EECS4030: Computer Architecture

Computer Abstractions and Technology (I)

Prof. Chung-Ta King
Department of Computer Science
National Tsing Hua University, Taiwan

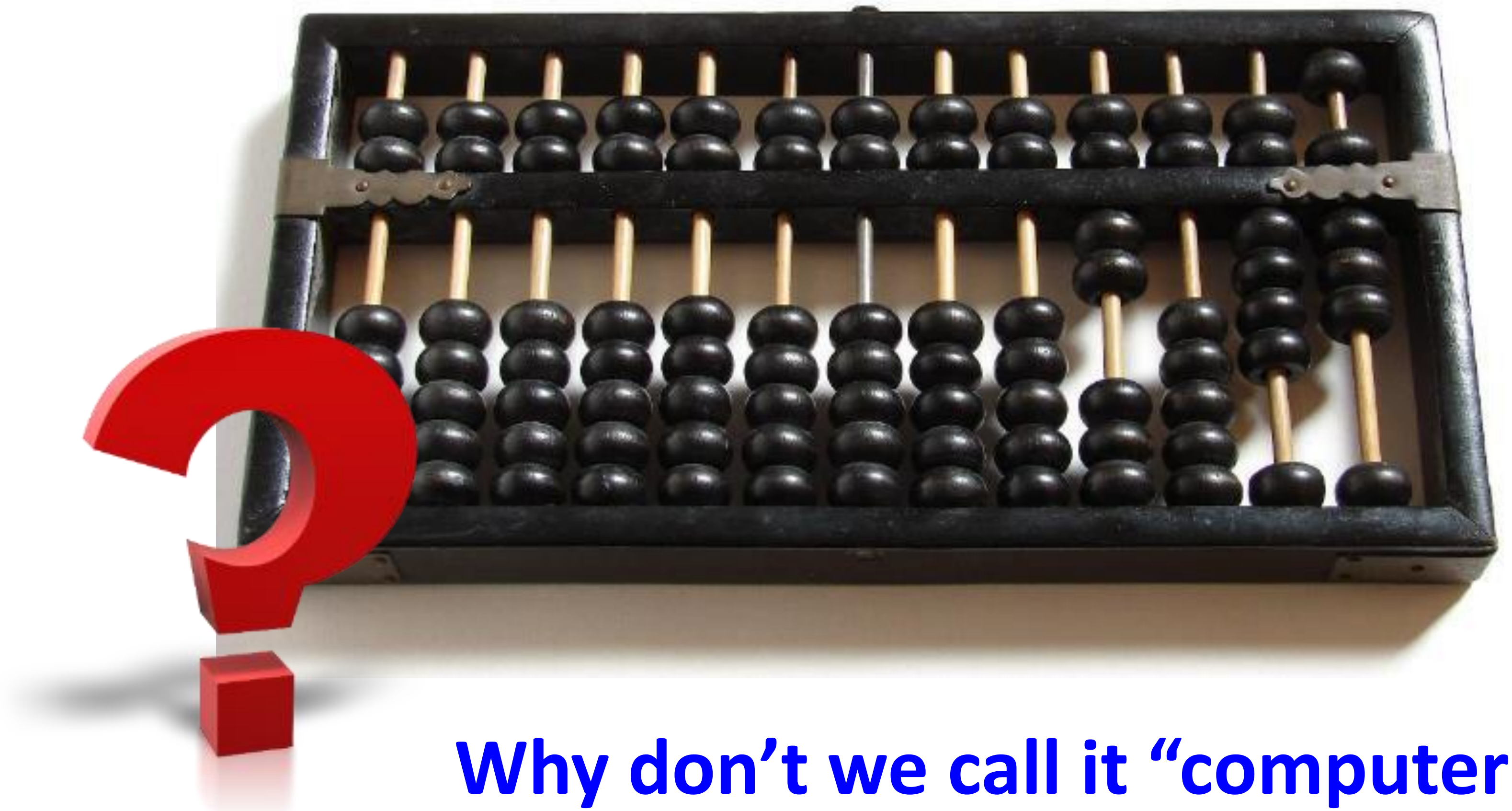
(Adapted from textbook slides https://www.elsevier.com/books-and-journals/book-companion/9780128122754/lecture-slides)

