Class Topics (클래스 홈페이지 참조)

- Part 1: Fundamental concepts and principles
  1) Invention of computers and digital logic design
  2) Abstractions to deal with complexity
  3) Data (versus code)
  4) Machines called computers
  5) Underlying technology and evolution since 1945
- Part 2: 빠른 컴퓨터를 위한 설계 (ISA design)
- Part 3: 빠른 컴퓨터를 위한 구현 (pipelining, cache)

Machines Called Computers

Part 5
Underlying Technologies and Evolution, More on Computers

References:

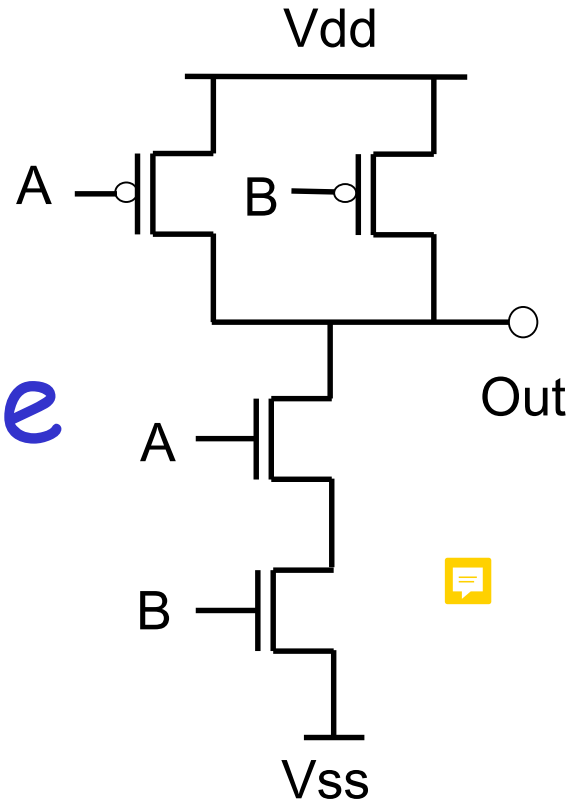
1. Computer Organization and Design & Computer Architecture, Hennessy and Patterson (slides are adapted from those by the authors)

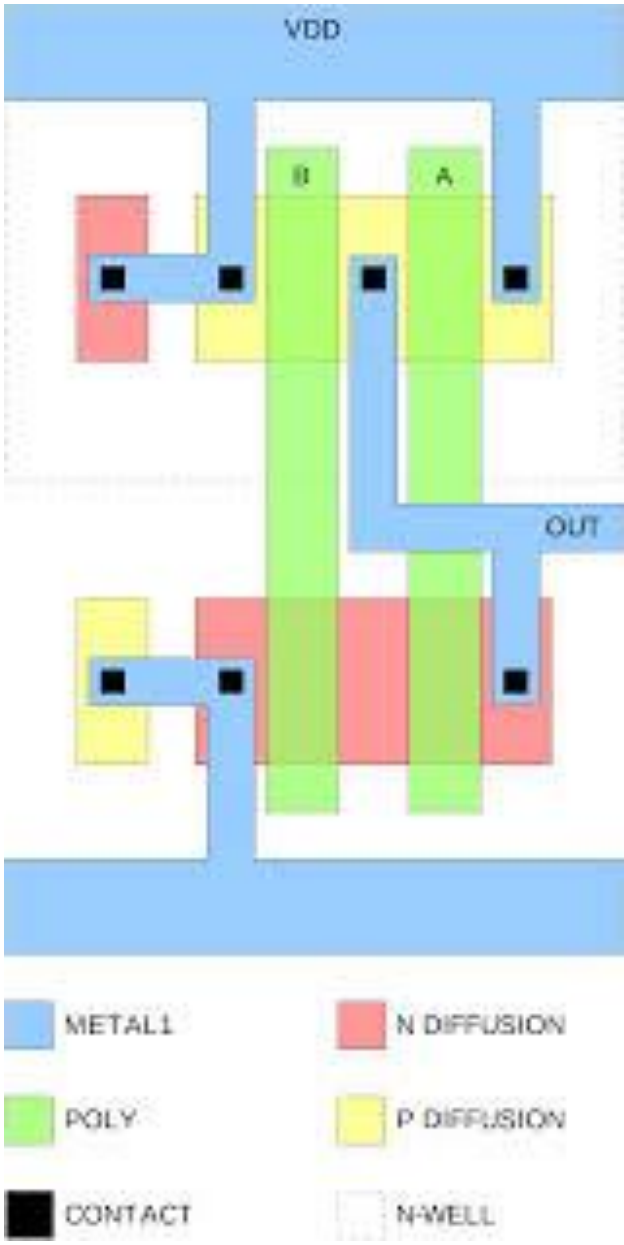
Semiconductor Technology
(컴퓨터라는 기계는 무엇으로 만드는가?)

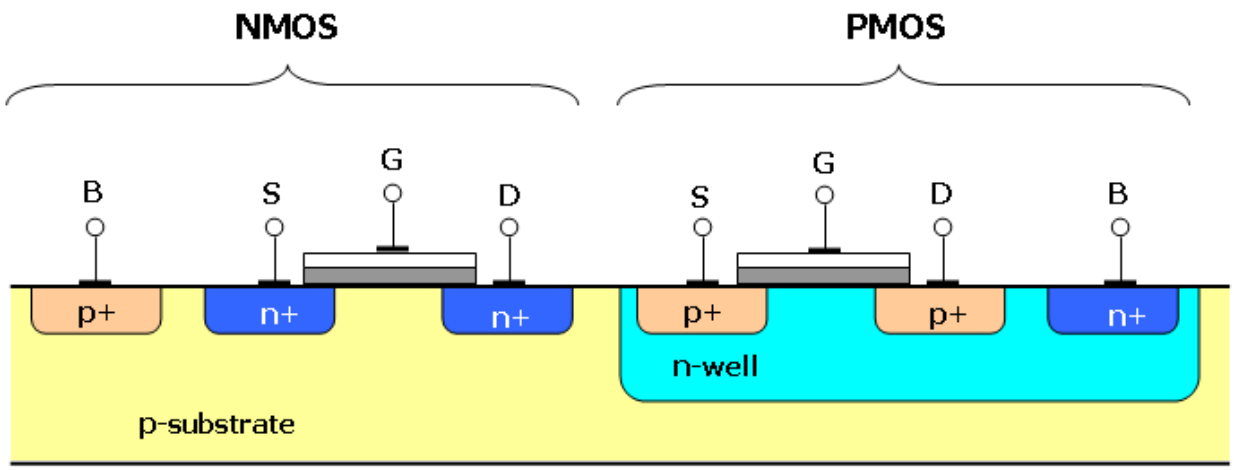
- Scaling의 개념과 잇점 그리고 결과적인 추세 이해하실 것
- 그리고 wafer, die, yield 라는 용어를 이해
- 숫자나 IC 제조공정은 암기대상이 아님

CMOS NAND Gate

(반복)





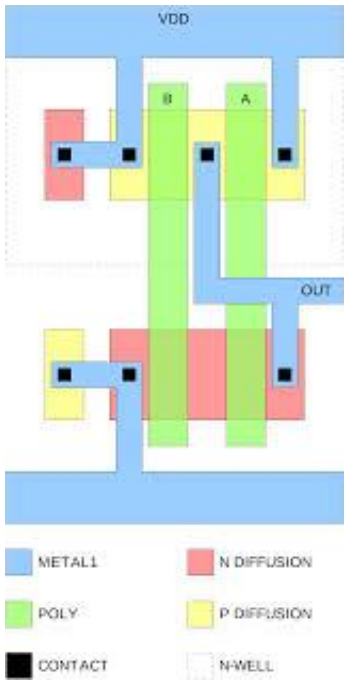


Semiconductor Technology

- Transistors invented in Bell labs. in 1947
  - Took 10 years to commercialize
- IC (integrated circuits) invented in 1958
  - Took 5 years to commercialize
  - SSI, MSI, LSI, VLSI
- Major driving force behind computer performance evolution
  - 실용적인 빠른 컴퓨터: 1970s 이후

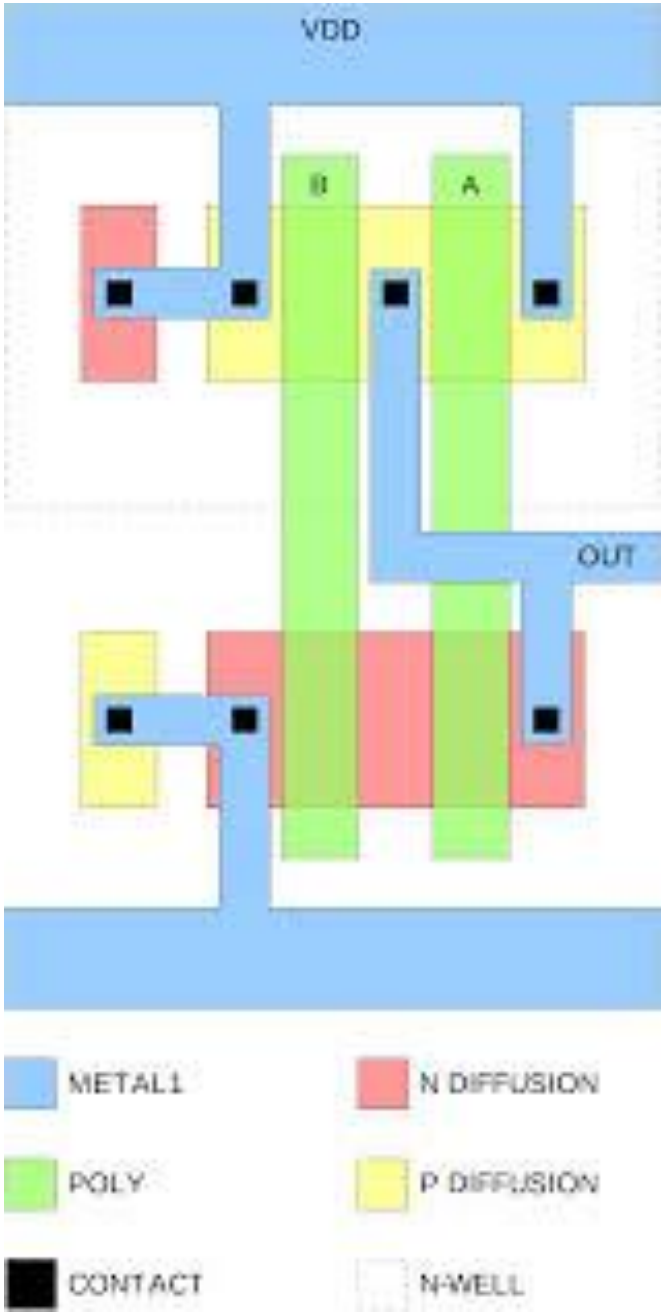
Smaller, Faster (반복)

- Minimum feature size (최소선폭)
  - 속도, 크기 (density)



10 nm

20 nm



Technology Scaling (Intel)

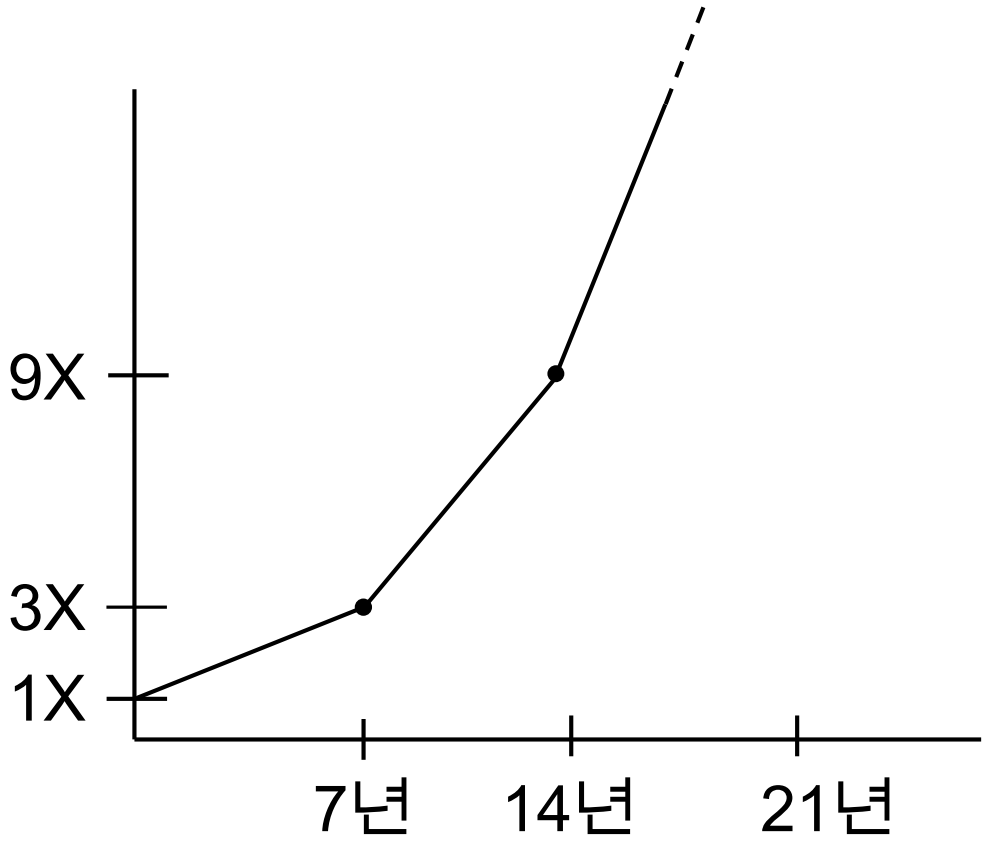
- Minimum feature size: exponential decrease

Min. Feature Size	year
10 μm	1971
3 μm	1977
1 μm	1984
350 nm	1993
130 nm	2001
45 nm	2007
14 nm	2014
5 nm	2020

https://en.wikipedia.org/wiki/14_nm_process

Technology Scaling

- Minimum feature size: exponential decrease
  - Speed and density: exponential increase



