DEPT. OF ELECTRONICS ENGINEERING & INST. OF ELECTRONICS NCTU

Computer Architecture

Lecture 0: Introduction

Chih-Wei Liu 劉志尉

National Chiao Tung University

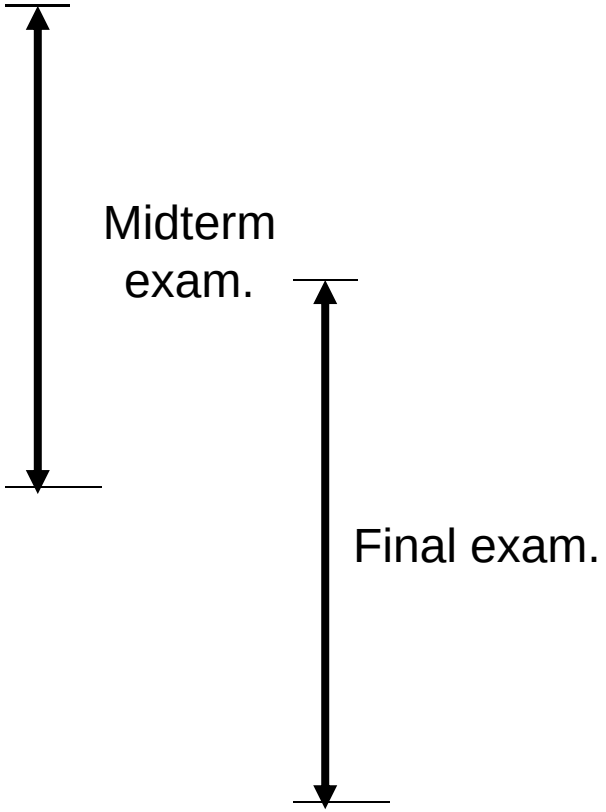
cwliu@twins.ee.nctu.edu.tw

Course Information

- Lecture:
  - Chih-Wei Liu 劉志尉 cwliu@twins.ee.nctu.edu.tw
  - 5731685, ED618
- Teaching Assistant:
  - 楊承彥　張鈞豪
  - 54225, ED412
- Course website: http://twins.ee.nctu.edu.tw
- Textbook: **J. L. Hennessy and D.A. Patterson, *Computer Architecture: A Quantitative Approach*, 5th Edition, Morgan Kaufmann Publishers, 2012**
- Prerequisites:
  - Computer organization
  - Computer programming

This Course Has 5 Modules

- Module 1
  - Instruction set architecture (ISA)
  - Pipelining and Hazards
- Module 2
  - Memory hierarchy
  - Caches and virtual memory
- Module 3
  - Instruction-level Parallelism
  - Branch prediction
  - Speculation and Reorder buffer
- Module 4
  - Thread-level Parallelism
  - Cache coherent protocols
- Module 5
  - Data-level Parallelism
  - Vector machines

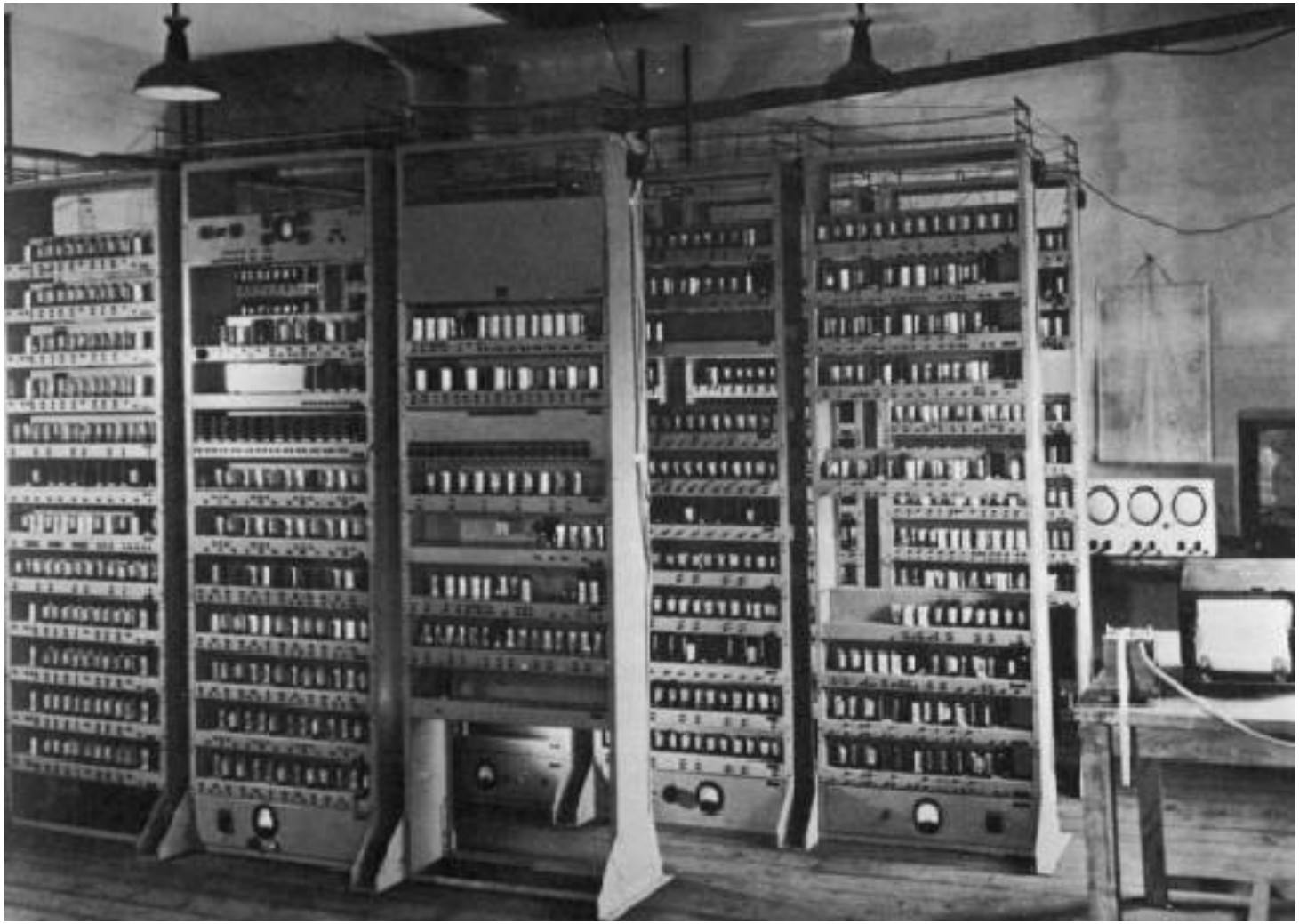


Course Grade (tentative)

- Lectures, Homework, and Quizzes: 25%
  - Adapted from Prof. David Patterson's class notes
  - Please avoid arriving late or leaving early
  - One problem sets with respect to each chapter
  - One-page reading report with respect to each lecture
  - Homework should be handed in on time
- LAB & Project: 25%
- Midterm and Final Exams.: 50%
- (Extra points 5%)

Computer ?

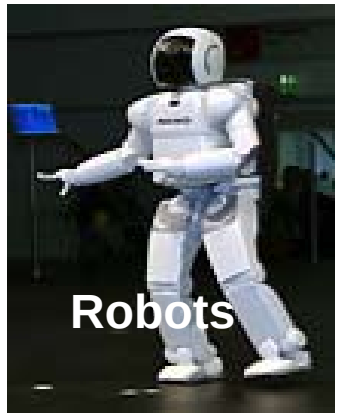
Computing Devices Then…



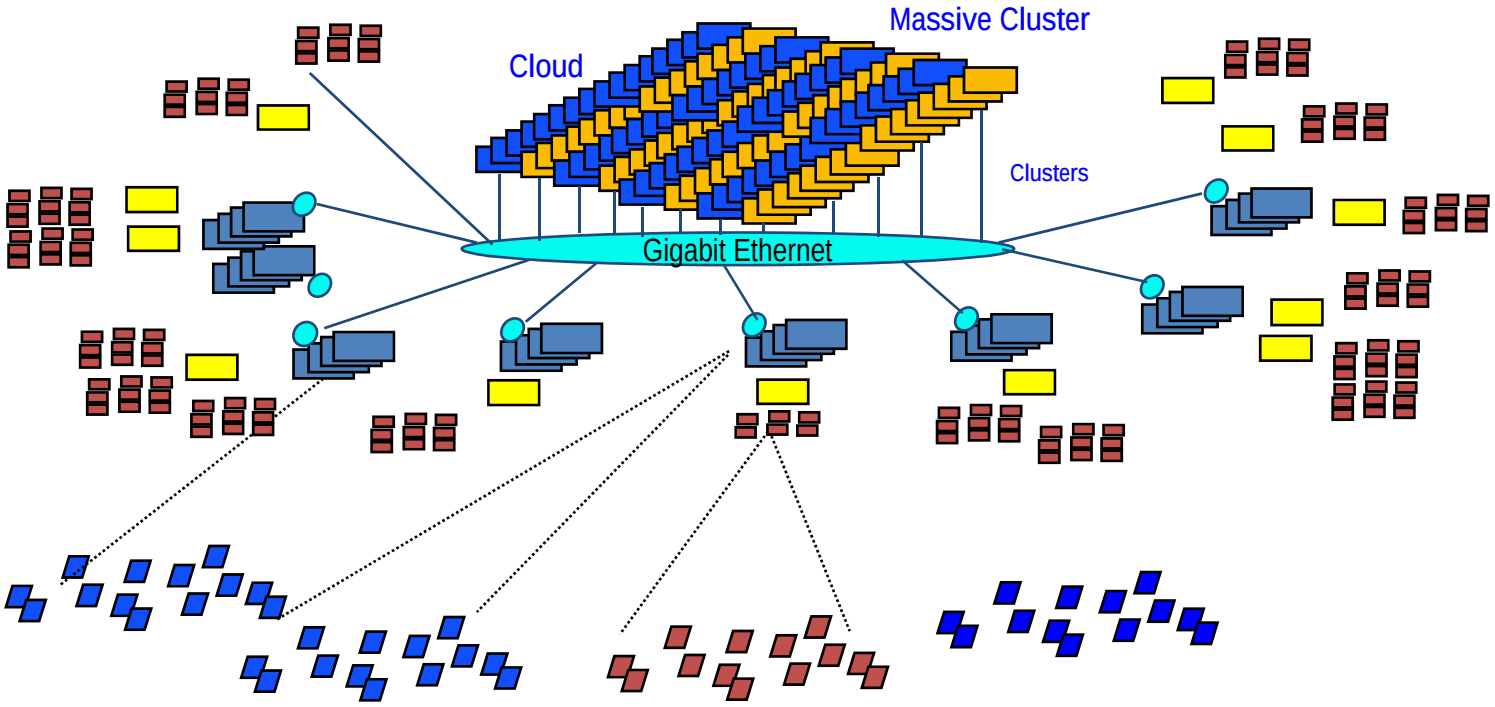
EDSAC, University of Cambridge, UK, 1949