Outline

- Computer: a historical perspective (Sec. 1.1, 1.5, 1.12)
  - Technology affects computer development
  - Great ideas in computer architecture require technology
- Great ideas in computer architecture (Sec. 1.2)
- Below your program (Sec. 1.3)
- Under the covers (Sec. 1.4)
- Technologies for building processors and memory (Sec. 1.5)
- Performance (Sec. 1.6)
- The power wall (Sec. 1.7)
- From uniprocessors to multiprocessors (Sec. 1.8)

When was "computer" developed?

About 1300 years ago …



Why don't we call it "computer"?

Have to Be Precise about What We Mean

- A device that computes, especially a programmable electronic machine that performs high-speed mathematical or logical operations or that assembles, stores, correlates, or otherwise processes information
  -- The American Heritage Dictionary of the English Language, 4th Edition, 2000

There Were Many Computing Devices

- Special-purpose versus general-purpose
- Non-programmable versus programmable
- Decimal versus binary (Boolean algebra)
- Scientific versus data processing
- Mechanical, electromechanical, electronic, …

Difference Engine
(C. Babbage, 1822)

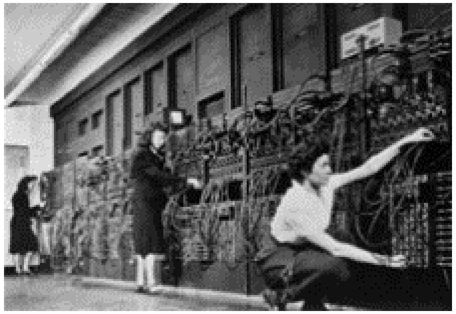
Tabulating machine
(H. Hollerith, 1889)

Harvard Mark I
(IBM, H. Aiken, 1944)

When was the first all-electronic, programmable, general-purpose computer developed?

First Modern "Computer"

- ENIAC: *Electronic Numerical Integrator and Calculator*
  - Widely accepted to be the world's first operational electronic, general-purpose computer
  - Built by J. Presper Eckert and John Mauchly at the Moore School of the University of Pennsylvania
  - Work started in 1943, operational during World War II, publicly disclosed after 1946