Technology Trends

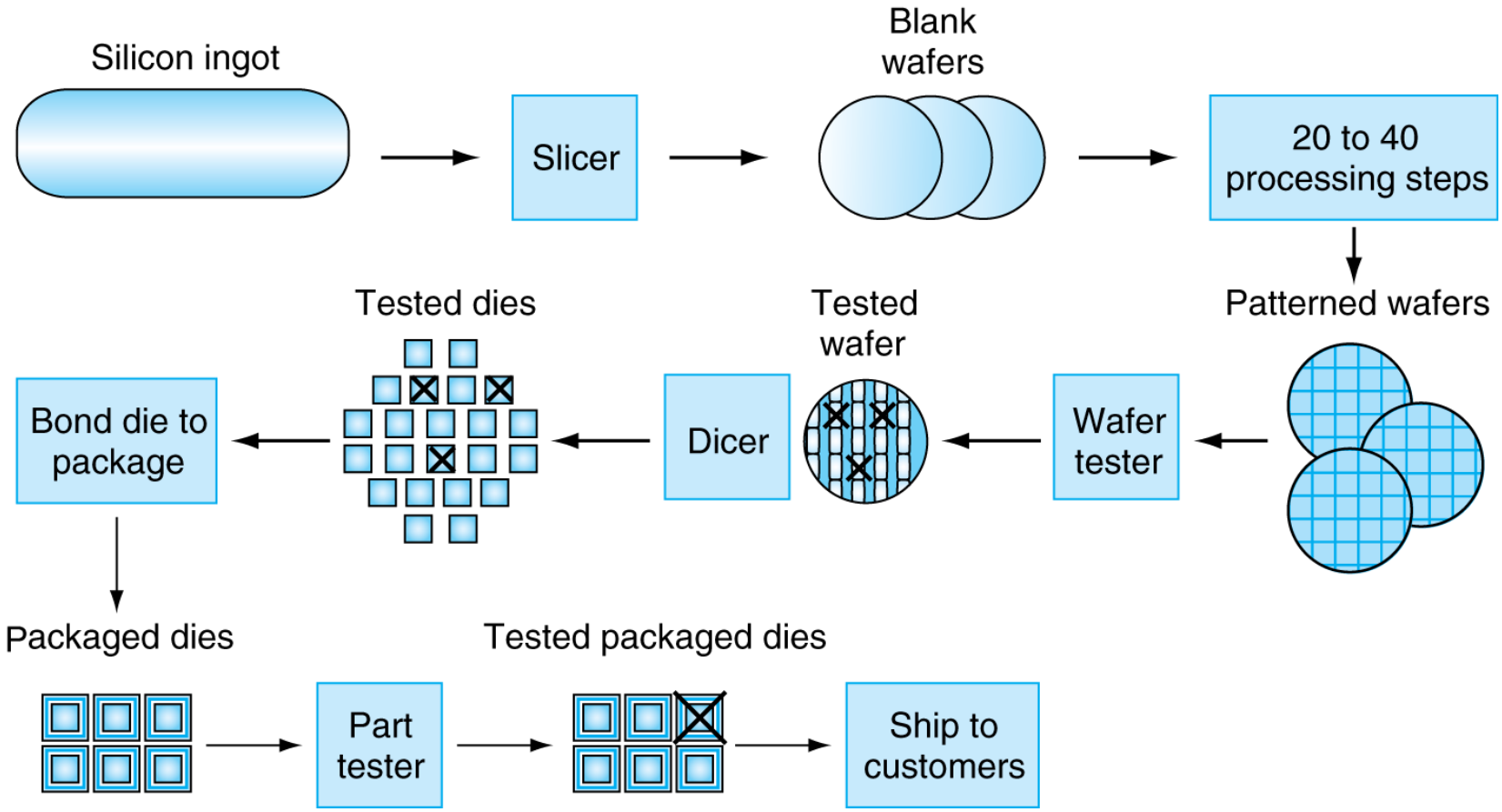
(Hennessy and Patterson slide, Computer Organization and Design, Morgan Kaufmann)

- 반도제 기술 (및 컴퓨터 설계 기술 - this class)

Year	Technology	Relative performance/cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2005	Ultra large scale IC	6,200,000,000

Manufacturing ICs (반복)

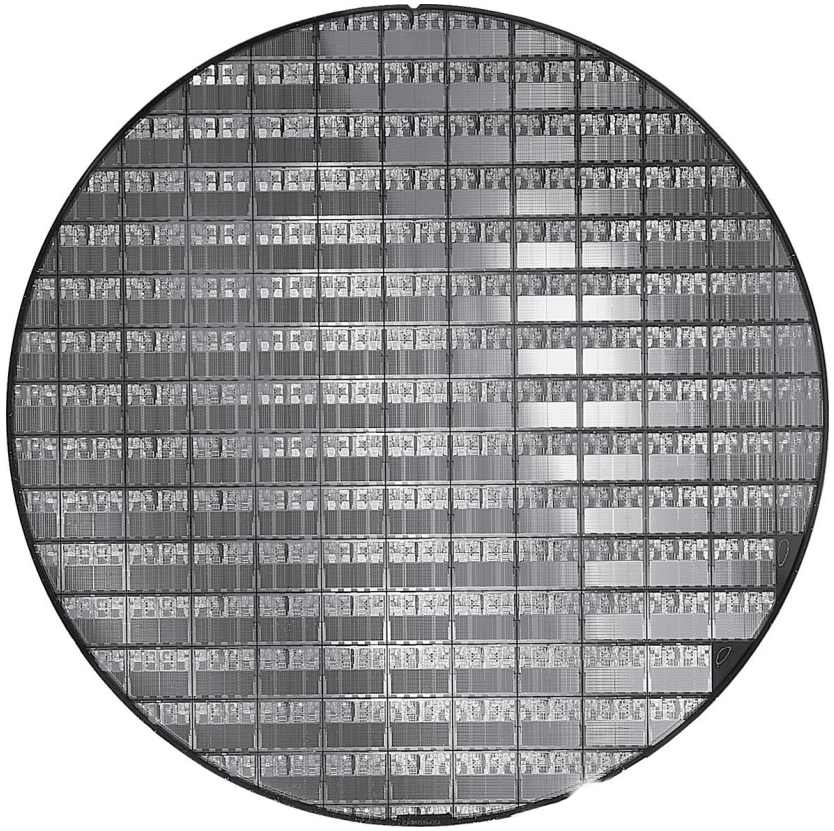
(Hennessy and Patterson slide, Computer Organization and Design, Morgan Kaufmann)



- Yield: proportion of working dies per wafer

AMD Opteron X2 Wafer

(Hennessy and Patterson slide, Computer Organization and Design, Morgan Kaufmann)

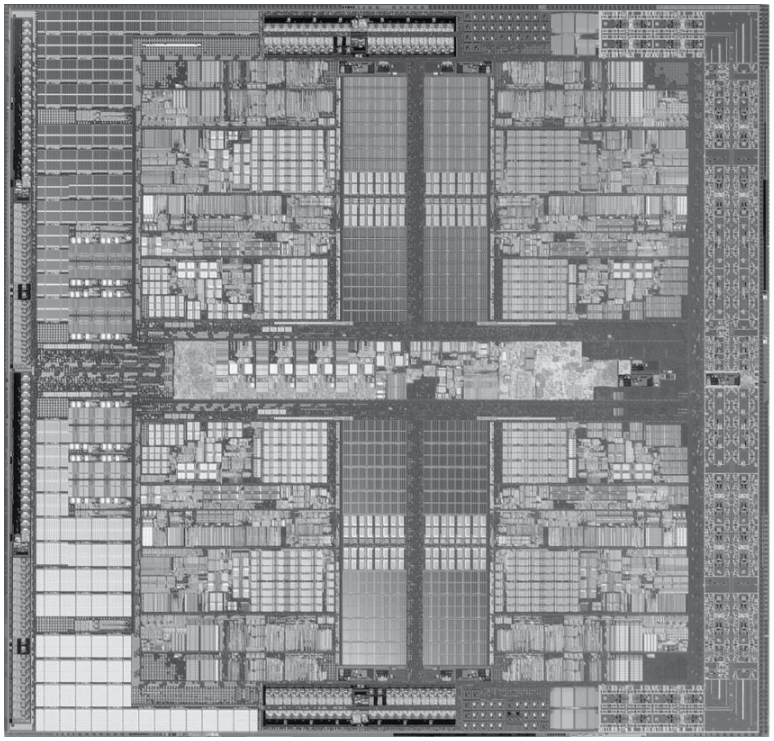


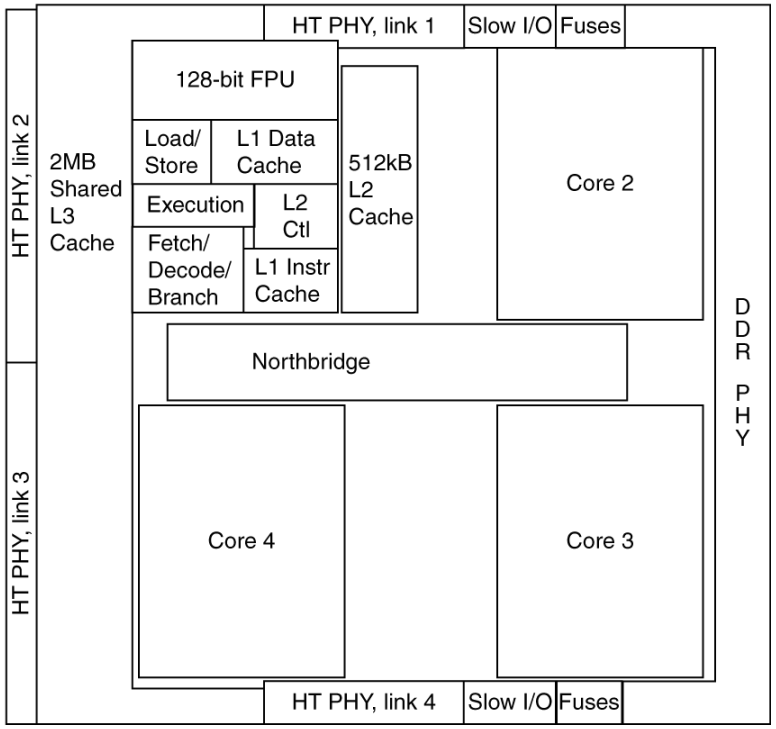
- X2: 300mm wafer, 117 chips, 90nm technology
- X4: 45nm technology

Inside the Processor

(Hennessy and Patterson slide, Computer Organization and Design, Morgan Kaufmann)

- AMD Opteron X4 (Barcelona): 4 processor cores





Semiconductor Technology

- What does it mean?
  - We have 5 nm technology
  - We process 300 mm wafers
- Moore's law: exponential growth
  - Number of transistors per chip double every 18 (or 12 or 24) months
  - Cost of fabrication facility increases exponentially over time

Processor Technology

- Microprocessor의 출현과 발전 흐름 이해
- 전체적인 흐름이 중요하고, 아주 세부적인 내용이나 숫자는 기억할 밀요 없음

Computer Architecture Technology

- Driving force behind computer performance evolution
  - Smaller transistors (semiconductor technology)
  - Increased die size to add more functionalities

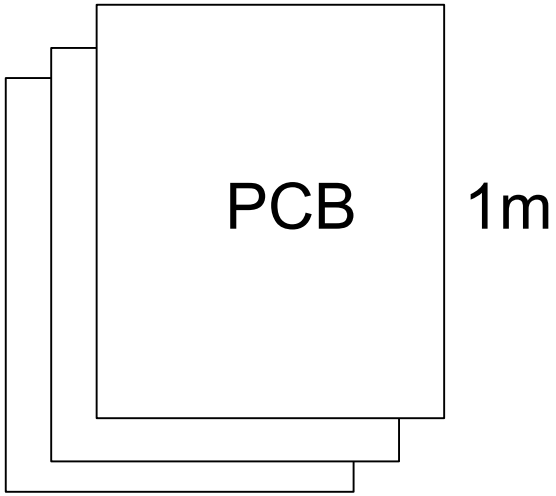
    (RISC ISA, pipelines, cache memory – this class)
- Processor perspective
  - Exponential growth in performance
- Around 2012
  - Highest transistor count in commercial processors
    - 2.5B in Intel's 10-core Xeon Westmere-EX (32nm)

Invention of Single-Chip Microprocessors

- Intel: 2차대전 후 Silicon Valley 의 반도체 회사 중 하나
  - Chip 주문 설계 및 생산
  - 컴퓨터는 미국 동부 (IBM)
- Breakthrough: 4004 microprocessor by Intel in 1971
  - Designed for Busicom to build calculators
  - Became a "computer" company
- 당시의 processors for mainframes
  - Only 4-bit microprocessor (with 2,300 transistors)
  - 크기, 가격, 성능 면에서 "micro"
- Intel announces 8-bit 8008 in 1972, 8080 in 1974

Processors in Mainframes in 1970s

- Started with 32-bit, but IC technology immature
  - Several large printed circuit boards
- Design cycle: 5 years (HW, OS and applications)



Single-Chip Microprocessors

- Altair: first microcomputer with 8080 (in 1975)
  - Along the style of best minicomputers
- Software and microcomputer startups
  - "모두의 책상에 컴퓨터를"
  - Microcomputers for business (word processing, excel)
- IBM 이 "IBM PC" 를 발표하고 PC 시장을 평정 (1981)
  - 제한된 형태로 참여
  - 16-bit 8088 and MS DOS (and no major applications)
- Applications: Intel/Microsoft platform 에서 돌아감

Single-Chip Microprocessors

- IBM 의 특허를 피하는 방법 고안됨
  - 누구나 IBM PC 제조할 수 있게 됨
  - From "IBM PC" to "PC"
- 이미 applications 은 Intel/MS platform 에서 돌아감
- Intel and Microsoft
  - Owner of PC platform ("Wintel")
  - 기술 개발
- Silicon Valley: 컴퓨터 산업의 주도권

Single-Chip Microprocessors

- From IBM PC to PC (beginning of Intel and MS era)
  - Intel microprocessors
    - Single chip, short design cycles
    - Take full advantage of semiconductor technology

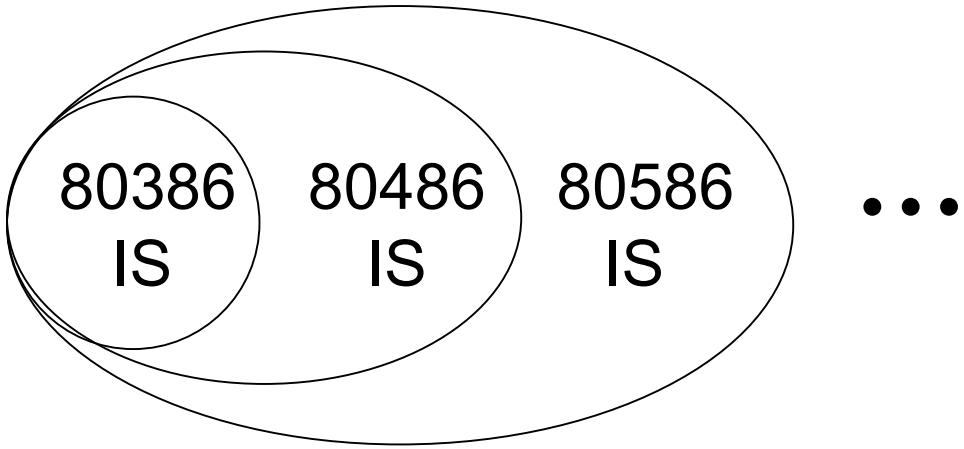
No. of Transistors (Exponential)

Processor	#Transistors	Year	Process (µm)	Area (mm^2)	
Intel 4004	2,300	1971	10	12	First µp
Intel 8080	4,500	1974	6	20	
Intel 8088	29,000	1979	3	33	IBM PC, 16 bit
Intel 80286	134,000	1982	1.5	49	PC/AT
Intel 80386	275,000	1985	1.5	104	x86, IA-32
Intel 80486	1,180,000	1989	1	160	
Pentium	3,100,000	1993	0.8	294	
Pentium II	7,500,000	1997	0.35	195	
Pentium 4	42,000,000	2000	0.18		
Itanium 2	220,000,000	2003	0.13		IA-64, RISC
Core i7 (Quad)	731,000,000	2008	0.045	263	x86-64
10-core Xeon Westmere-EX	2,600,000,000	2011	0.032	512	x86

https://en.wikipedia.org/wiki/Transistor_count

Single-Chip Microprocessors

- x86 ISA compatibility (and golden shackle)

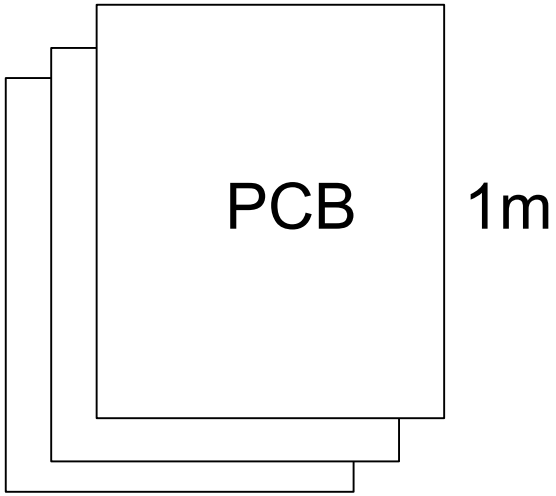


- From IBM PC to PC (beginning of Intel and MS era)
  - Intel microprocessors: single chip, short design cycles
    - Take full advantage of semiconductor technology