Computing Systems Today …

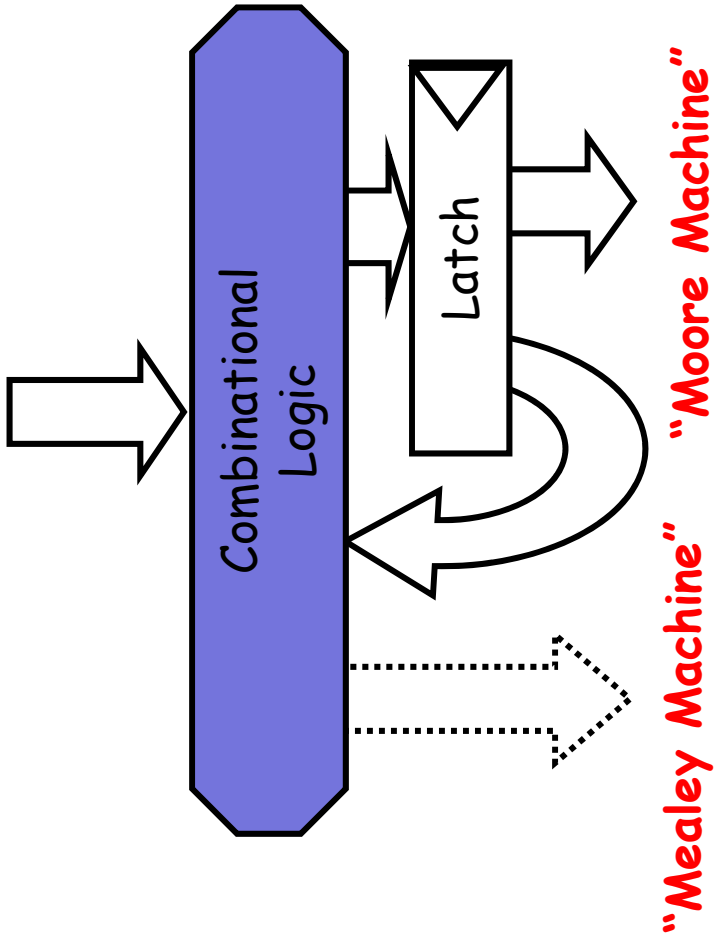
- Computers & Microprocessors in everything
  - Vast infrastructure behind them

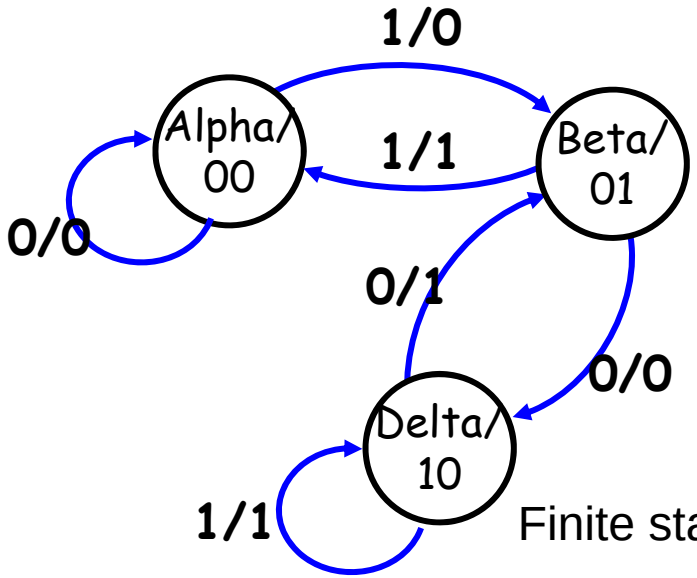


Overview of Computer Systems



Computer is a State-Controlled Machine: Implementation as Comb logic + Latch



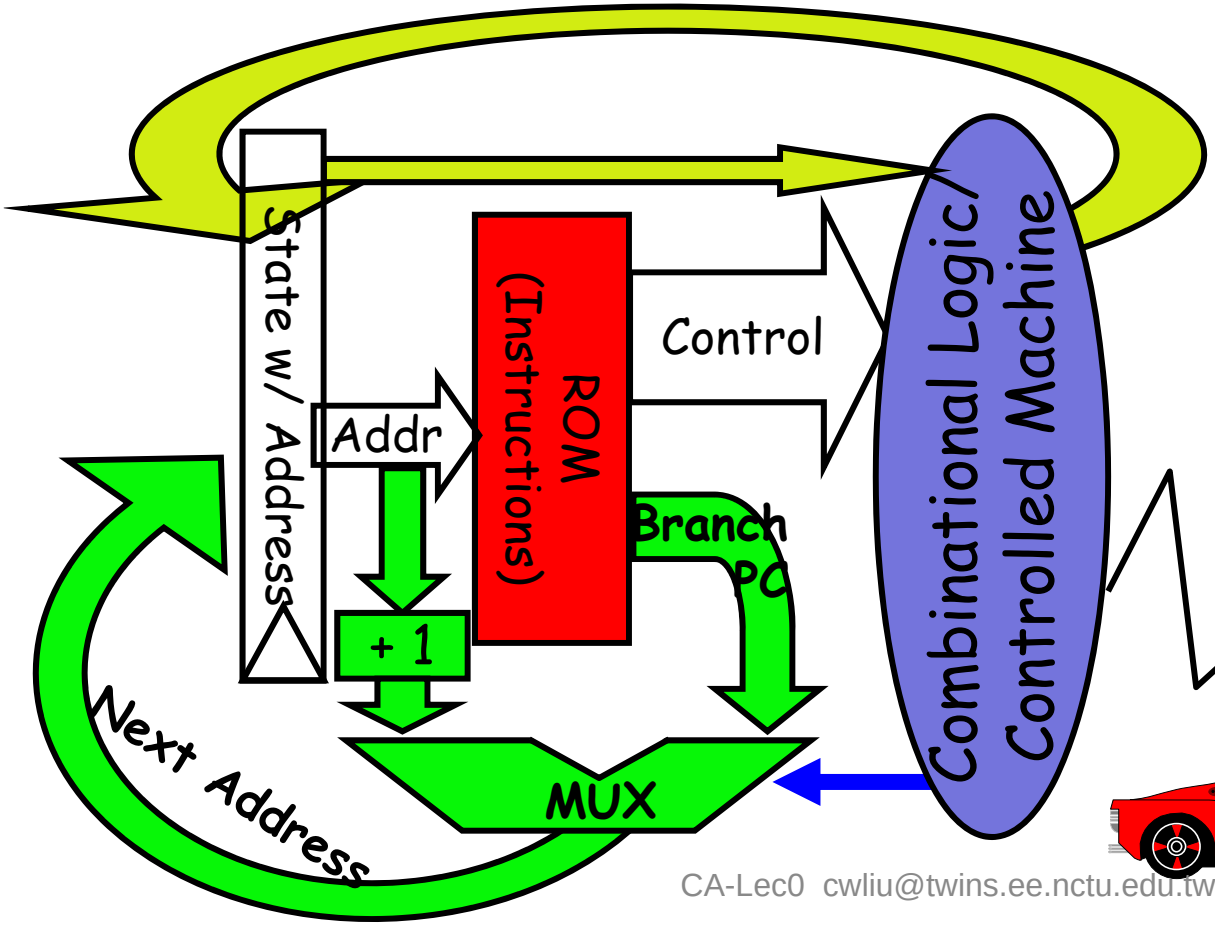


Finite state machine

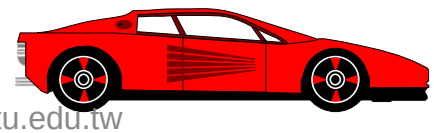
Input	$State_{old}$	$State_{new}$	Div
0	00	00	0
0	01	10	0
0	10	01	1
1	00	01	0
1	01	00	1
1	10	10	1

Computer is a Microprogrammed Controller

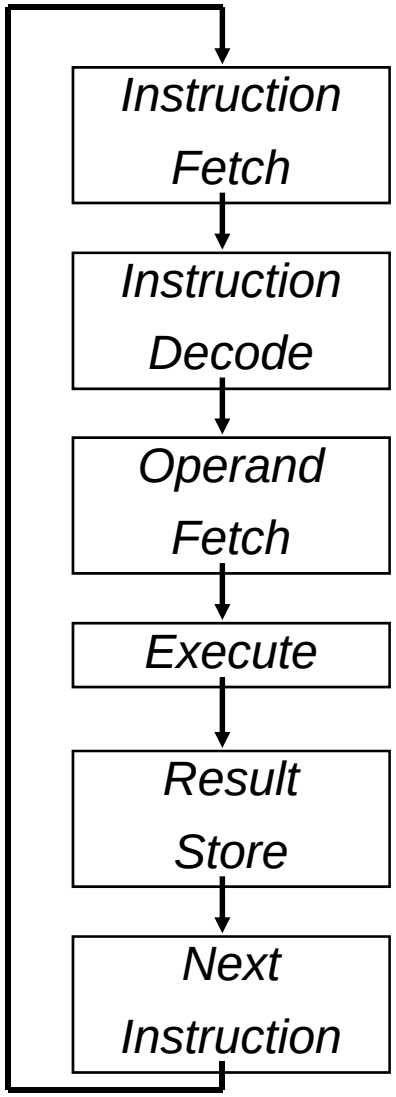
- State machine in which part of state is a "micro-pc".
  - **Explicit circuitry for incrementing or changing PC**
- Includes a ROM with "microinstructions".
  - Controlled logic implements at least branches and jumps



	Instruction	Branch
0:	forw 35	xxx
1:	b_no_obstacles	000
2:	back 10	xxx
3:	rotate 90	xxx
4:	goto	001



