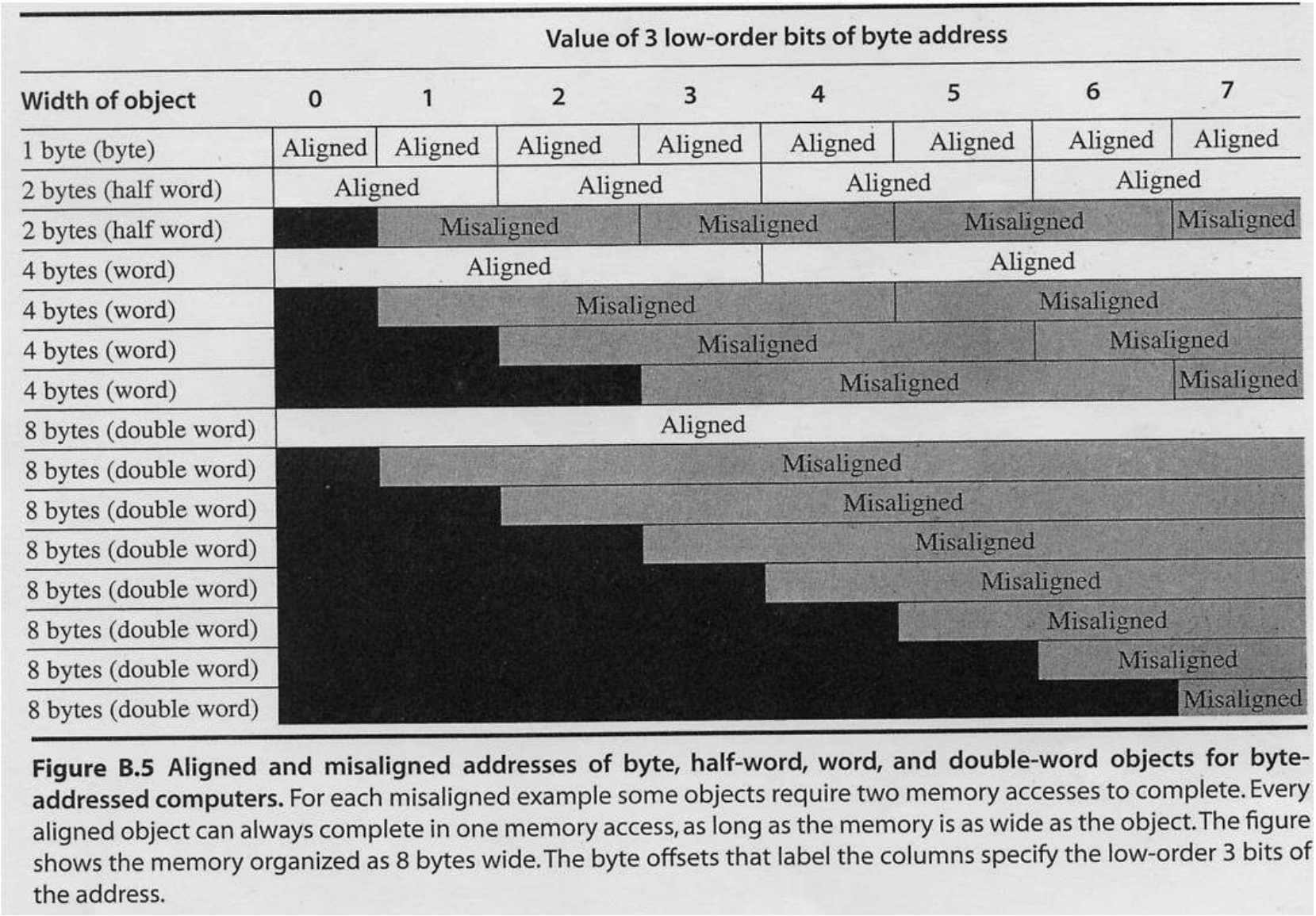


Figure B.5 Aligned and misaligned addresses of byte, half-word, word, and double-word objects for byte-addressed computers. For each misaligned example some objects require two memory accesses to complete. Every aligned object can always complete in one memory access, as long as the memory is as wide as the object. The figure shows the memory organized as 8 bytes wide. The byte offsets that label the columns specify the low-order 3 bits of the address.

Question?