Processors in Mainframes in 1970s

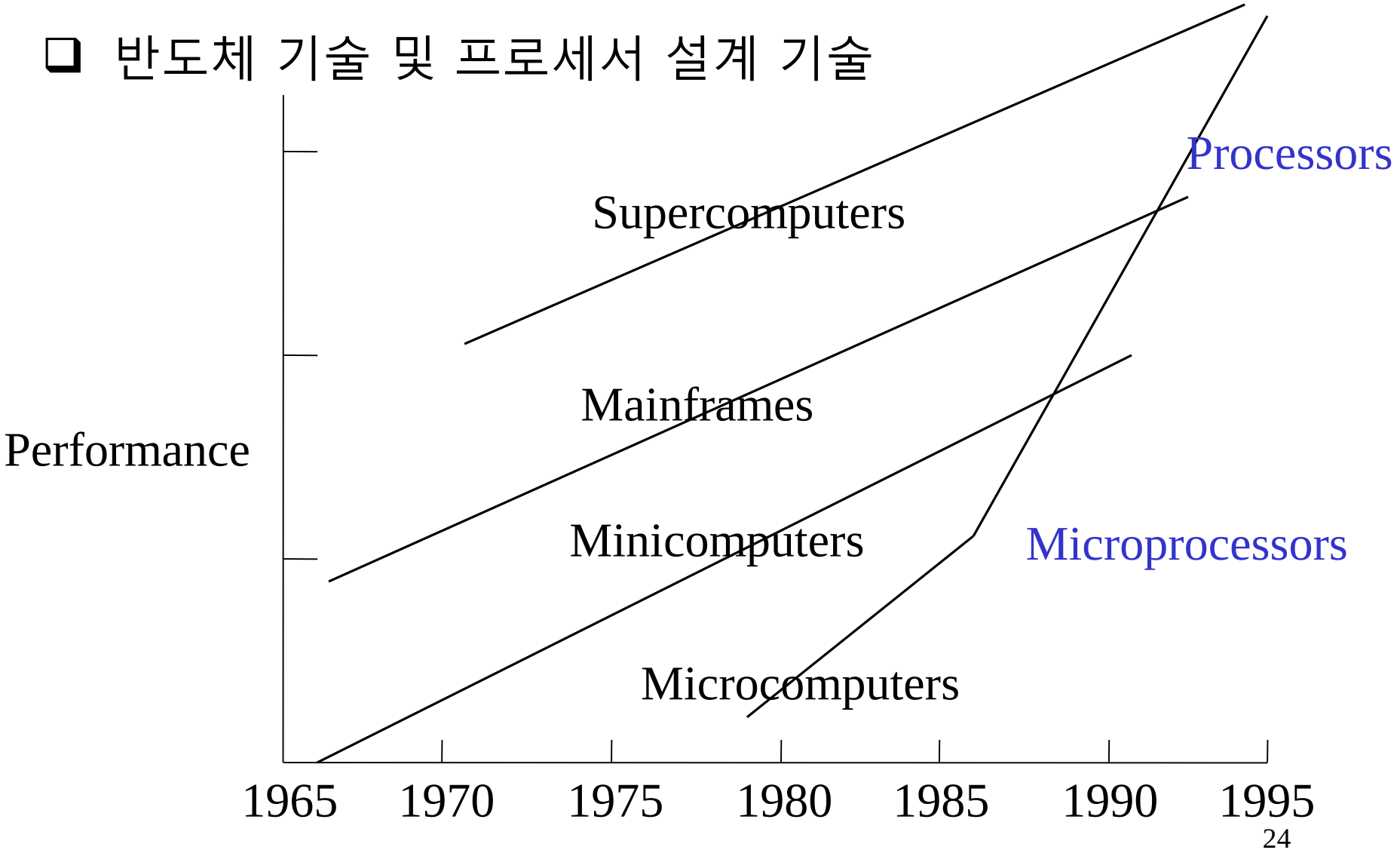
(반복)

- Started with 32-bit, but IC technology immature
  - Several large printed circuit boards
- Design cycle: 5 years (HW, OS and applications)



(Approximate) Performance Trend

- 반도체 기술 및 프로세서 설계 기술



Single-Chip Processors

- Microprocessors became powerful processors
  - Transistor scaling plus improved design
- Computer companies
  - Start to buy processors from processor vendors
  - Massive job changes in computer companies in 1980s
    - From hardware to software (for example, 5:5 → 1:9)
  - Focus on systems, software, service

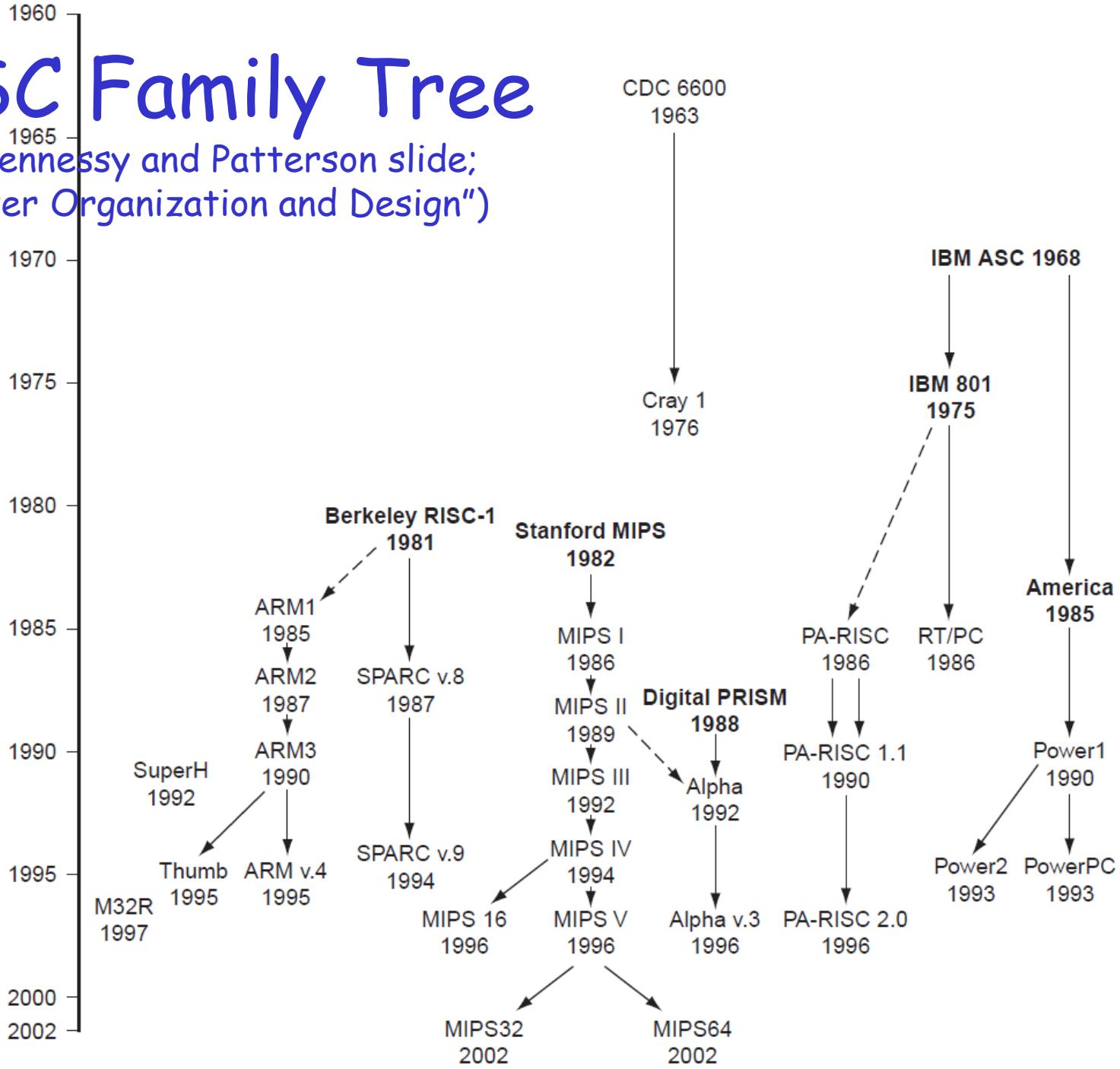
† Small microprocessors still there for low-cost embedded systems (many of them)

Instruction Set 변화 and RISC

- Processor design in 1970s: what we call CISC
  - Constraint: memory expensive
- 1980s: renaissance of processor design (RISC style)
  - Semiconductor technology
    - Memory became cheaper (move to RISC style)
  - Open Unix operating system
  - High-level programming
- Emergence of powerful 32-bit RISC processers
  - PowerPC, PA-RISC, MIPS, SPARC, Alpha, ARM
  - Exception is Intel x86 ISA

RISC Family Tree

(from Hennessy and Patterson slide;
"Computer Organization and Design")

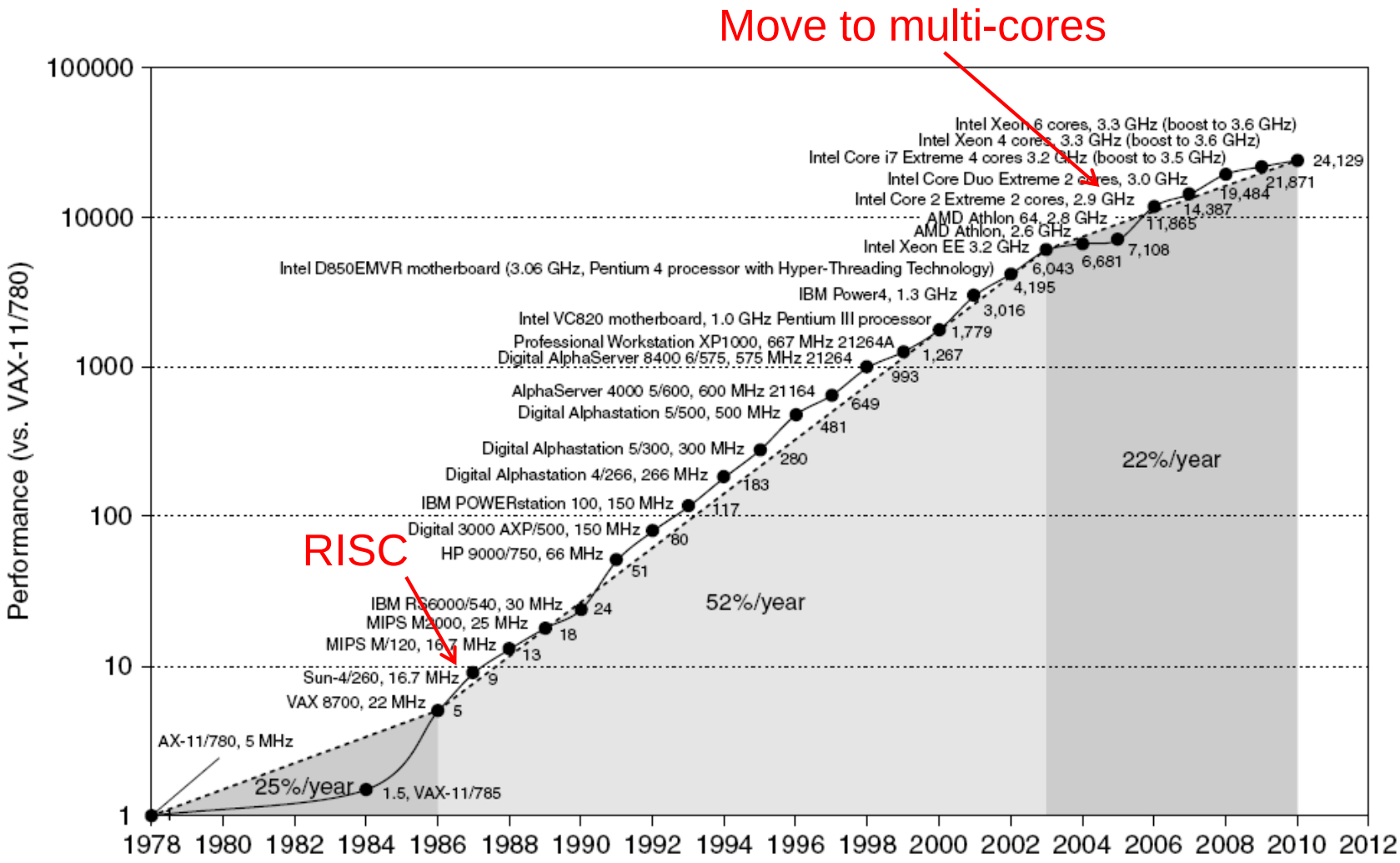


Trends in Technology (Exponential)

- Processor technology
  - Transistor density: 35%/year
  - Die size: 10-20%/year (why increase?)
  - Integration overall: 40-55%/year

Single-Processor Performance

(Hennessy and Patterson slide, Computer Organization and Design, Morgan Kaufmann)



Trends in Technology (Exponential)

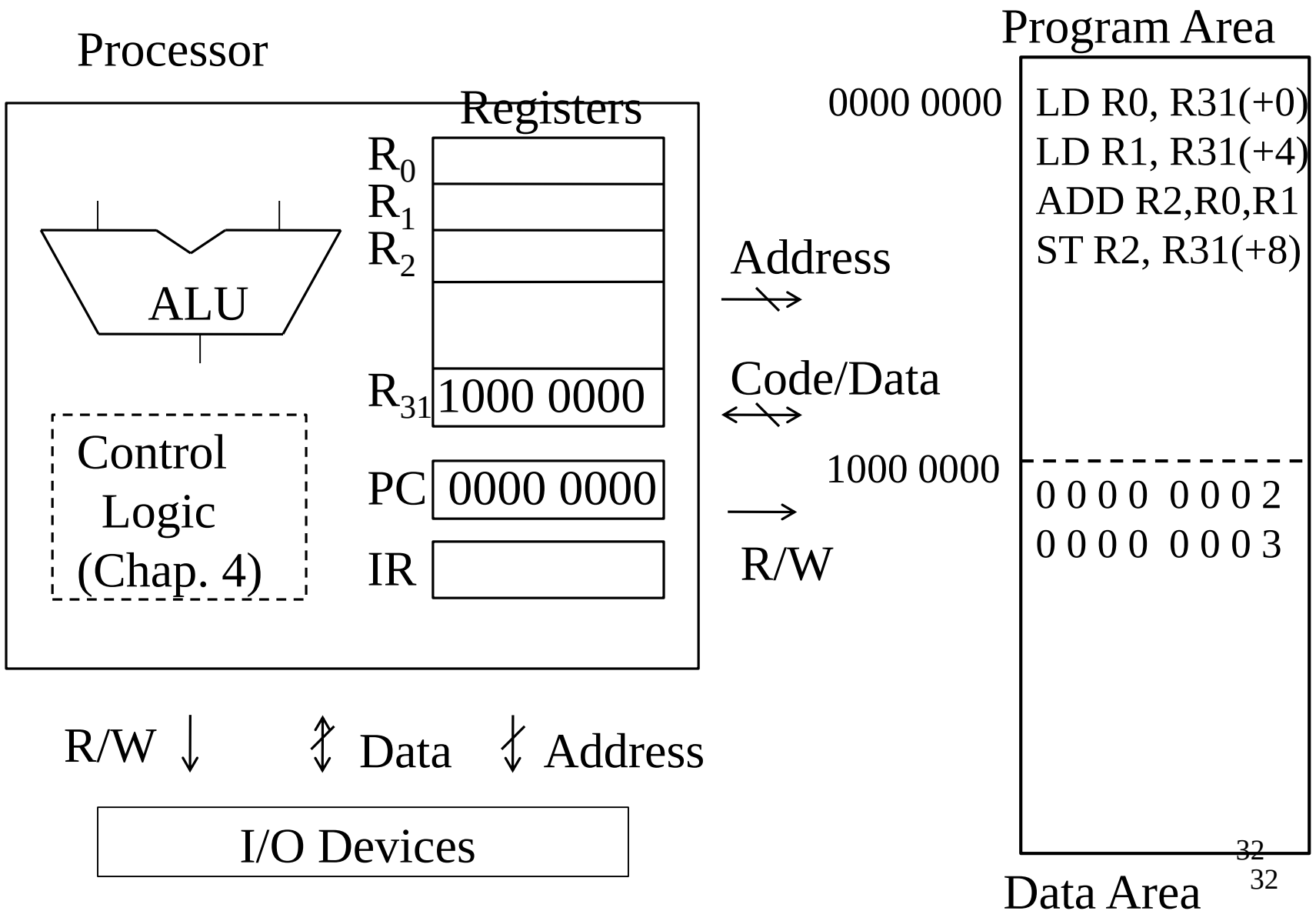
- Processor technology
  - Transistor density: 35%/year
  - Die size: 10-20%/year (pipelines, cache memory)
  - Integration overall: 40-55%/year
- DRAM capacity: 25-40%/year
- Flash memory capacity: 50-60%/year
  - 15-20X cheaper/bit than DRAM
- Magnetic disk capacity: 40%/year
  - 15-25X cheaper/bit than Flash
  - 300-500X cheaper/bit than DRAM