Instruction Execution Cycle



Obtain instruction from program storage

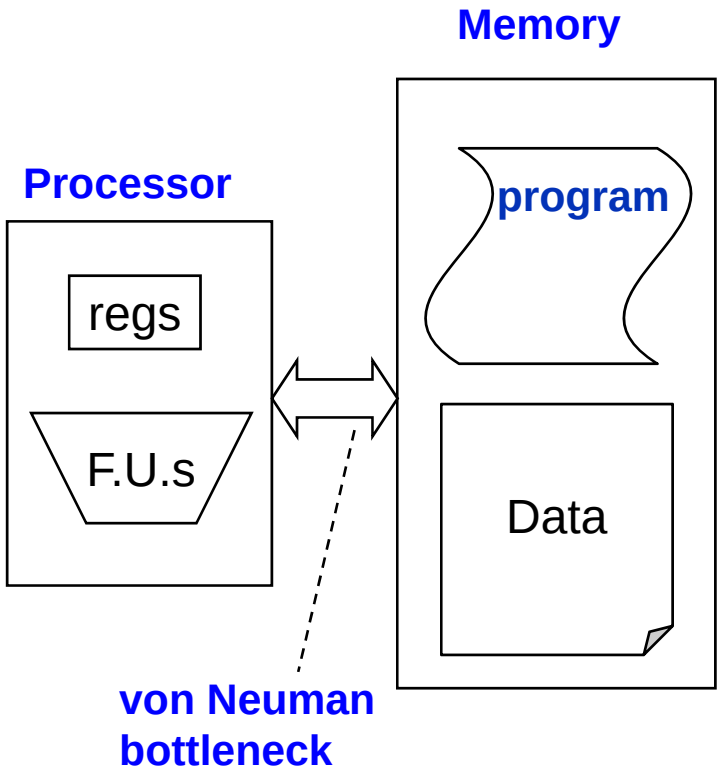
Determine required actions and instruction size

Locate and obtain operand data

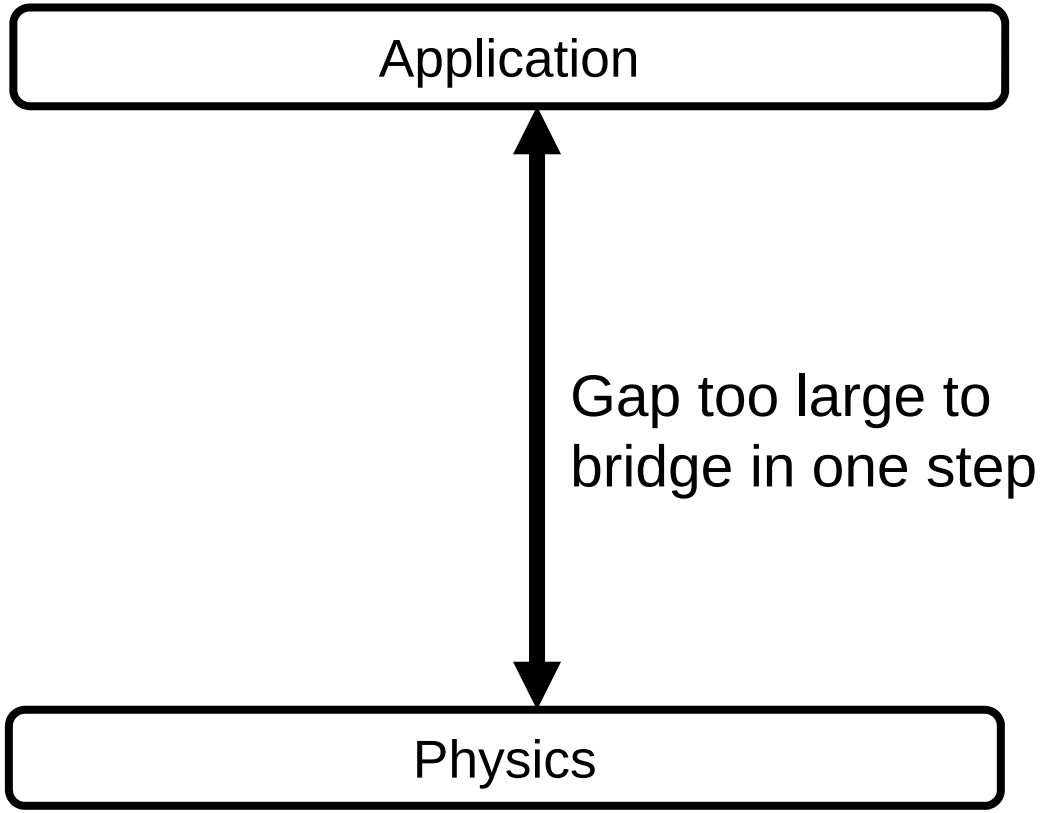
Compute result value or status

Deposit results in storage for later use

Determine successor instruction

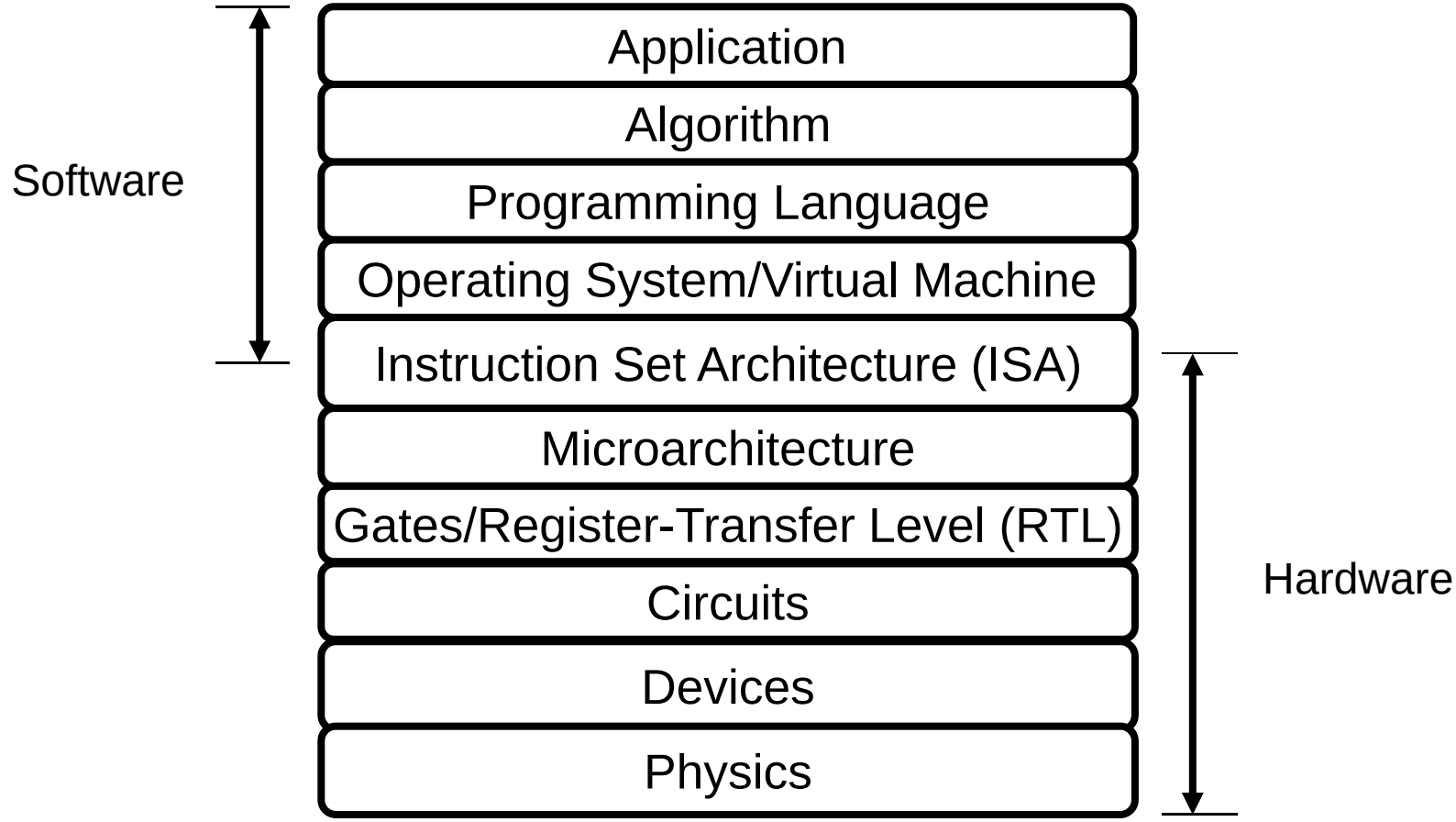


Computer Architecture?

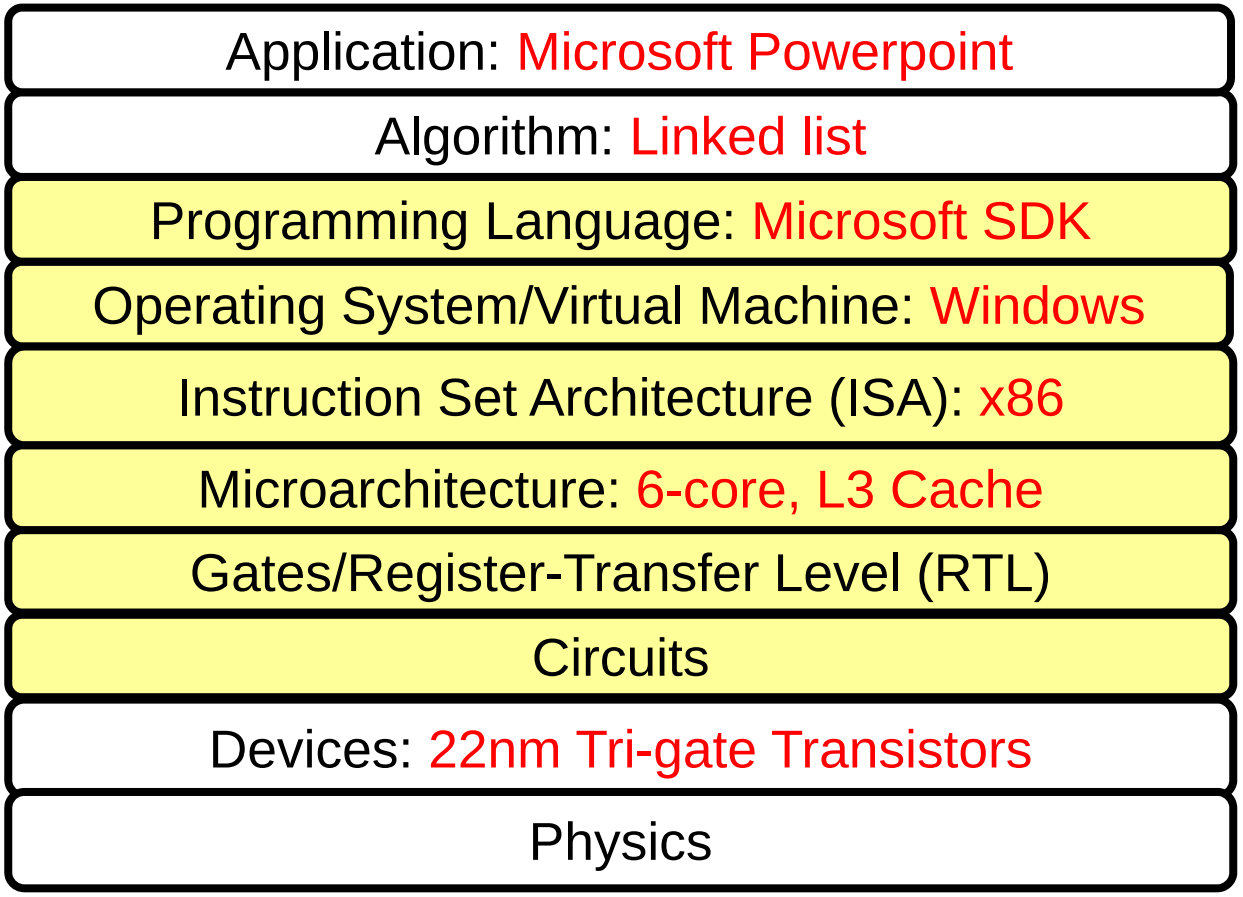


In its general definition, computer architecture is the *design of the abstraction layers* that allow us to implement information processing applications efficiently using available manufacturing technologies.