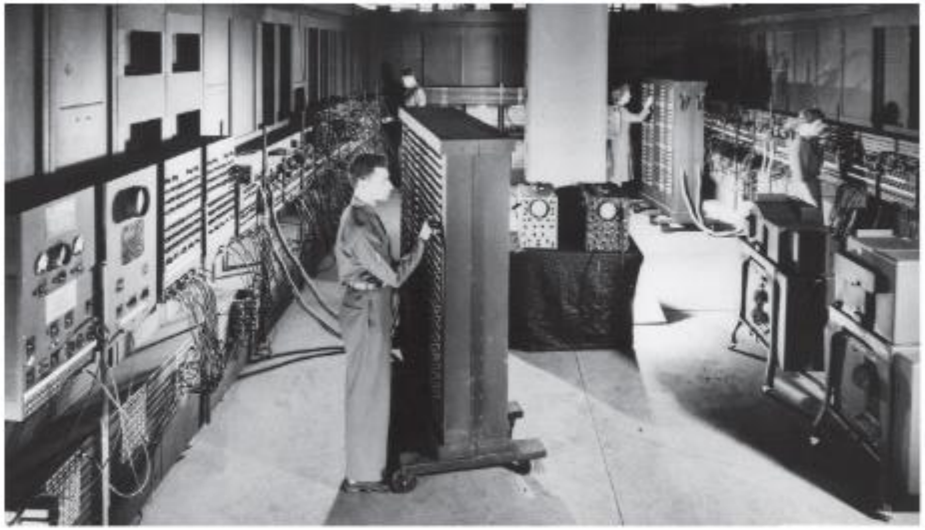
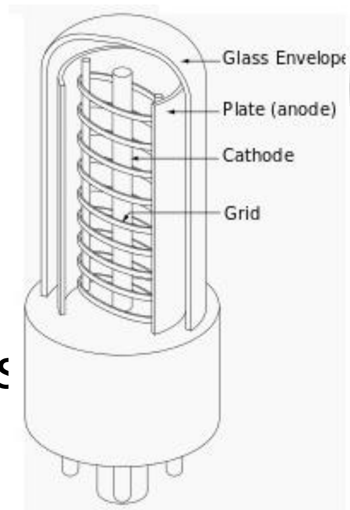
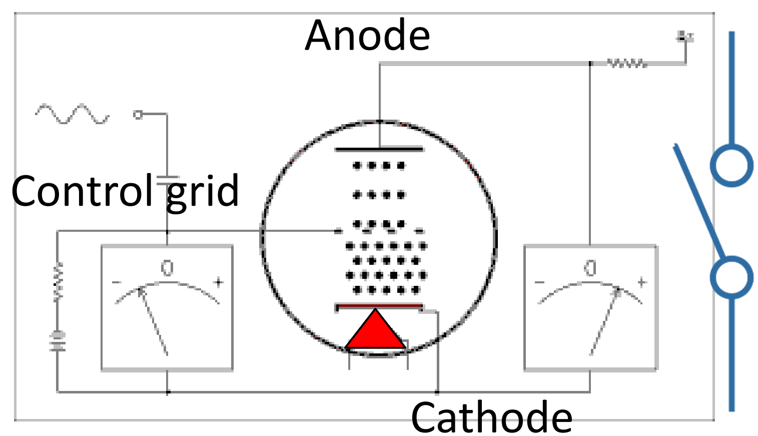


Fig. 1.12.1

ENIAC

- Size:
  - 80 feet long by 8.5 feet high, several feet
  - 20 10-digit registers, each 2 feet long
- Programming:
  - Manually by plugging cables and setting s
- Performance:
  - 1900 additions/sec
- Technology:
  - *Vacuum tube* (used 18,000 vacuum tubes, which are electronic switches developed around 1906)