Speed vs. Capacity

More on Computers

(x-bit Computers, Addressing)

32-Bit Computers (반복)

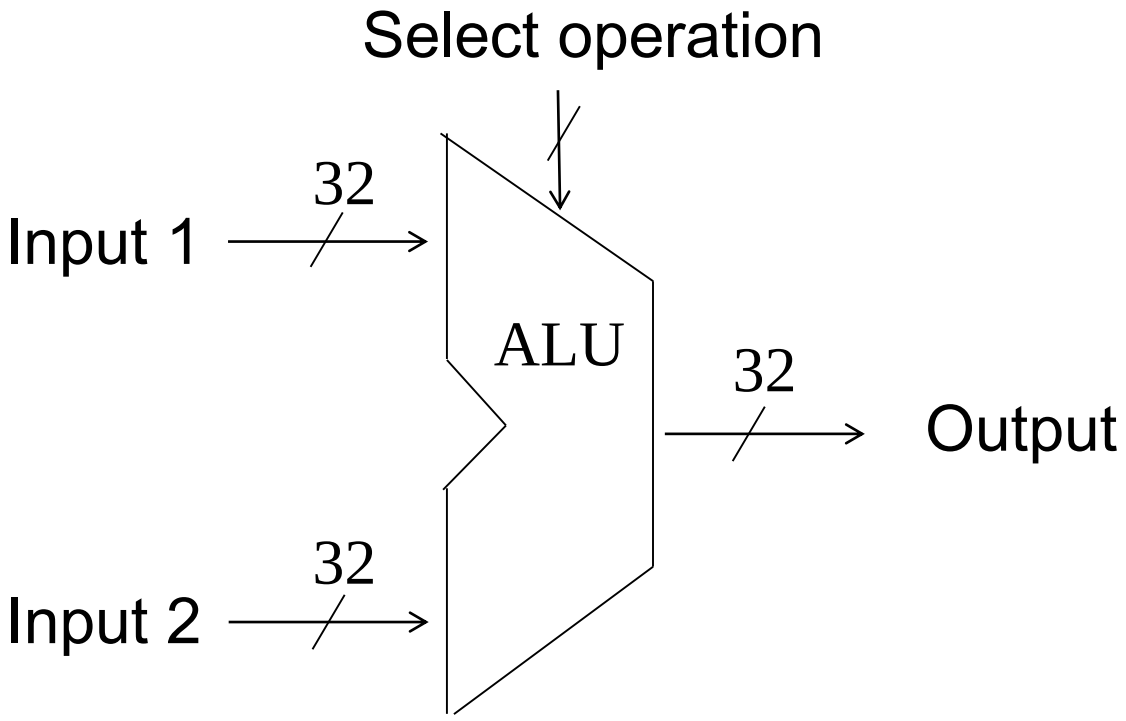


Quiz: X-Bit Computers

- What does 'x-bit' means?
  - Size of ALU input operands
  - Size of registers
  - Width of processor data bus
  - Width of processor address bus
  - Width of I/O bus
  - No. of data lines (pins) of processor
  - No. of address lines (or pins) of processor
  - Length of instructions

X-Bit Computers

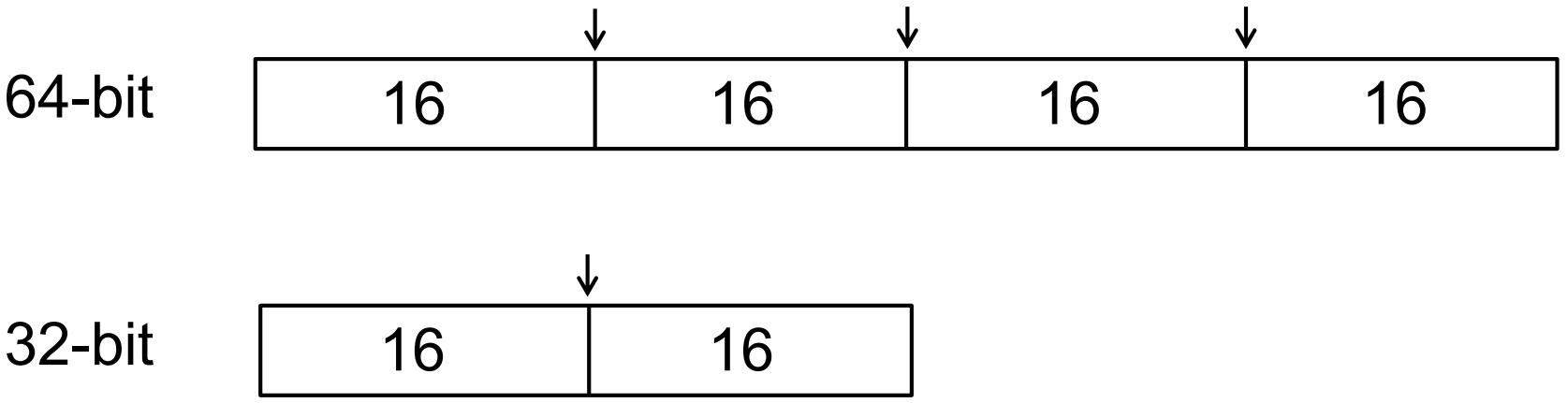
- What does 'x-bit' mean?
  - Size of ALU input operands



- 32-bit vs. 64-bit

X-Bit Computers

- Is a 64-bit computer better than a 32-bit computer?
  - Larger numbers
  - Multimedia: speedup with parallel operations
    - SIMD instructions or subword parallelism (Ch. 3)



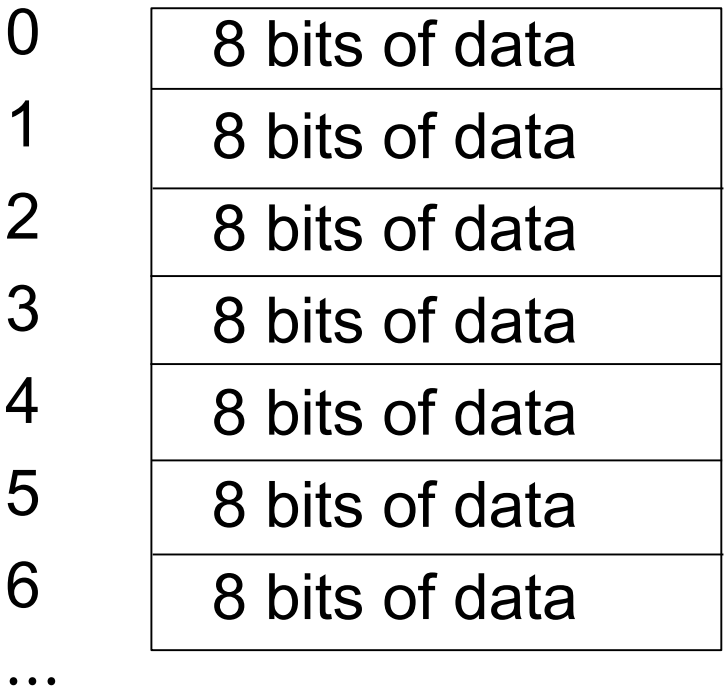
Sizes of Address Spaces

- What else is important?
  - Sizes of address spaces
    - What does that mean to programmers?
- History of single-chip processors (what about mainframes)

Processors	Data Size	Address Size
8-bit	8	16
16-bit	16	16 ($+\alpha$)
32-bit RISC	32	32
64-bit	64	?

Byte Addressing (반복)

- Memory: a large, single-dimension array, with an address
  - A memory address is an index into the array
- Byte addressing: the index points to a byte of memory



Byte Addressing (반복)

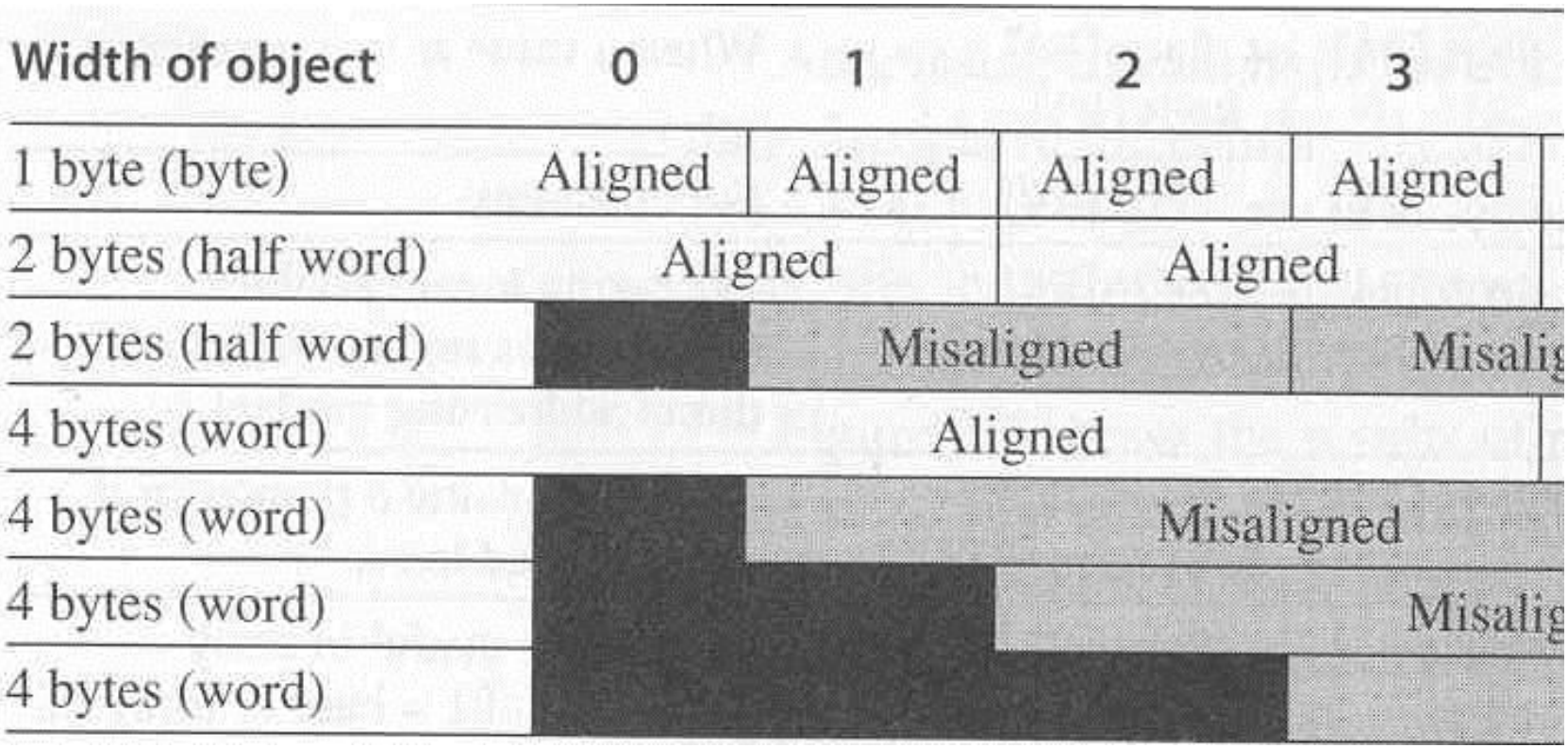
- For 32-bit MIPS, a word is 32 bits (4 bytes)
  - Registers hold 32 bits of data
- Bytes in 4열 종대

0	0	1	2	3	32 bits of data
4	4	5	6	7	32 bits of data
8	8	9	10	11	32 bits of data
12	12	13	14	15	32 bits of data
...					

- Alignment

Alignment

(Hennessy and Patterson, Computer Organization and Design, Morgan Kaufmann)

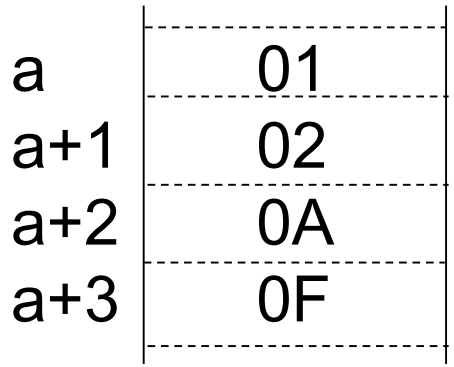


- What are the least 2 significant bits of a word address?