Abstraction Layers in Modern Systems



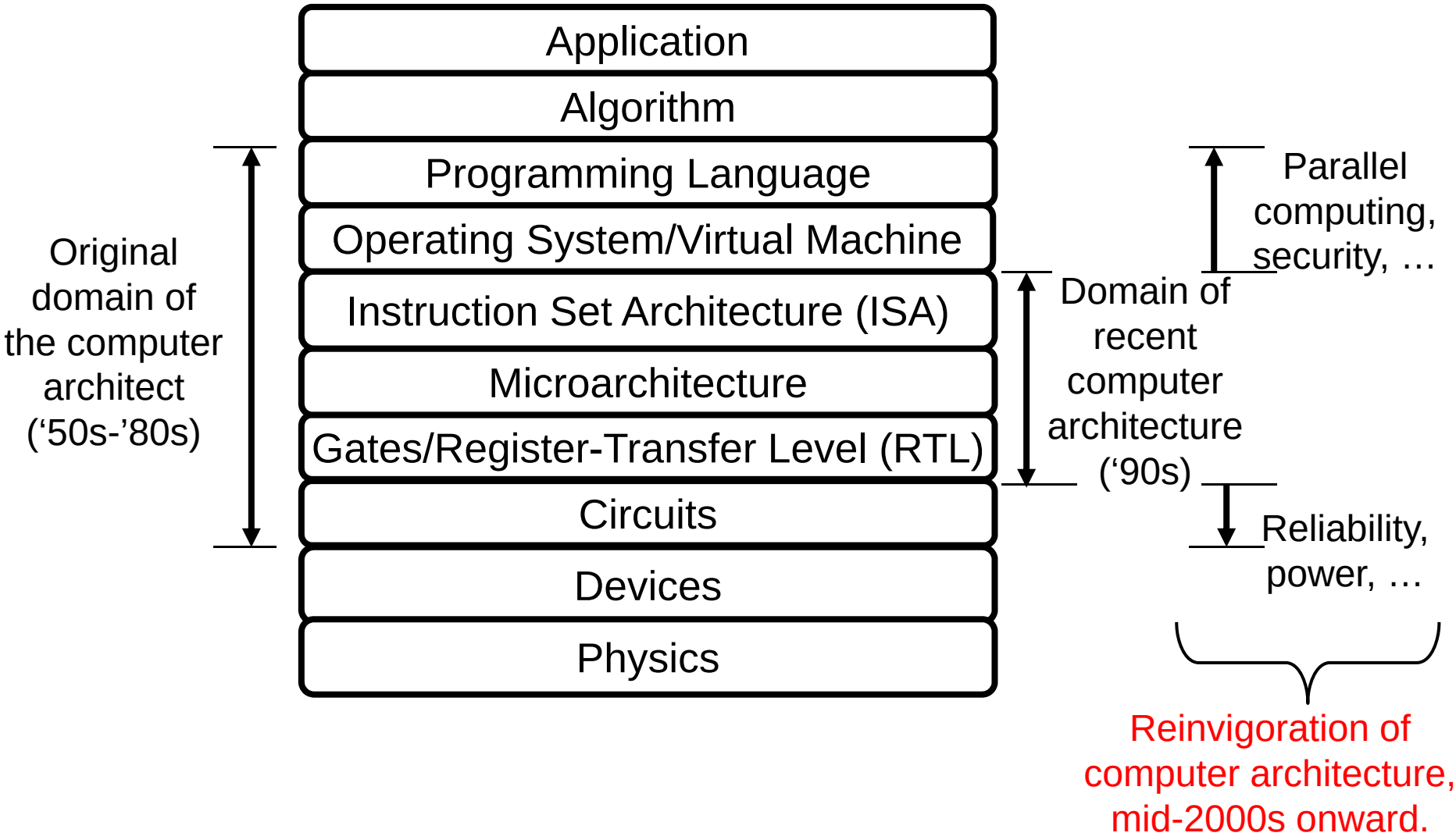
Example: PC


Application: Microsoft Powerpoint
Algorithm: Linked list
Programming Language: Microsoft SDK
Operating System/Virtual Machine: Windows
Instruction Set Architecture (ISA): x86
Microarchitecture: 6-core, L3 Cache
Gates/Register-Transfer Level (RTL)
Circuits
Devices: 22nm Tri-gate Transistors
Physics

Example: Smart Phone

Application: Angry Birds
Algorithm: Trajectory
Programming Language: Android SDK
Operating System/Virtual Machine: Android/Dalvik
Instruction Set Architecture (ISA): ARM
Microarchitecture: Quad-core, L2 Cache
Gates/Register-Transfer Level (RTL)
Circuits
Devices: 32nm CMOS
Physics

Computer Architecture Course

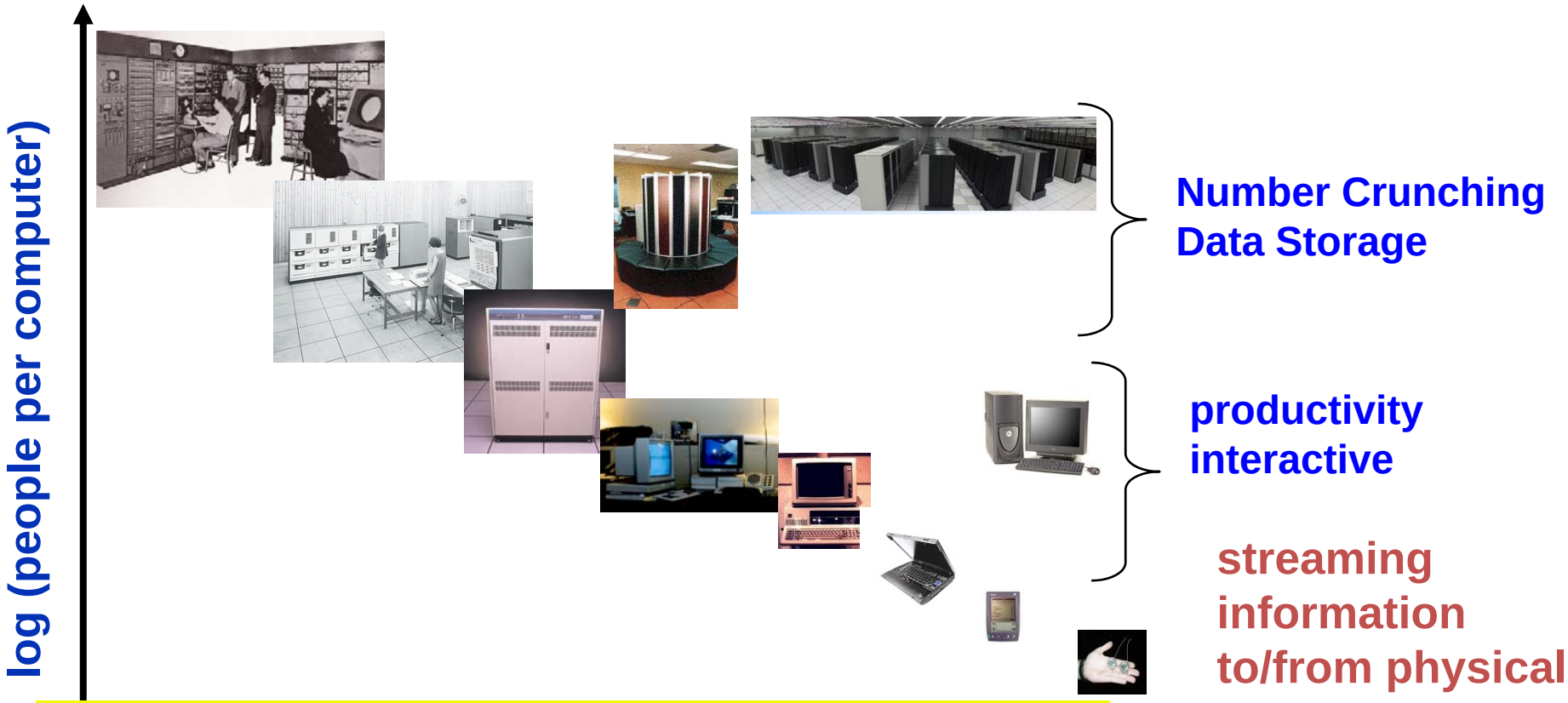


Concluding Remarks

- The abstraction layers provide
  - Environmental stability within generation
  - Environmental stability across generation
  - Consistency across a large number of units
- Modern computer architecture cannot avoid paying attention to software and compilation issues.

