programmable electronic machine :computer

the first all-electronic, programmable, general-purpose computer: ENIAC

◆ Vacuum tube (used 18,000 vacuum tubes, which are electronic switches developed around 1906)

Transistor computers: 2nd generation computers

Integrated a transistor with resistors and capacitors on a single semiconductor chip, which is a monolithic IC

Moore’s law comes into play (2X transistors/chip every 1.5~2 years)

◆ Shrinking transistors

CISC (complex instruction set computer

RISC (reduced ….)

一張含有 文字 的圖片

自動產生的描述

一張含有 桌 的圖片

自動產生的描述

Cloud computing ◆ Warehouse Scale Computers (WSC) ◆ Software as a Service (SaaS) ◆ Portion of software run on a PMD and a portion run in the Cloud

2-fold effects of IC technology scaling on computer performance: • Faster without change of design • More transistors to implement new architecture features

一張含有 文字 的圖片

自動產生的描述

requires innovative architectural ideas to ride the technology scaling

p22 eight great ideas in computer architecture

Operating system: service code p Loading and executing user programs p Handling input/output p Managing memory and storage p Managing networking p Managing system security p Scheduling tasks & sharing resources

⦿ High-level language ◆ Level of abstraction closer to problem domain

Aeeambly language: main focus of ISA

⦿ Algorithm ◆ Determines number of operations executed

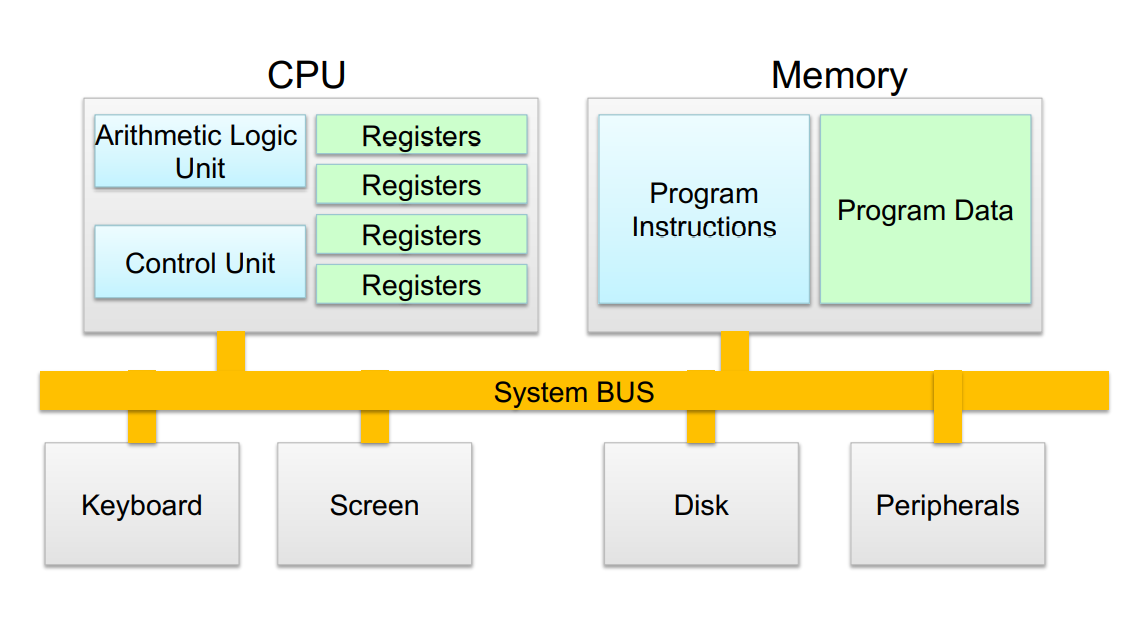
⦿ Programming language, compiler, architecture ◆ Determine number of machine instructions executed per operation

⦿ Processor and memory system ◆ Determine how fast instructions are executed

⦿ I/O system (including OS) ◆ Determines how fast I/O operations are executed

⦿ Same components for all kinds of computer ◆ Desktop, server, embedded

⦿ Input/output includes ◆ User-interface devices p Display, keyboard, mouse ◆ Storage devices p Hard disk, CD/DVD, flash ◆ Network adapters p For communicating with other computers



CPU control unit: finite state machine (FSM)

◆ Retrieves and decodes program instructions ◆ Generate signals to coordinate computer operations: load/store registers, perform ALU functions, take branches, etc

CPU Arithmetic logic unit(ALU): perform mathematical and logical operations

◆ Capacitive allows multi-touch controls 多點觸控

//////////inside CPU/////////////////////////////////////

CPU cores + GPU cores ◆ Datapath: performs operations on data ◆ Control: sequences安排…的順序；發現…的順序 datapath, memory, ..