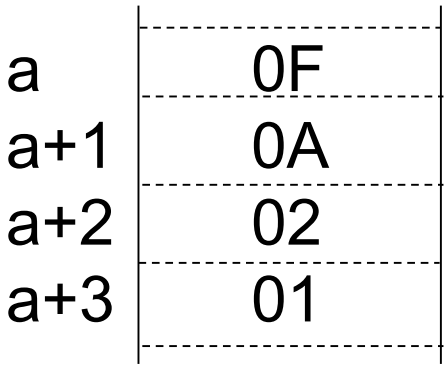
Little Endian and Big Endian

- Around 1950s
  - A few mainframes in the world (like "islands")
  - "not invented here"
- Byte address: multiple conventions (refer to databook)

01020A0F // 32-bit word to store



Big endian

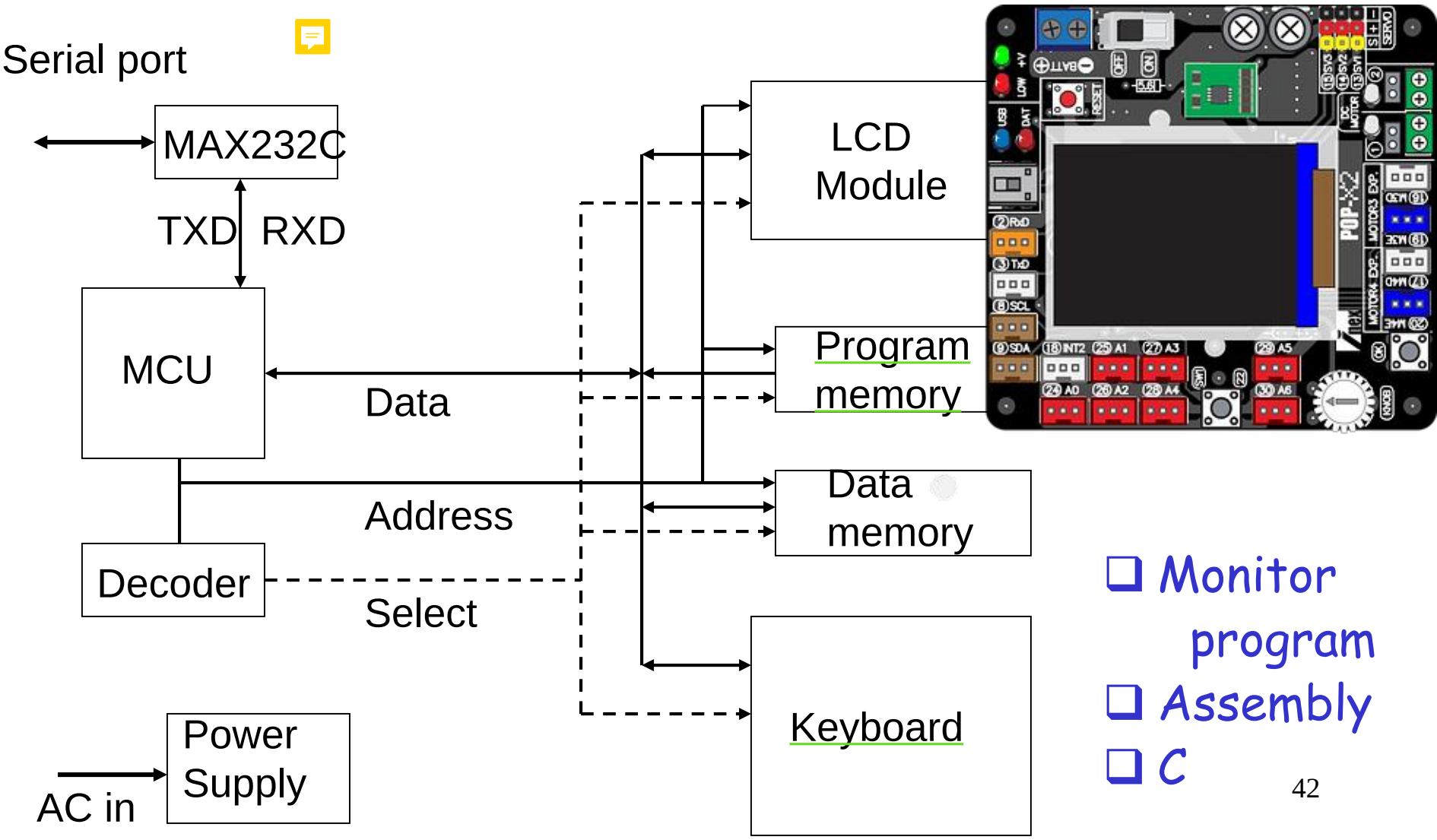


Little endian

More on Computers

(Memory Map, Microcontrollers)

8/16-Bit "Microcontroller" Boards

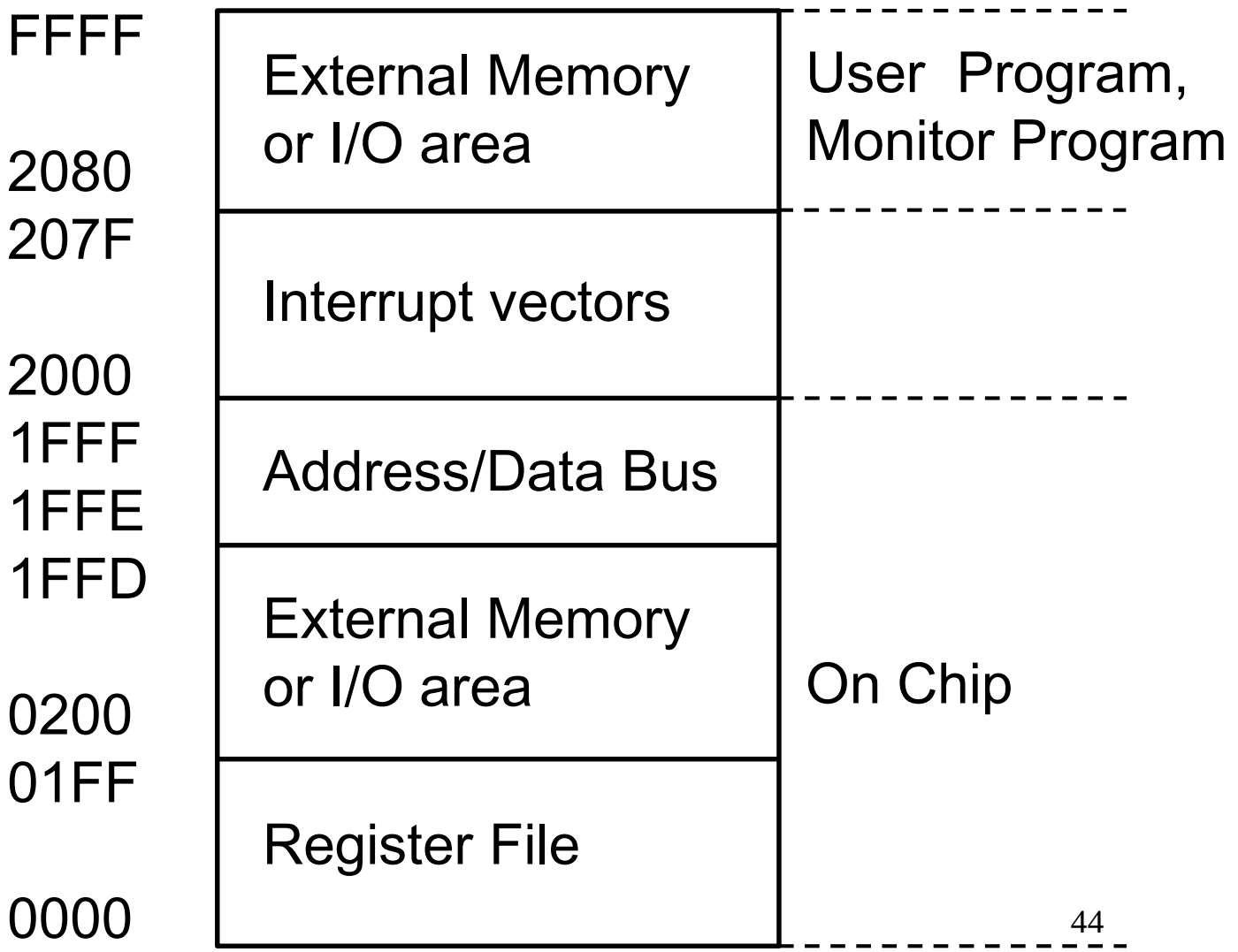


- Monitor program
- Assembly
- C

Memory Map - Address Assignment

- Memory
  - Large number of addresses (for instructions and data)
  - Each memory word has an address
- I/O peripherals (e.g., LCD display)
  - A few addresses for each
- 컴퓨터 하드웨어 설계 과정 중의 하나
  - Unique addresses for memory words and I/O devices
    - Determine memory map to build hardware
  - Programmers use addresses accordingly

Memory Map – 80196 Example



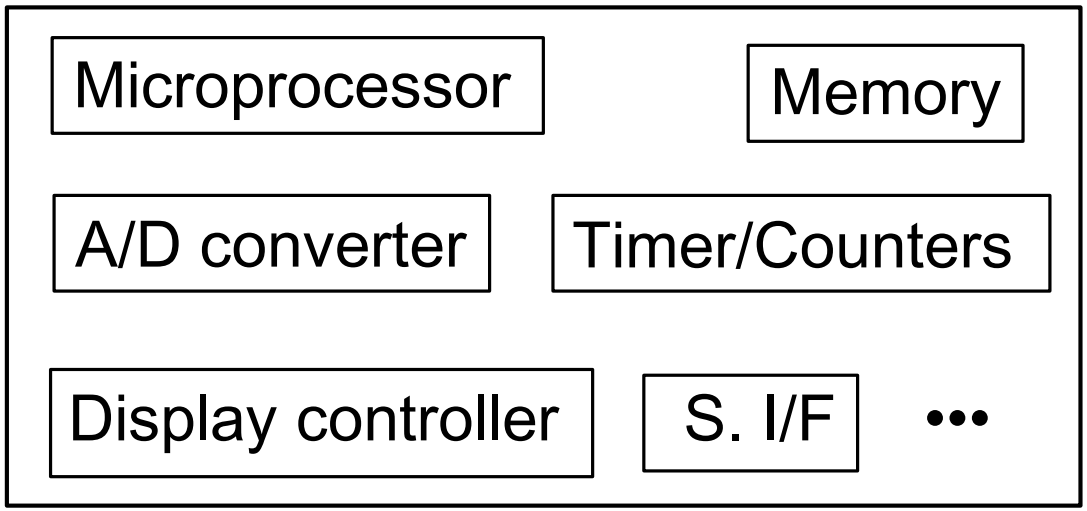
- Memory map of PC?

Processor Databook

- Processor data book - what do you expect to see?
  - ISA (프로그래머 사용법)
    - Instructions, addressing modes, encoding
  - Physical interface (HW designer 사용법)
    - Pins, how to use them, timing
    - Others
      - † Environmental range,
        clock speed, power supply

Microcontrollers

- Embedded systems (e.g., refrigerator controllers)



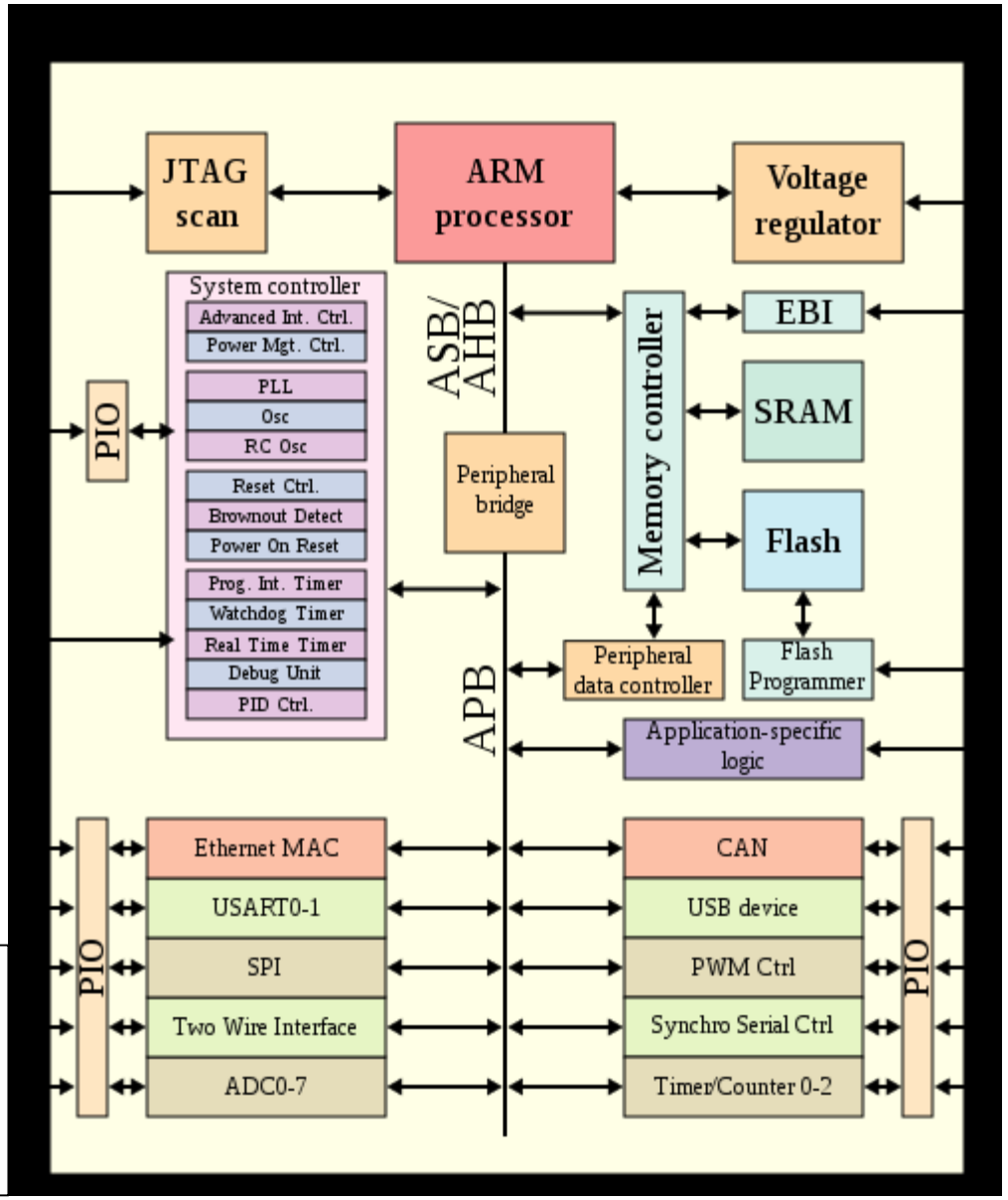
PCB with multiple chips

- Microcontrollers (e.g., ATmega128, 80196)
  - Microprocessors plus set of commonly-used peripherals
- Ideal: single-chip solution
  - Less expensive, more reliable, faster

System on Chip (SoC)

- ARM System on Chip
- Mobile AP

SAMSUNG Exynos



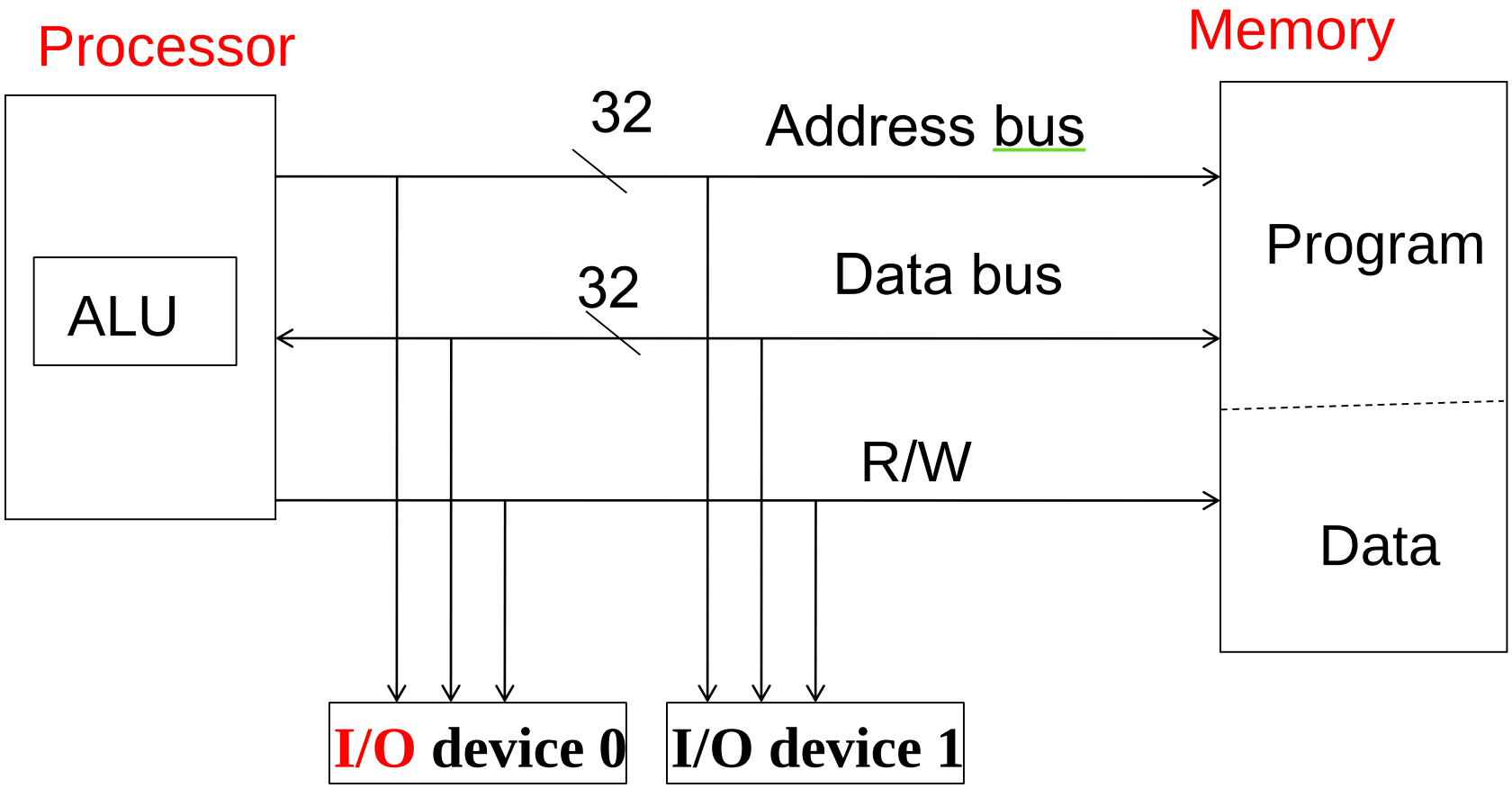
More on Computers

(I/O, Interconnection)