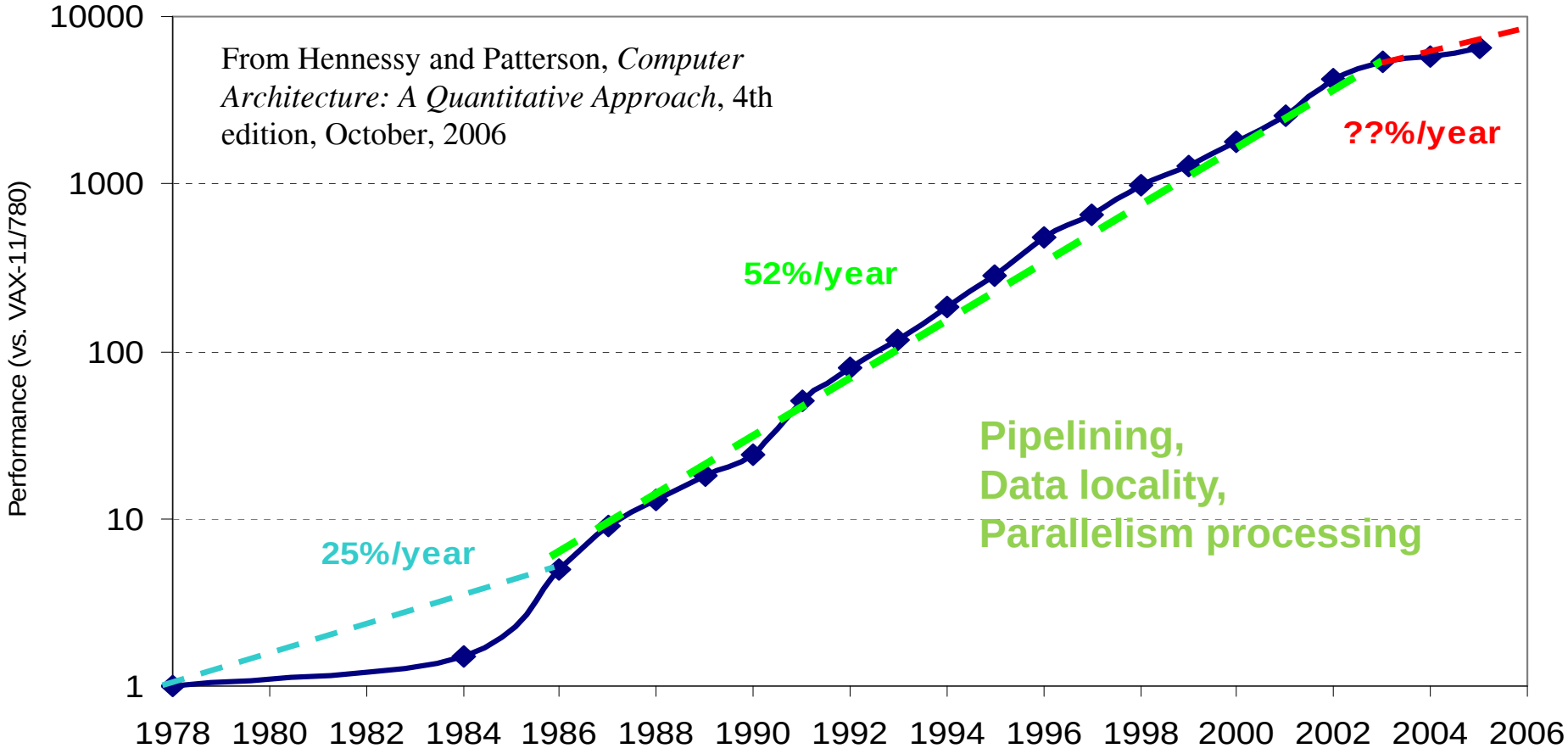
- Computer architecture is engineering design under constraints
  - Average case & worst case, peak power & energy per operation, soft errors & hard errors, hardware cost & software cost, ….

"Bell's Law" – new class per decade



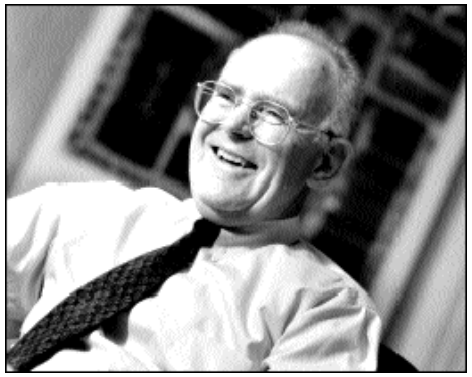
- Enabled by technological opportunities
- Smaller, more numerous and more intimately connected
- Brings in a new kind of application
- Used in many ways not previously imagined

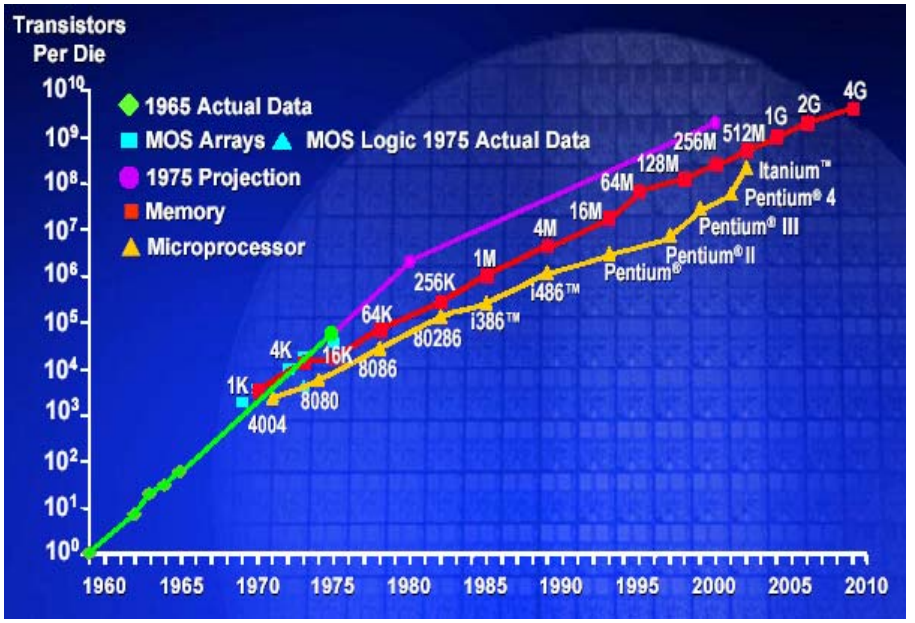
Uniprocessor Performance



- **VAX : 25%/year 1978 to 1986**
- **RISC + x86: 52%/year 1986 to 2002**
- **RISC + x86: ??%/year 2002 to present**