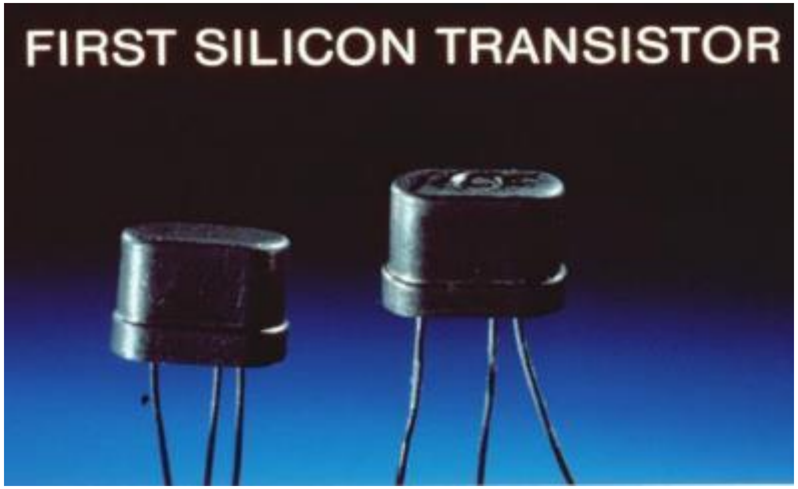


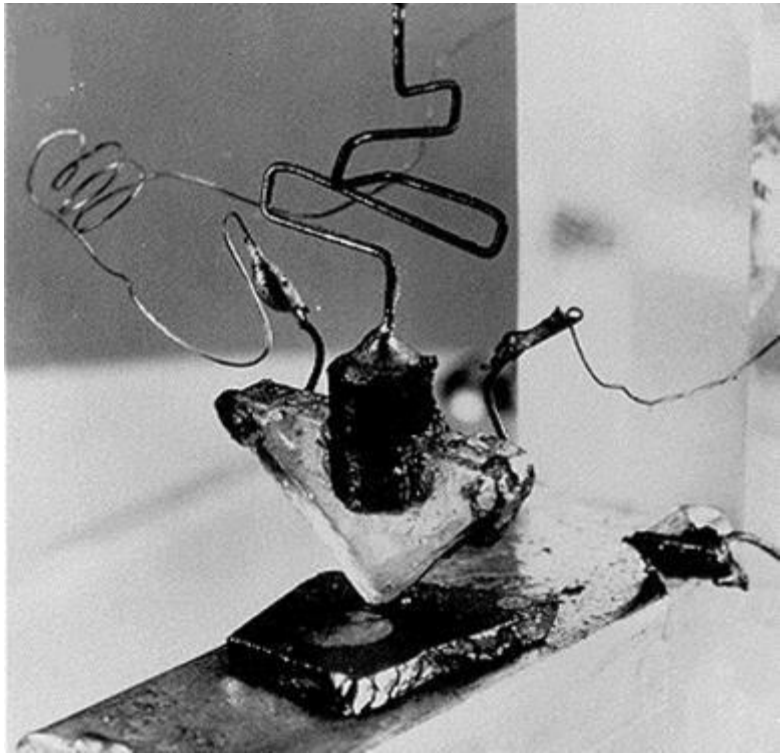


https://electronics.stackexchange.com/questions/156301/how-does-a-vacuum-tube-amplifier-work

Transistors Were Invented at Same Time

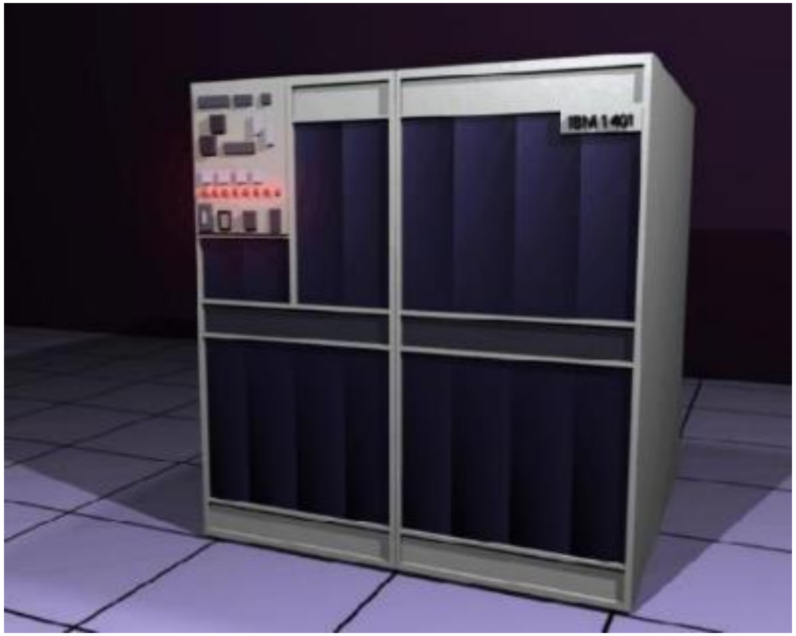
- W. Shockley, J. Bardeen, W. Brattain of Bell Lab. (1947)
  - More reliable, smaller, faster than vacuum tubes
  - Electronic switches in "solids"