Interconnection

- Connecting processor, memory, I/O devices
  - Data bus and address bus
- What is a bus?
  - Shared medium (e.g., subway)
    - Simple, low cost, widely used
      - † Limited performance
    - Bus arbitration, bus protocol, bus controllers
  - PCI, ISA, CAN, Ethernet

32-Bit Computers



- 4G = 2^{32} memory and I/O locations
- Given address, enable corresponding location

Interconnection

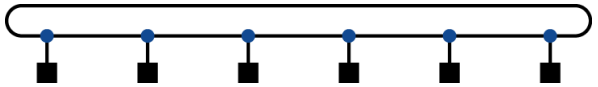
- Connecting many processors (data centers, supercoms.)
  - Cannot use bus for performance reason
- The other extreme
  - Fully connected (order of n^2)
- Alternate topologies
  - Different cost-performance trade-offs
  - Mesh, tree, hypercube, multistage networks, …

Interconnection Networks (see §6.8)

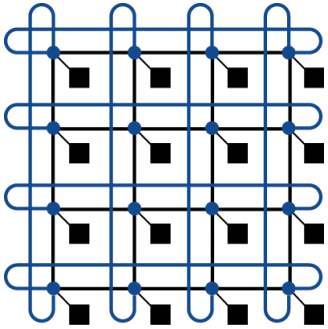
(Hennessy and Patterson slide, Computer Organization and Design, Morgan Kaufmann)

- Network topologies
  - Arrangements of processors, switches, and links

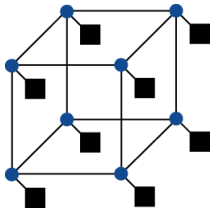
Bus



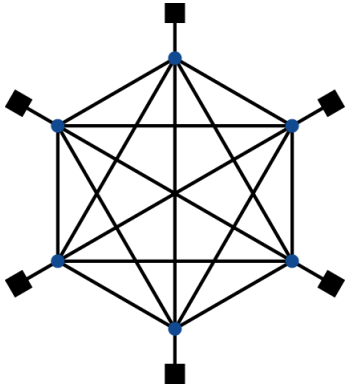
Ring



2D Mesh



N-cube (N = 3)



Fully connected

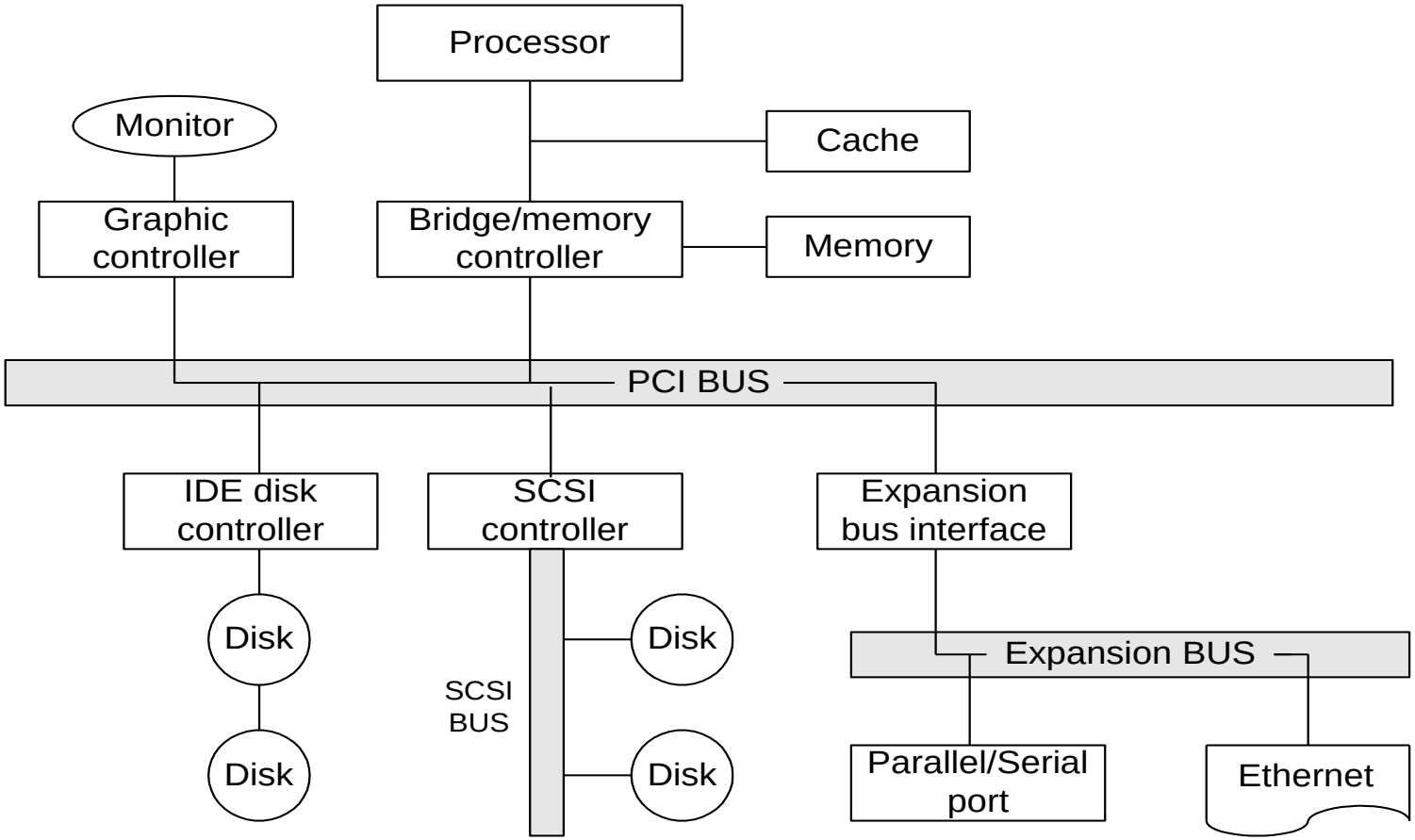
Interconnection

- General interconnection issues
  - Unique address
  - Routing: how to deliver messages
- Bus (broadcast)
- Internet (cf. postal service)

Interconnection

(Hennessy and Patterson slide, Computer Organization and Design, Morgan Kaufmann)

- Processor-memory bus: proprietary
- I/O bus: standard (industry-driven)



More on Computers

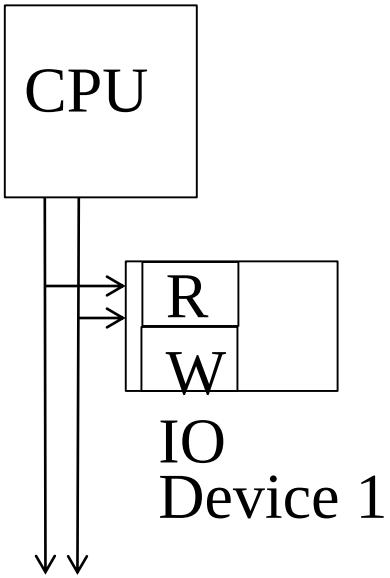
(I/O, Interrupts)

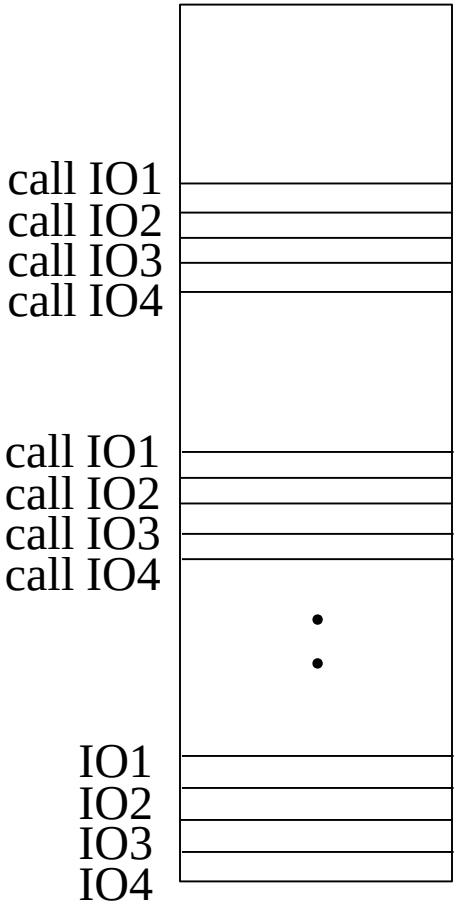
Programmed I/O vs. Interrupt

- I/O devices: many of them, slow, sporadic use
- What is programmed I/O?
  - Periodic polling with existing hardware (address and data bus)
- Why programmed I/O not sufficient?
  - Can be burden to processor (in large systems)
  - Response time
- Interrupts
  - Extra hardware to process requests from I/O devices