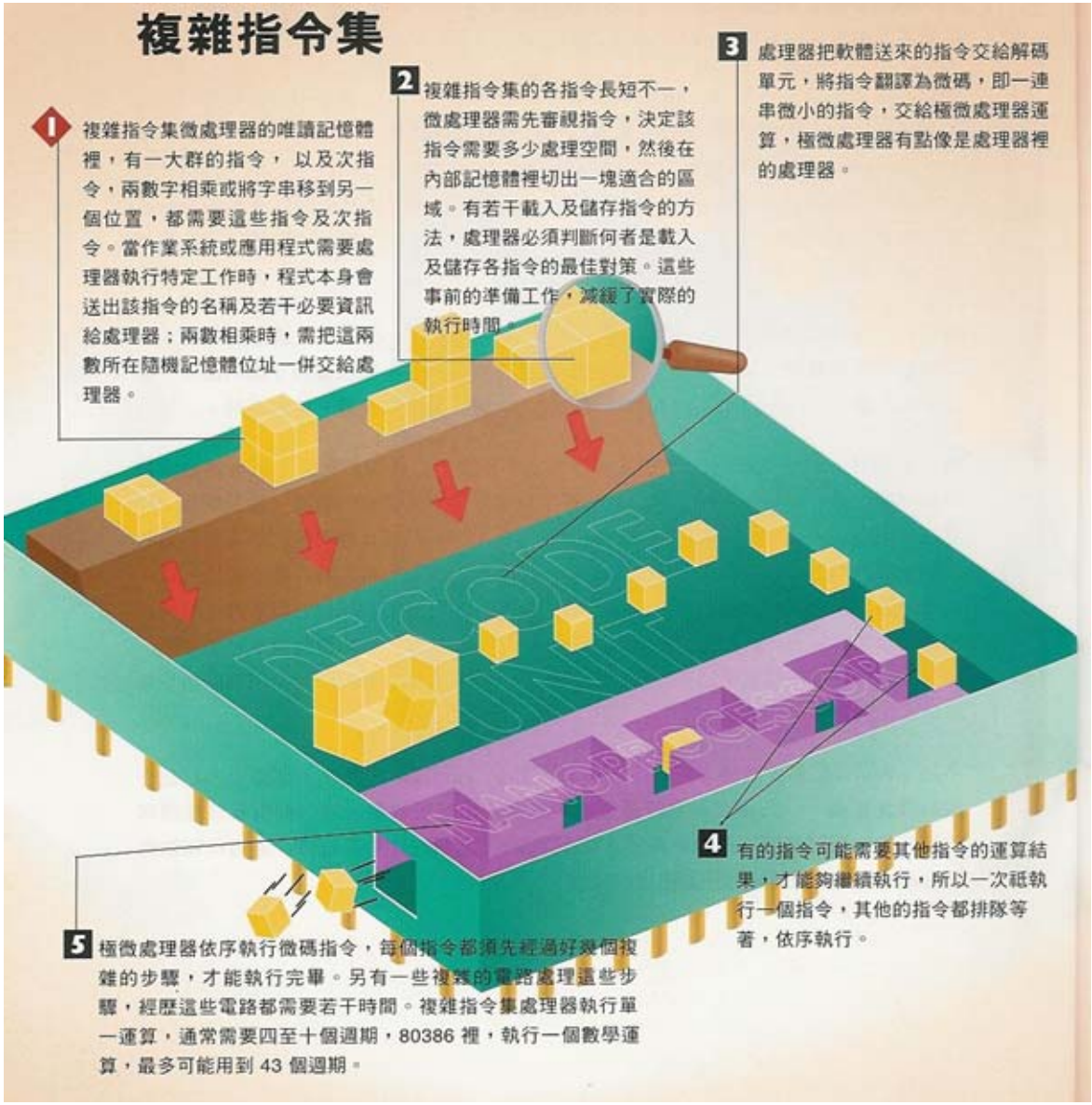
Driven Technology: Moore's Law



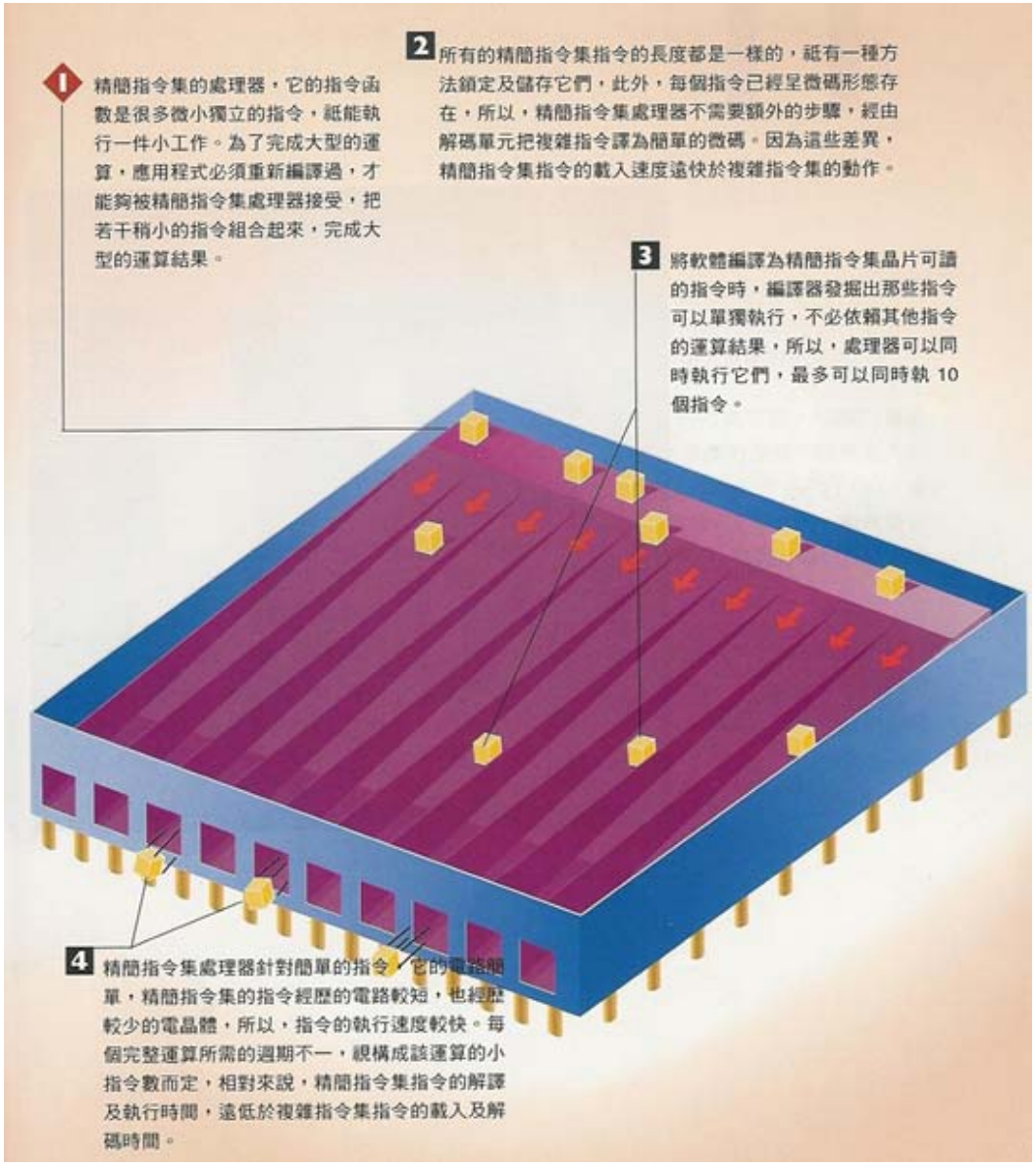


- "Cramming More Components onto Integrated Circuits"
  - Gordon Moore, Electronics, 1965
- # of transistors on cost-effective integrated circuit double every 18 months

CISC (Complex Instruction Set Computer)



RISC (Reduced Instruction Set Computer)

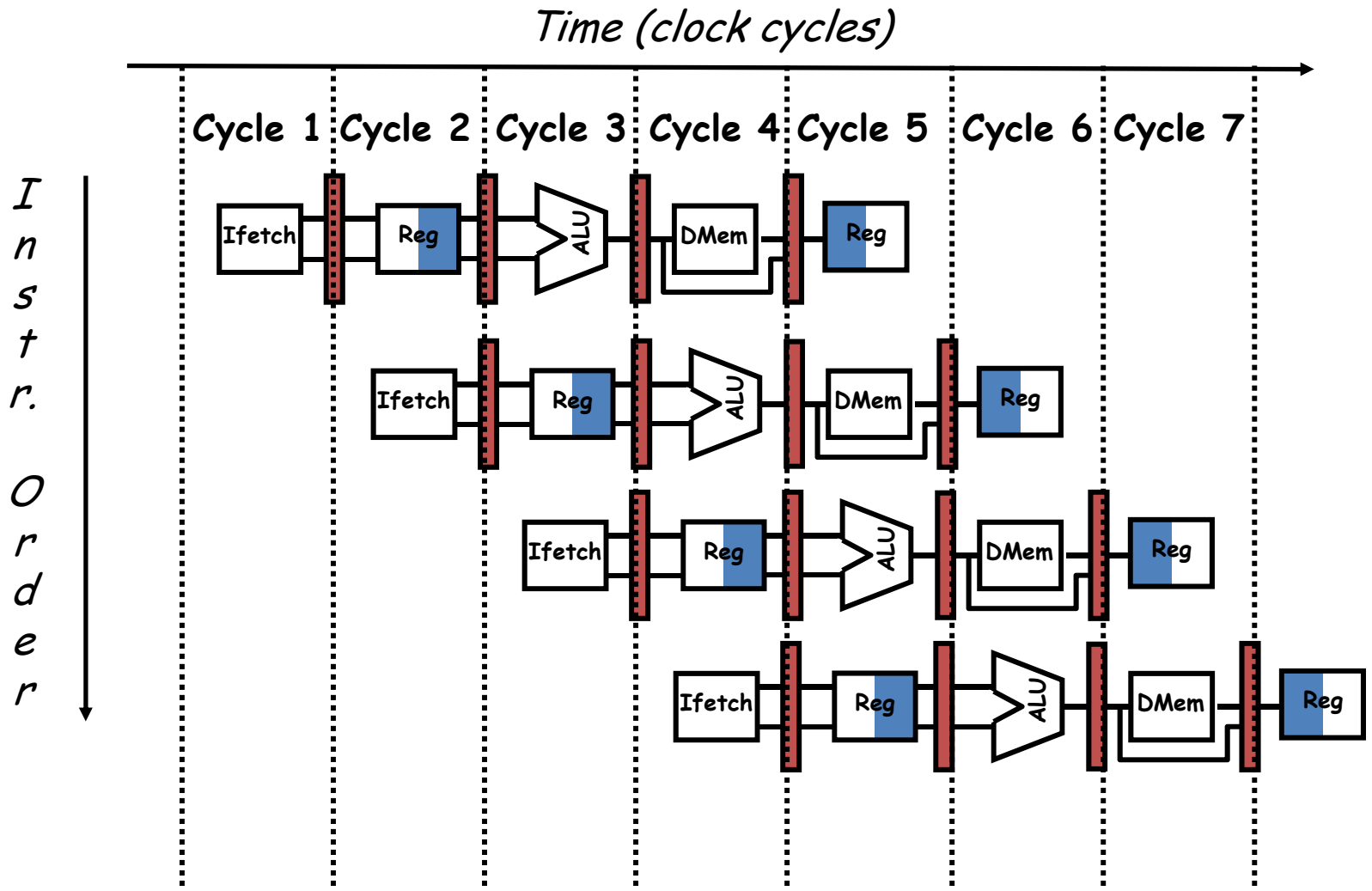


UniProcessor Performance

- Early 1970s, mainframes and minicomputers
  - 25%~30% growth per year in performance
- Late 1970, microprocessor
  - 35% growth per year in performance
- Early 1980s, Reduced Instruction Set Computer (RISC) architectures
  - 2 critical performance techniques
    - ILP (initially through pipelining and later through multiple instruction issue)
    - Cache
  - 50% growth per year in performance
- 1998~2000, relative performance
  - By technology: 1.35×per year
  - By technology + architecture, overall: 1.58 ×per year

  Note: 1.58≈1.35×(1+15%), the architecture improvement factor is 15%

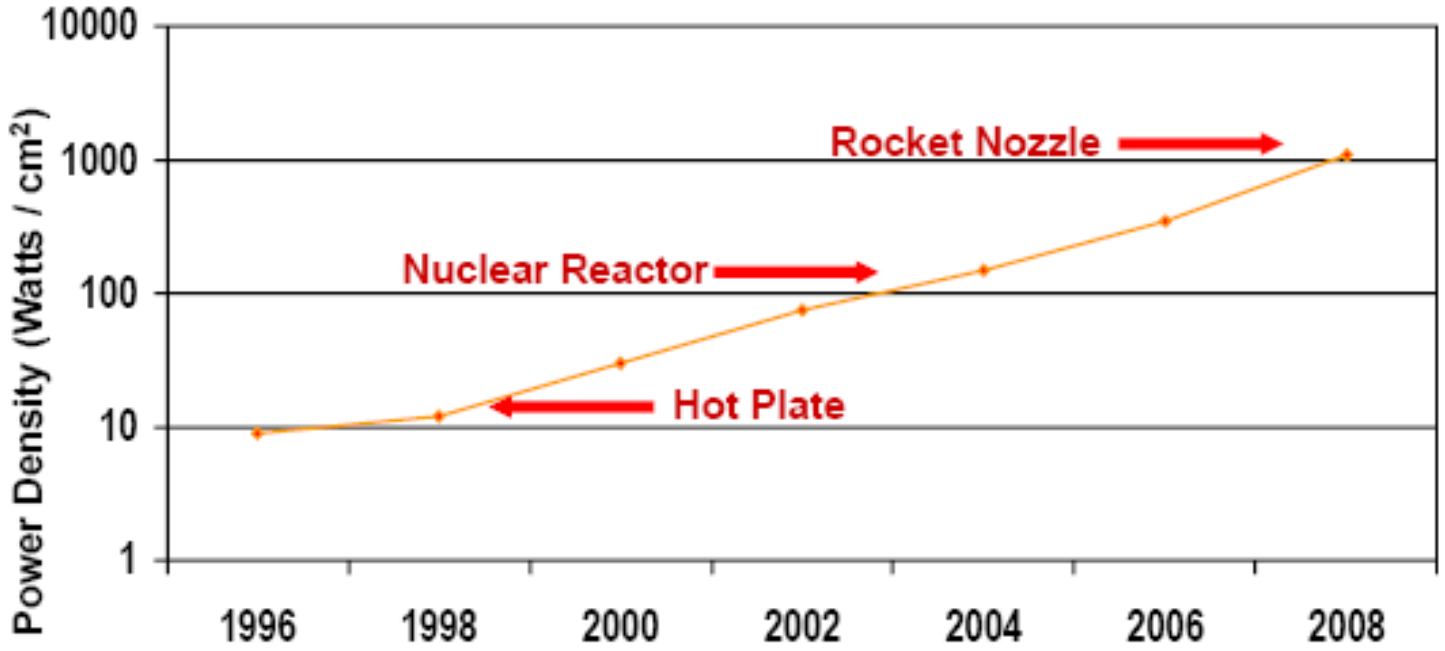
Pipelined Instruction Execution



Limiting Force: Power Density

Moore's Law Extrapolation:

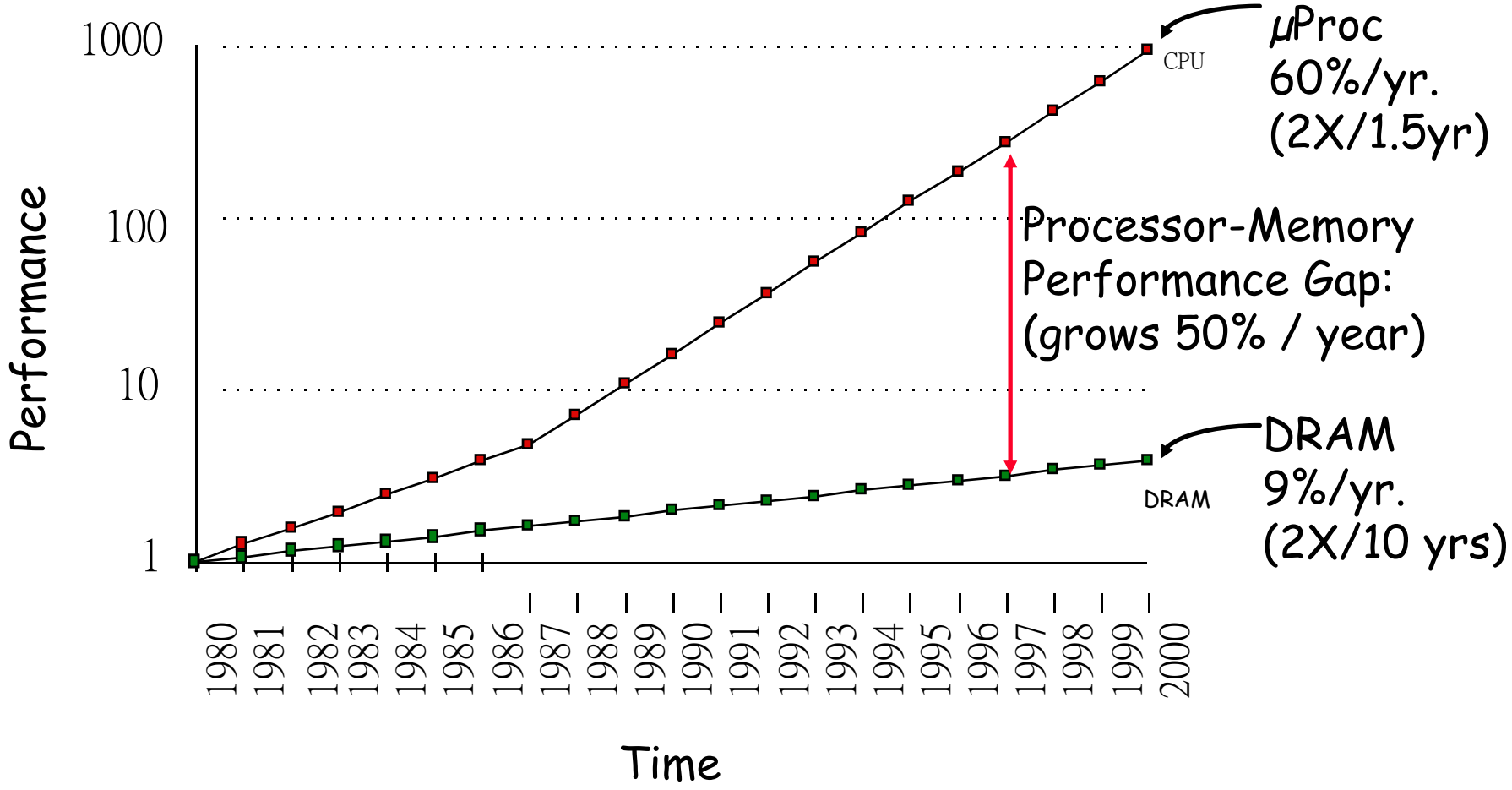
Power Density for Leading Edge Microprocessors



Power Density Becomes Too High to Cool Chips Inexpensively

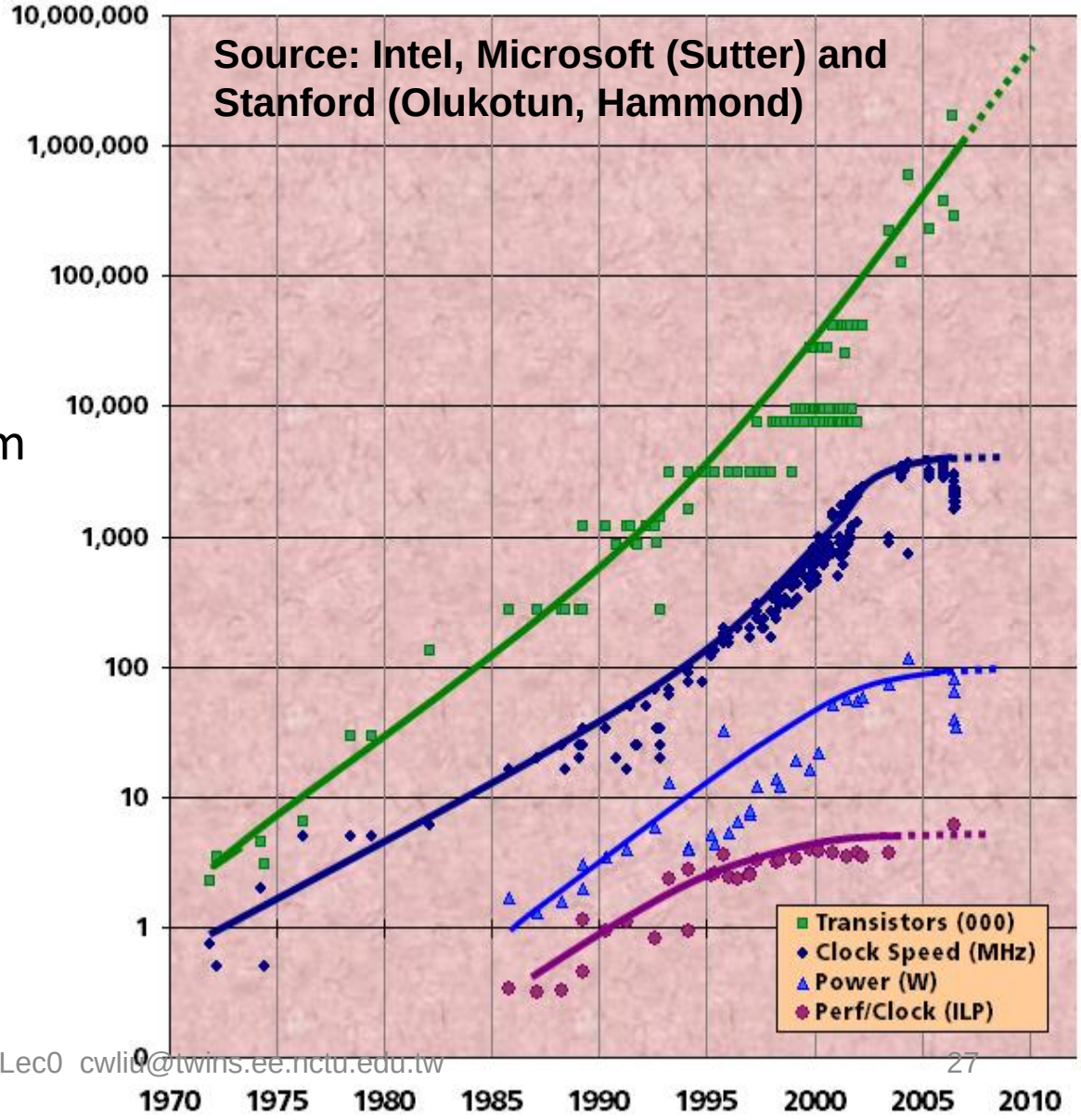
Source: Shekhar Borkar, Intel Corp

Limiting Force: Processor-DRAM Memory Gap (latency)



Limiting Force: Clock Speed and ILP

- Chip density is continuing increase ~2x every 2 years
  - Clock speed is not
  - # processors/chip (cores) may double instead
- There is little or no more Instruction Level Parallelism (ILP) to be found
  - Can no longer allow programmer to think in terms of a serial programming model
- Conclusion: Parallelism must be exposed to software!



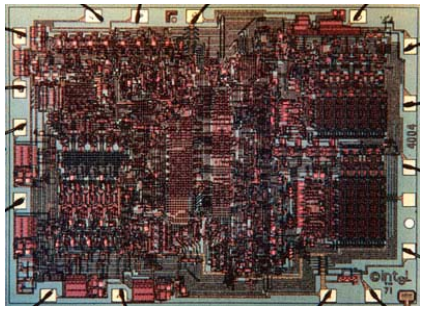
Crossroads in Computer Architectue

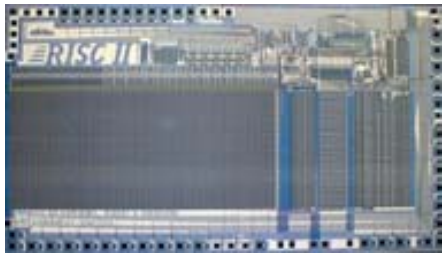
- Old Conventional Wisdom: Power is free, Transistors expensive
- New Conventional Wisdom: "Power wall" Power expensive, Xtors free (Can put more on chip than can afford to turn on)
- Old CW: Sufficiently increasing Instruction Level Parallelism via compilers, innovation (Out-of-order, speculation, VLIW, …)
- New CW: "ILP wall" law of diminishing returns on more HW for ILP
- Old CW: Multiplies are slow, Memory access is fast
- New CW: "Memory wall" Memory slow, multiplies fast (200 clock cycles to DRAM memory, 4 clocks for multiply)
- Old CW: Uniprocessor performance 2X / 1.5 yrs
- New CW: Power Wall + ILP Wall + Memory Wall = Brick Wall
  - Uniprocessor performance now 2X / 5(?) yrs

⇒ Sea change in chip design: multiple "cores" (2X processors per chip / ~ 2 years)

- More power efficient to use a large number of simpler processors rather than a small number of complex processors