Cache memory ◆ Small fast SRAM memory for immediate access to data

DRAM ◆ Dynamic random access memory

不穩定的；易變的；易怒的，喜怒無常的volatile 易揮發的

◆ Loses instructions and data when power off

///////////////////////////////////////////////////////////////////////////

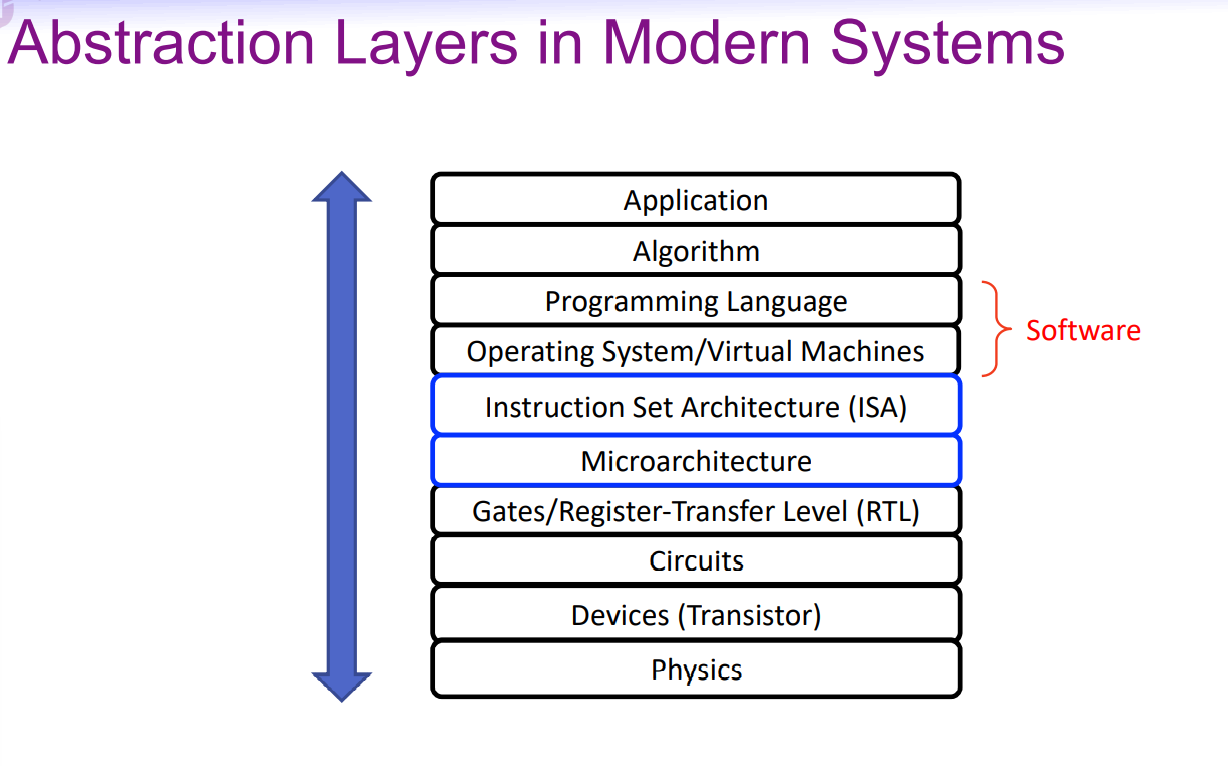
Non volatile secondary memory : magnetic disk ,flash memory(記憶卡隨身碟) ,optical disk(光碟)

LAN Ethernet, WAN Internet

Silicon Can add materials to allow tiny areas to transform into one of three devices: conductors, insulators, switches

一張含有 文字 的圖片

自動產生的描述



⦿ Abstraction helps us deal with complexity ◆ Hide lower-level detail

⦿ Instruction set architecture (ISA) ◆ The hardware/software interface

⦿ Application binary interface (ABI) ◆ The ISA plus system software interface

⦿ Implementation (microarchitecture) ◆ The details underlying an interface

Execution time (response time, latency): how long it takes to do a task (focusing more on non-interactive apps)

Throughput: total work done per unit time p E.g., tasks/transactions/… per hour

Performance = 1 / Execution time

X is n time faster than y performanceX/performanceY = n

Execution time:

◆ Elapsed time (or (total) response time, wall clock time):

Total time to complete a program, including everything: l Memory accesses, I/O activities, OS overhead, idle time

Determines system performance

◆ CPU time (= User CPU time + System CPU time):

Time the CPU spent processing a job: discounts I/O time, other jobs’ shares

CPU time of a program further consists of

1. User CPU time: CPU time spent in the specific program

2. System CPU time: CPU time spent in OS performing tasks on behalf of this program

Computer time: seconds or system clks

Clock period = longest paths between registers (complexity of computation

////////////////////////

CPU time = CPU clock cycles x clock cycle time = CPU clock cycles/ clock rate

CPU performance can be improved by: reduce num of clock cycles, increase clock rate