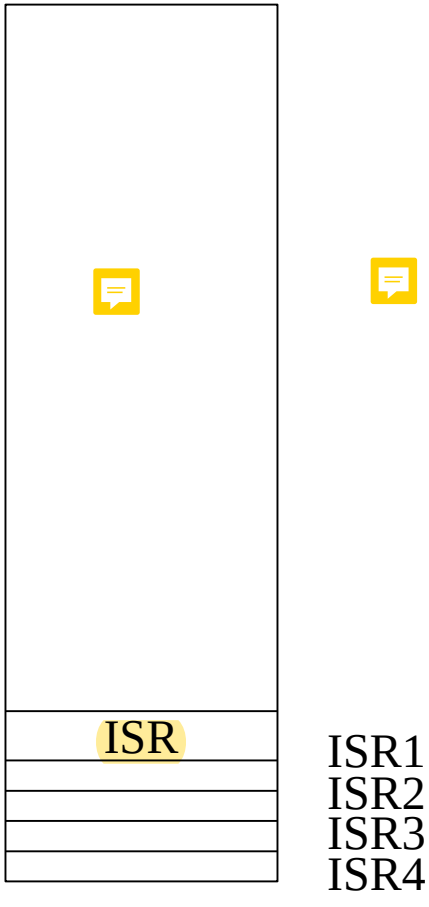
Programmed I/O vs. Interrupt

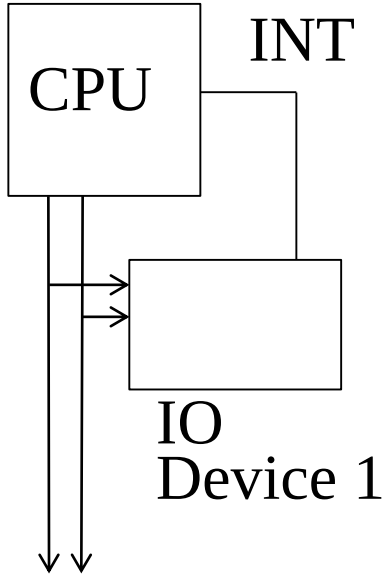
Processor: periodic IO checks





Processor: extra mechanisms

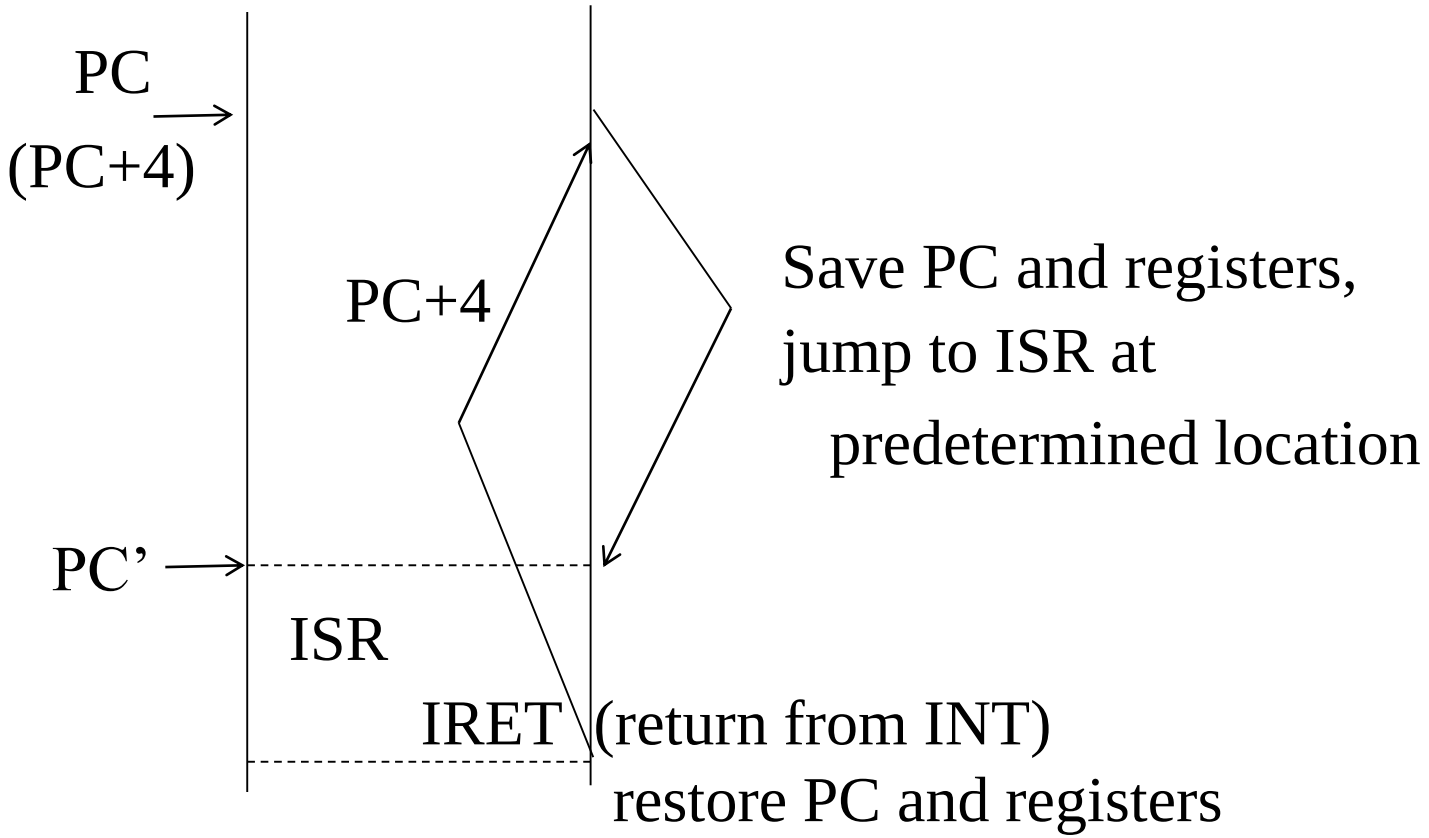




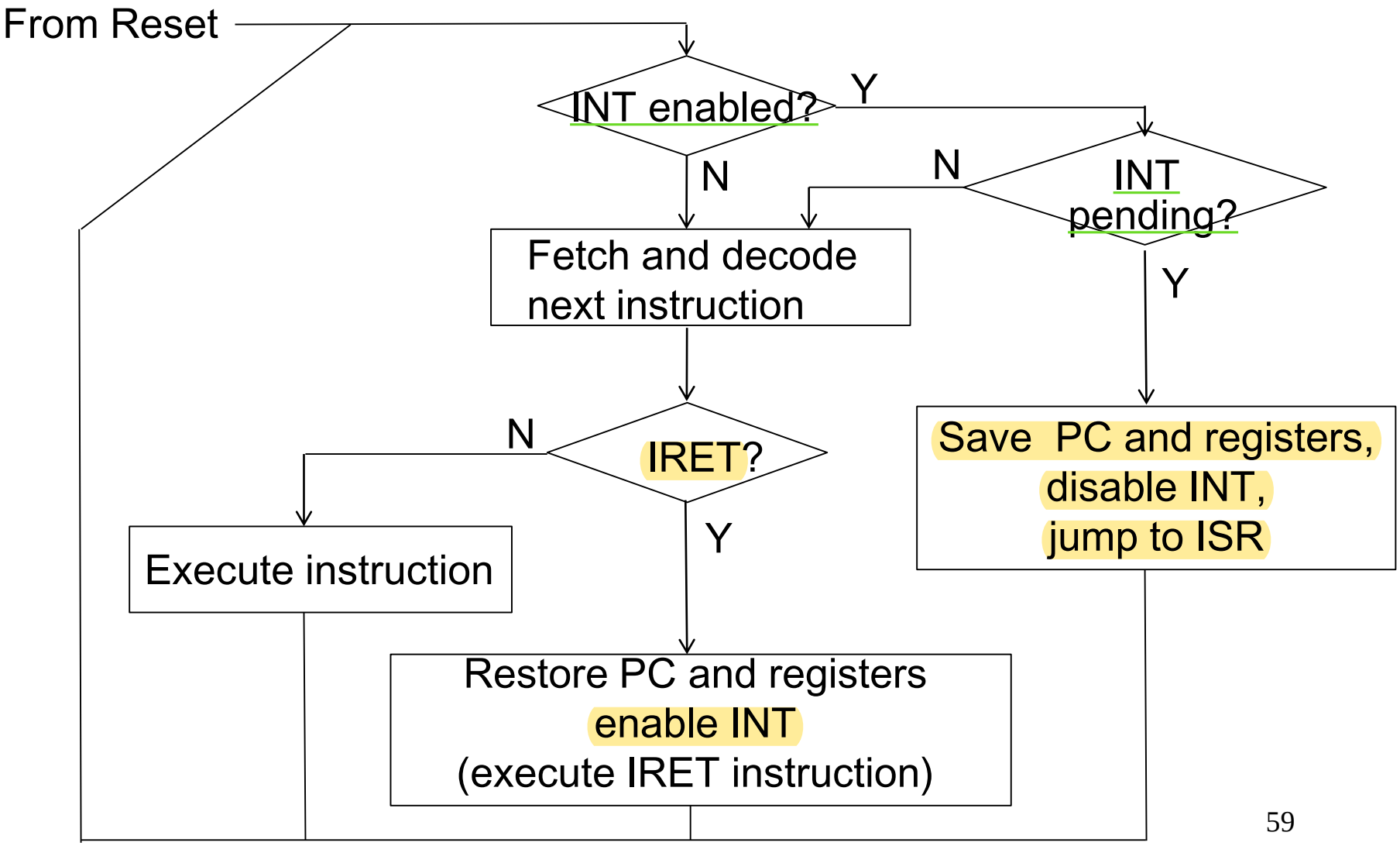
- On accepting INT, jump to ISR

Jump to ISR, Return from INT

- Interrupt Service Routines (ISRs)

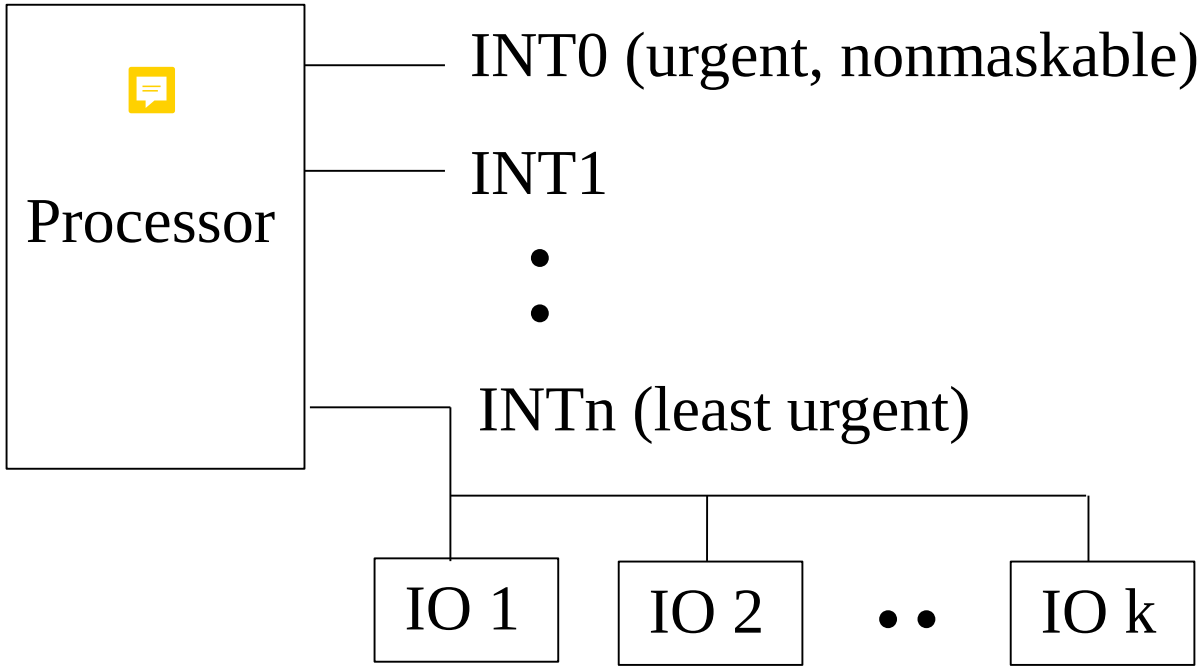


Interrupt Processing and F-D-E



Multiple INTs and INT Priority

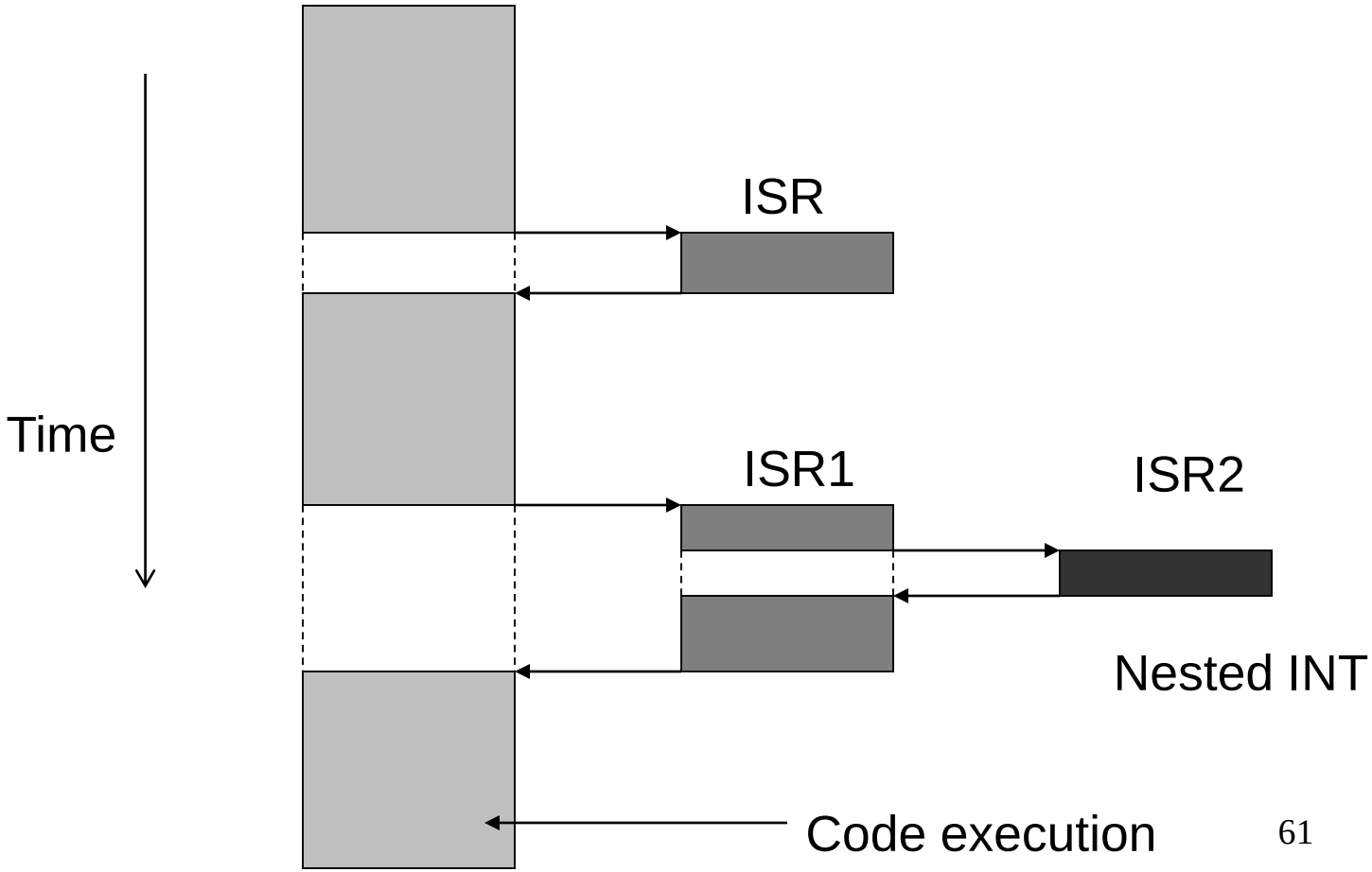
- System designers allocate I/O devices accordingly



- Interrupts (외부 요인) vs. exceptions (실행 중인 프로그램)
  - 범용 컴퓨터에서는 두 경우 다 OS 불려짐

Interrupt Nesting

- INT priority, INT vectors



Fetch-Decode-Execute & Interrupts

- Computers
  - Fetch-decode-execute, adequate ISA, interrupts
    - Special instructions: enable INT, disable INT
      - † Privileged machine instructions
- Machine instructions
  - Atomic (all or nothing)
    - Interrupts checked after an instruction is finished
    - Indivisible atoms

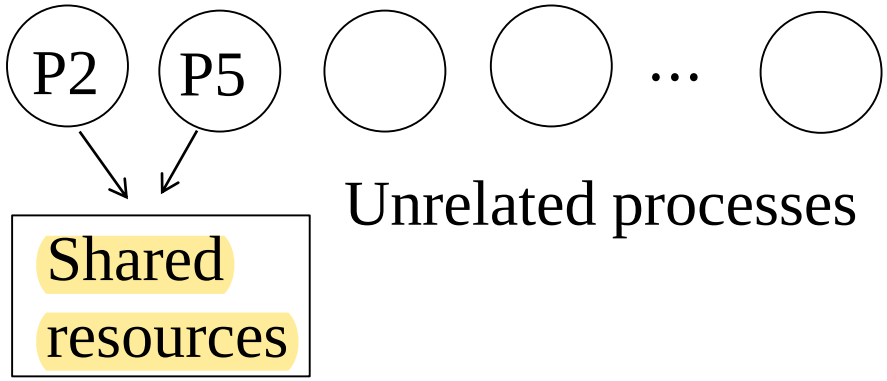
Atomic Operations

- Sequence of machine instructions need to be executed atomically
  - Critical sections in OS
  - Transactions in database
    - e.g., ticket reservation
- Implementing atomic operations
  - May use "enable INT" and "disable INT" instructions
    - Only in small embedded systems

Atomic Operations

- Small embedded systems (direct control of hardware)

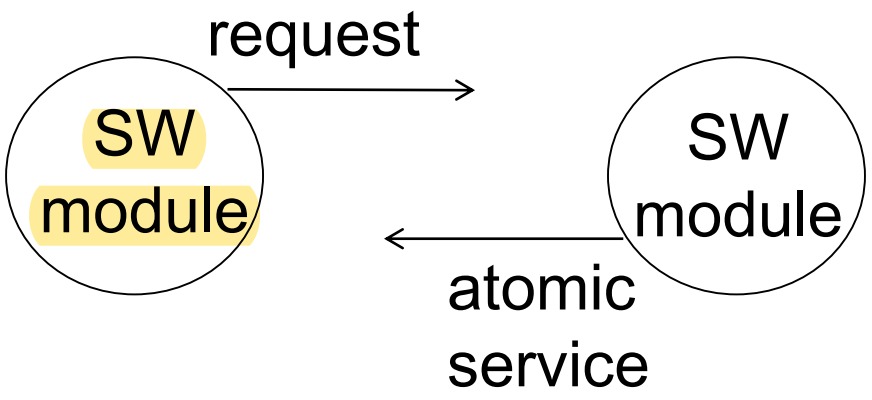
```
disable INT
    critical sections
    or transactions
enable INT
```



- General-purpose systems (many unrelated processes)
  - Problem: too much intrusion, all others affected
    - Disable timer interrupt and OS process scheduling
  - Solution: synchronization instructions (and lock library)
    - Textbook Section 2.10

Atomicity and Recoverability

- Perfect or fault-tolerant modules would be ideal
  - Next best: recoverability with atomic services



- HW-SW interactions
  - Synchronization instructions, enable/disable INT

Real-Time Systems (참고)

- Real-time systems (response time, deadline)
  - Hard RTS
  - Soft RTS
- RTOS (real-time OS) for embedded systems
  - Priority-based preemptive scheduling
    - General-purpose OS: fairness

3단계 수업 세부 목표

- 컴퓨터, 컴퓨터구조, 컴퓨터 사이언스
  - 기본적이고 핵심적인 개념과 원리
- ISA 를 어떻게 설계해야 빠른 컴퓨터를 만들 수 있는가?
  - Performance, RISC versus CISC
- ISA 를 어떻게 구현해야 빠른 컴퓨터를 만들 수 있는가?
  - 파이프라인 및 캐시 메모리