Sea Change in Chip Design



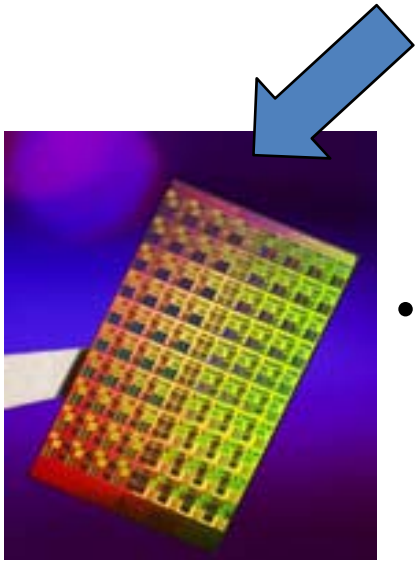


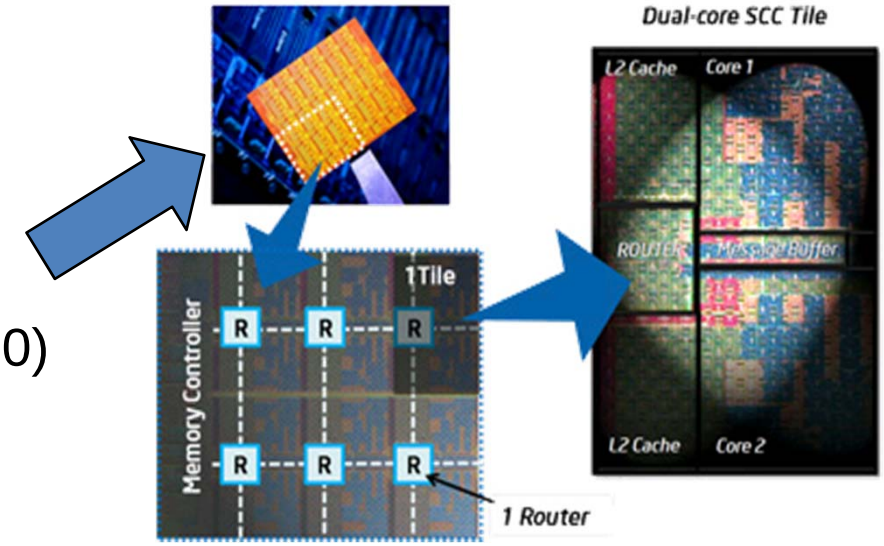
- Intel 4004 (1971):
  - 4-bit processor,
  - 2312 transistors, 0.4 MHz,
  - 10 μm PMOS, 11 mm^2 chip
- RISC II (1983):
  - 32-bit, 5 stage
  - pipeline, 40,760 transistors, 3 MHz,
  - 3 μm NMOS, 60 mm^2 chip
- 125 mm^2 chip, 65 nm CMOS
  = 2312 RISC II+FPU+Icache+Dcache
  - RISC II shrinks to ~ 0.02 mm^2 at 65 nm
  - Caches via DRAM or 1 transistor SRAM (www.t-ram.com) ?
  - Proximity Communication via capacitive coupling at > 1 TB/s ? (Ivan Sutherland @ Sun / Berkeley)

- Processor is the new transistor?

ManyCore Chips: The trend is here

- Intel 80-core multicore chip (Feb 2007)
  - 80 simple cores
  - Two FP-engines / core
  - Mesh-like network
  - 100 million transistors
  - 65nm feature size
- Intel Single-Chip Cloud Computer (August 2010)
  - 24 "tiles" with two IA cores per tile
  - 24-router mesh network with 256 GB/s bisection
  - 4 integrated DDR3 memory controllers
  - Hardware support for message-passing





- "ManyCore" refers to many processors/chip
  - 64? 128? Hard to say exact boundary
- How to program these?
  - Use 2 CPUs for video/audio
  - Use 1 for word processor, 1 for browser
  - 76 for virus checking???
- Something new is clearly needed here…

Problems with Sea Change

- Algorithms, programming languages, compilers, operating systems, architectures, libraries, … not ready for 1000 CPUs / chip
  - Thread-level parallelism (TLP)
  - Data-level parallelism (DLP)

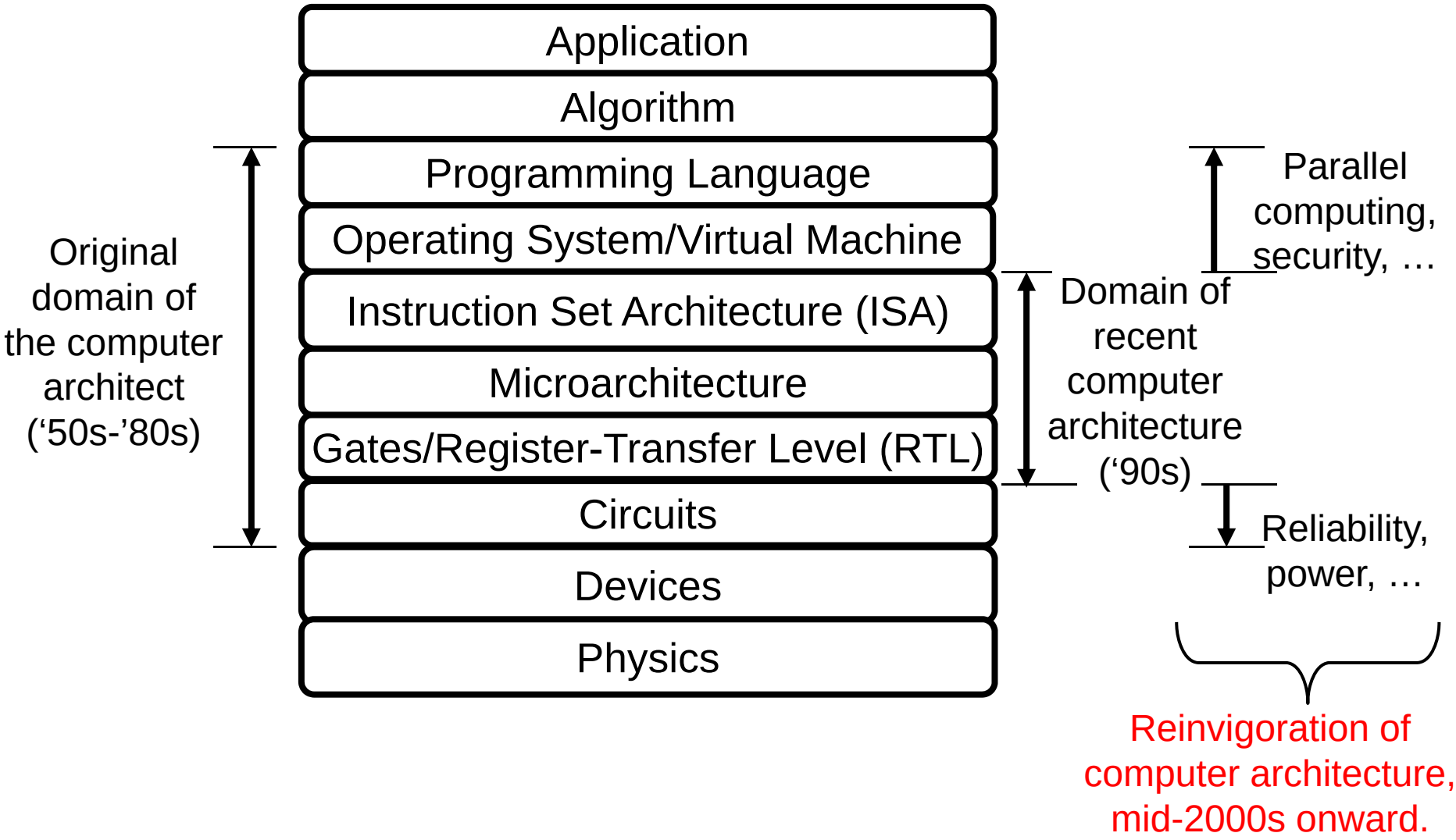
- Four architectures exploit DLP and TLP
  - Instruction-level parallelism – chapter 3
  - Vector architectures and GPUs – chapter 4
  - Thread-level parallelism – chapter 5
  - Request-level parallelism – chapter 6

DEPT. OF ELECTRONICS ENGINEERING & INST. OF ELECTRONICS NCTU

Course Information

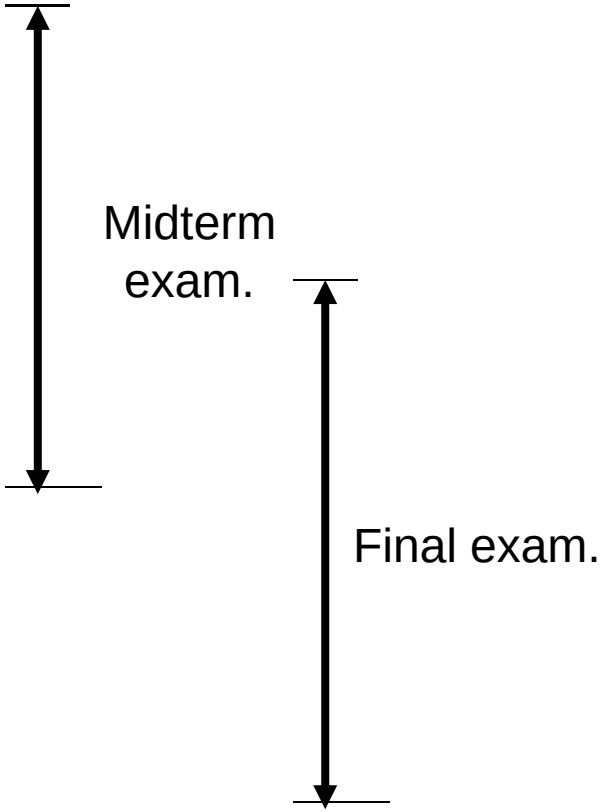
- Lecture:
  - Chih-Wei Liu 劉志尉 cwliu@twins.ee.nctu.edu.tw
  - 5731685, ED618
- Teaching Assistants:
  - 楊承彥, 張鈞豪
  - 54225, ED412
- Course website: http://twins.ee.nctu.edu.tw
- Prerequisites:
  - Computer organization
  - Computer programming

Computer Architecture Course



This Course Has 5 Modules

- Module 1
  - Instruction set architecture (ISA)
  - Pipelining and Hazards
- Module 2
  - Memory hierarchy
  - Caches and virtual memory
- Module 3
  - Instruction-level Parallelism
  - Branch prediction
  - Speculation and Reorder buffer
- Module 4
  - Thread-level Parallelism
  - Cache coherent protocols
- Module 5
  - Data-level Parallelism
  - Vector machines



Course Grade (tentative)

- Lectures, Homework, and Quizzes: 25%
  - Adapted from Prof. David Patterson's class notes
  - Please avoid arriving late or leaving early
  - One problem sets with respect to each chapter
  - One-page reading report with respect to each lecture
  - Homework should be handed in on time
- LAB & Project: 25%
- Midterm and Final Exams.: 50%
- (Extra points 5%)