一張含有 文字 的圖片

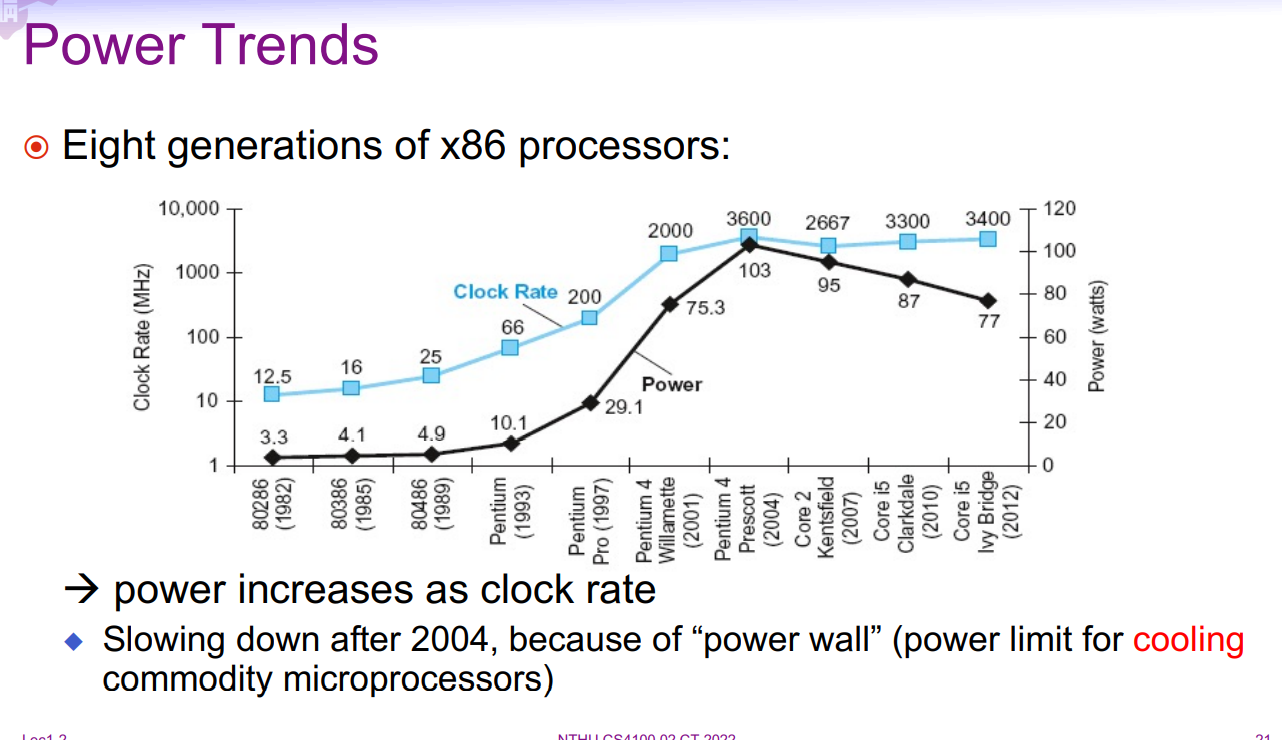
自動產生的描述

一張含有 文字 的圖片

自動產生的描述

一張含有 桌 的圖片

自動產生的描述



⦿ For IC technology based on CMOS, primary energy consumption is dynamic energy

Energy ∝ 1/ 2 ×Capacitive load × Voltage^2

Power ∝ 1/2 ×Capacitive load × Voltage^2× Frequency switched

! Problem with lowering voltage ◆ Static energy consumption (due to leakage current, even when a transistor is off) becomes dominant

The power wall ◆ We cannot reduce voltage further ◆ We cannot remove more heat

◆ A 2-core chip normally consumes less power than a 1-large-core chip of same size and performance

⦿ Free lunch for software in single-core era ◆ IC technology scaling (e.g., clock rate) automatically improves program performance ◆ Architecture innovation on instruction level parallelism lets hardware ➔ exploit parallelism among instructions, and ➔ execute multiple parallel instructions at once p Programmer/compiler can view hardware as executing instructions sequentially and no need to change programs

Problems with Multiprocessors ⦿ Require programmers and compilers to be aware of the parallel hardware and to explicitly rewrite their programs to be parallel ⦿ Hard to do: ◆ Need to program for performance, not just correctness ◆ Load balancing parallel tasks ◆ Optimizing communication and synchronization among parallel tasks

easy to compare two computers using one program, but what about multiple programs? ⦿ Why compare performance using multiple programs? à benchmarking

◆ A standard set of programs specifically chosen to measure and compare computer performance p Represent a workload that will predict the performance of the actual workload

一張含有 桌 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

⦿ Performance is specific to a particular program/s ◆ Total execution time is a consistent summary of performance

⦿ For a given architecture performance increases come from: ◆ Increases in clock rate (without adverse CPI affects and power limits) ◆ Improvements in processor organization that lower CPI ◆ Compiler enhancements that lower CPI and/or instruction count

一張含有 文字 的圖片

自動產生的描述

Corollary;必然結果 直接結果

Pitfall 隱患 陷阱

一張含有 文字 的圖片

自動產生的描述

Fallacy 謬見 謬誤

⦿ Designing for performance and designing for energy efficiency are unrelated goals

⦿ Energy = power x time ◆ Reduced time often leads to less energy

一張含有 文字 的圖片

自動產生的描述

Eight great architecture ideas and hierarchical layers of abstraction ◆ In both hardware and software

⦿ Execution time: the best performance measure

Power is a limiting factor ◆ Use parallelism to improve performance

一張含有 文字 的圖片

自動產生的描述