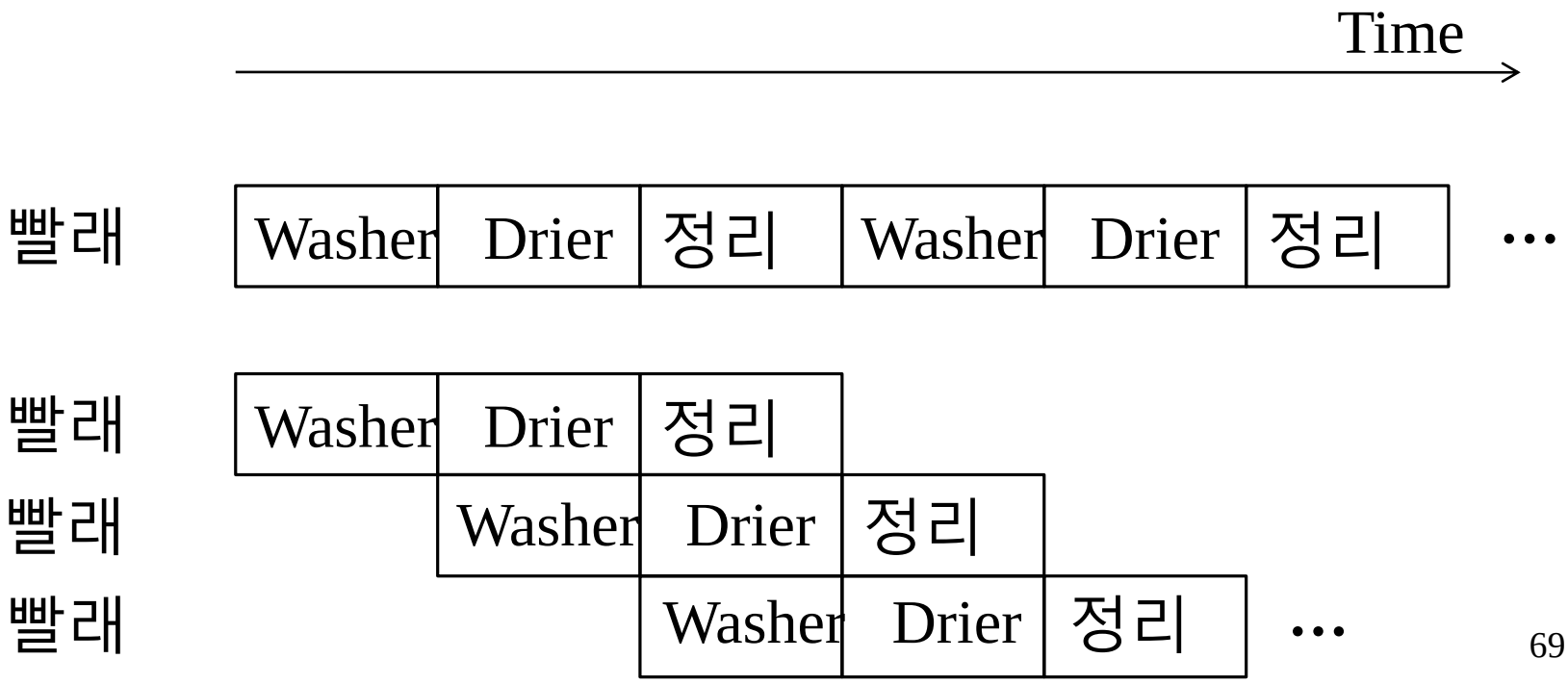
Part 3 미리보기

성능 높이기 위한 핵심적인 구현 방법

- Pipelining (Chapter 4)
- Cache Memory (Chapter 5)

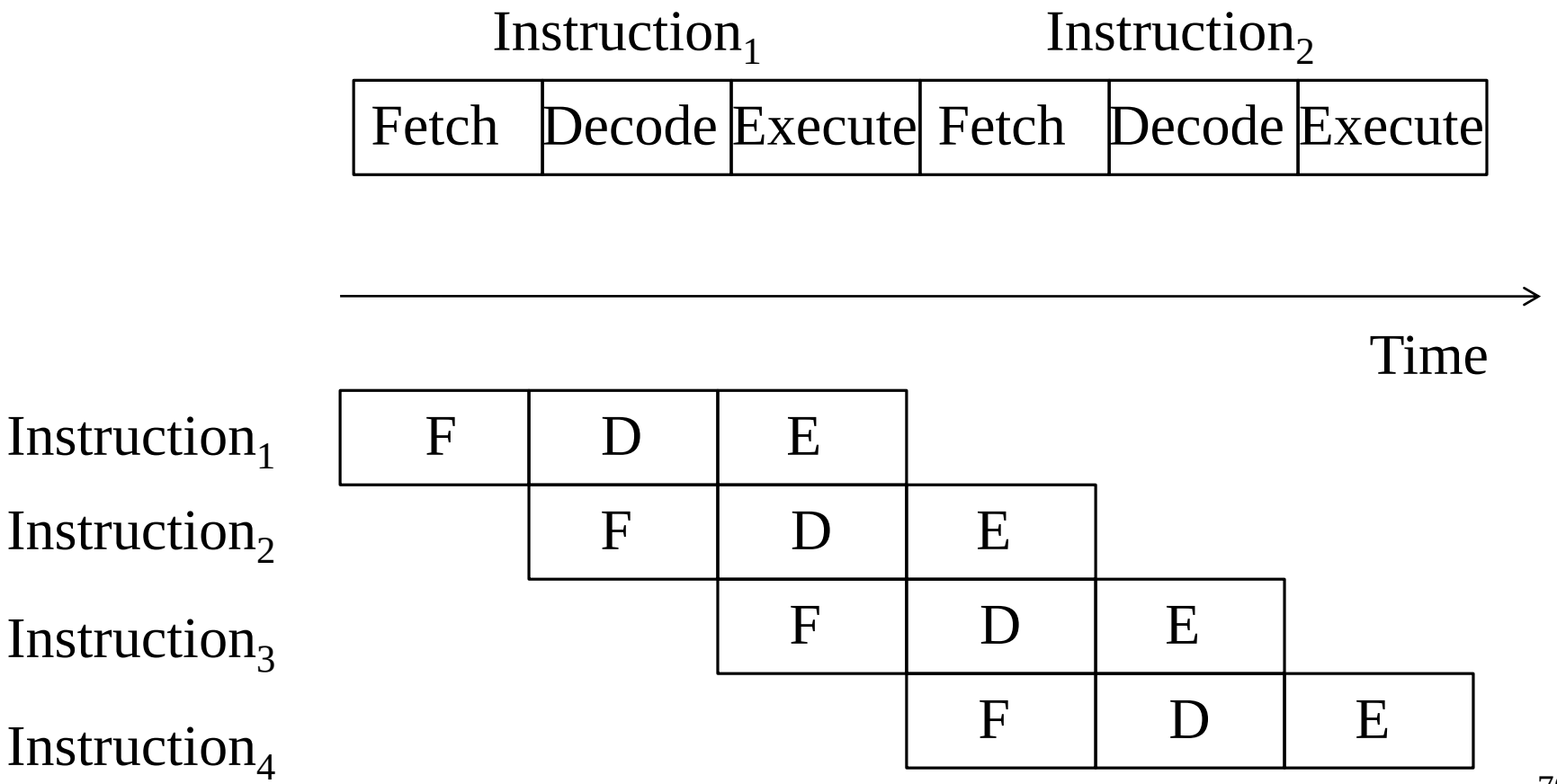
Pipelining: General Speedup Technique

- 3-stage pipeline (e.g., washer-dryer example)
  - Speedup?



Pipelining

- 3-stage pipeline for fetch-decode-execute



Advanced Pipelining (skip)

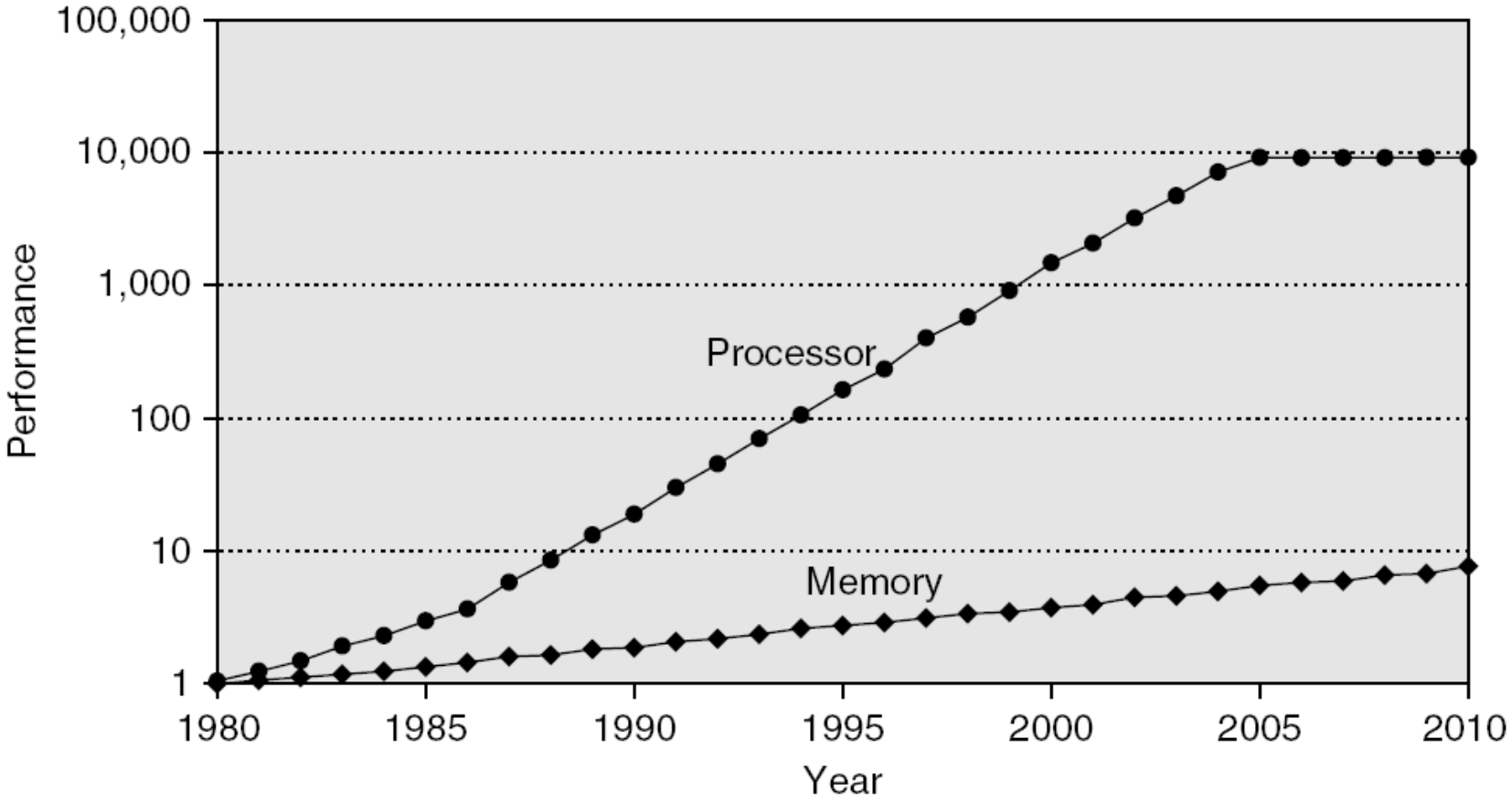
- Powerful processors
  - Ideal speedup for 10-stage pipeline?
  - What if we build 4 pipelines on a processor?
    - What is the ideal speedup?

Cache Memory

- Major driving forces behind computer performance evolution
  - Smaller transistors 및 컴퓨터 설계 기술 발전
- Smaller transistors and increased die size
  - Processor perspective
    - Exponential growth in performance
  - Memory perspective
    - Exponential growth in capacity
    - Speed (i.e., access time) improves slowly

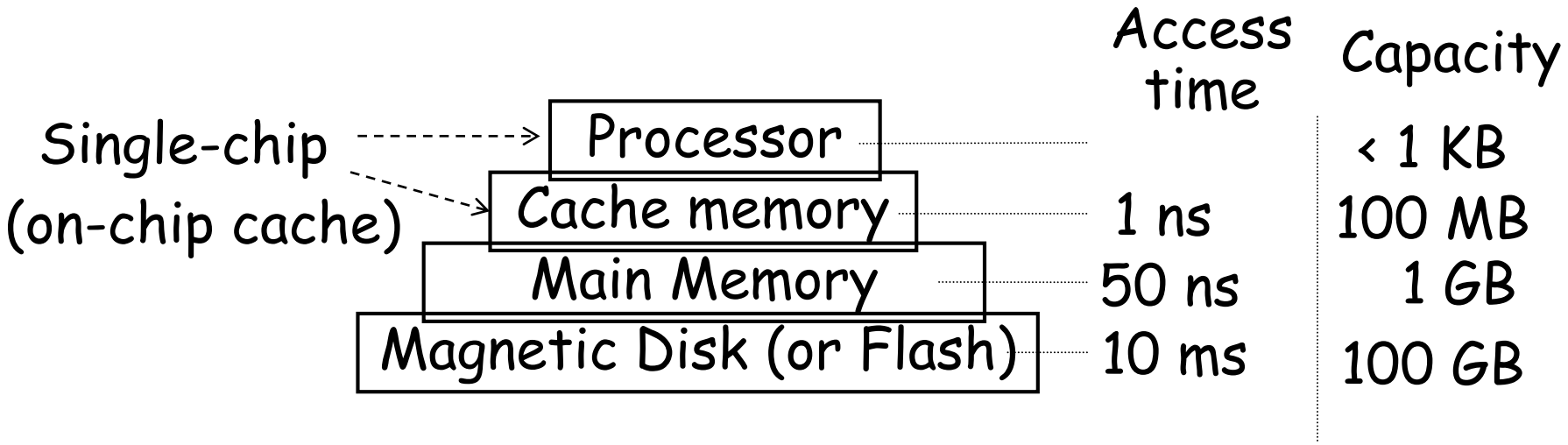
Processor-Memory Performance Gap

(Hennessy and Patterson, Computer Organization and Design, Morgan Kaufmann)



Cache Memory

- Memory is slow, performance bottleneck
- How to hide it?
  - Caching and cache memory



Semiconductor Memory

- SRAM
  - Flip-flop invented by Eccles and Jordan in 1918
  - Cache memory, volatile
- DRAM invented in 1966, IBM
  - Main memory, volatile
- Secondary, non-volatile memory
  - NAND Flash memory
    - Invented around 1980, Toshiba
    - Compete with hard disks, took over mobile market

† Hard disks, invented by IBM in 1953

Summary

- Semiconductor technology
- Processor technology and Intel
- More on computers
  - X-bit computers
  - Byte addressing, memory map
  - Microcontrollers and SoC
  - I/O, interconnection, interrupts
- 미리보기
  - Pipelines and cache memory

Homework #4 (see Class Homepage)

1) Write a report summarizing the materials discussed in Topic 0-5
2) Read the textbook section 1.12 and write a summary report - you can obtain the section 1.12 by clicking "online companion materials" in the Class Homepage and then clicking "Historical perspectives and further reading" on top-left

** 문장으로 써도 좋고 파워포인트 형태의 개조식 정리도 좋음

- Submit electronically to Blackboard

Class Topics (클래스 홈페이지 참조)

- Part 1: Fundamental concepts and principles
- Part 2: 빠른 컴퓨터를 위한 ISA design
  - Topic 1 Computer performance and ISA design (Ch. 1)
    - 1-1 Performance evaluation & performance models
    - 1-2 RISC versus CISC, power limit
  - Topic 2 RISC (MIPS) instruction set (Chapter 2)
  - Topic 3 Computer arithmetic and ALU (Chapter 3)
- Part 3: ISA 의 효율적인 구현 (pipelining, cache memory)

Questions on Data Loss (skip)

- I don't want to lose the data in my PC
  - Backup in optical disks or in external hard disks?
  - How long would it last?
    - What is your solution?
- Financial companies in New York
  - Risk: war, earthquake, tsunami, …
    - What is the state-of-the-art?
- Heard about company specialized in backup and archive?
  - What kind of facilities would they have?

Redundancy and Diversity (skip)

- Storage
  - RAID, disk mirroring
- Controllers in early spacecraft
  - 4 identical processors monitoring each other
  - 1 separate processor with different SW for backup
- Given no reply, resend requests
- Error detection and correction code
- Redundancy in natural languages