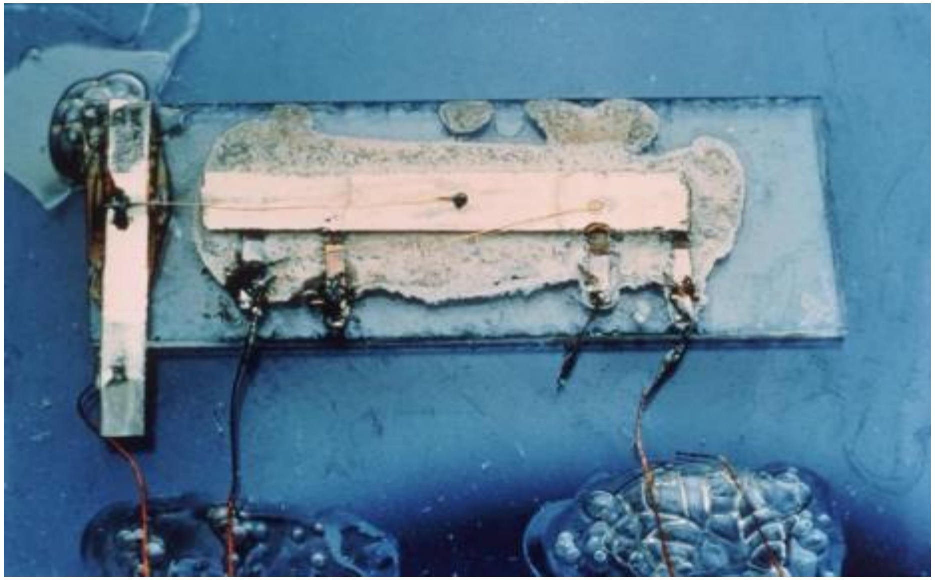
Computers Start Leveraging Transistors

- Transistor computers: *2nd generation computers*
  - First operational transistor computer: *Transistor Computer* by Univ. of Manchester, operational in Nov. 1953
  - Commercial transistor computers also appear soon after, e.g. IBM 1401 (1959)



Another Technology Breakthrough: IC

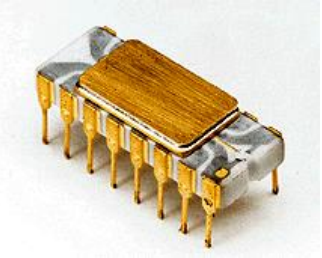
- Jack Kilby of Texas Instrument (1958):
  – Integrated a transistor with resistors and capacitors on a single semiconductor chip, which is a monolithic IC
  – Fast on-chip signal communication, smaller size, more reliable, easier to implement, cheaper



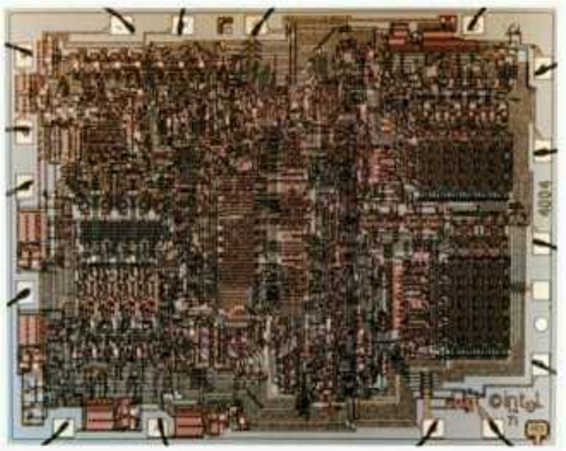
- Moore's law comes into play (2X transistors/chip every 1.5~2 years)
  – Shrinking transistors

As IC Chips Scale …

- We can put an entire processor onto a chip
  - First microprocessor (μP): Intel 4004 (1971)
    - 108 KHz, 0.06 MIPS, 2250 transistors (10 microns)
  - 8-bit and 16-bit μP soon appeared, enabling low-cost, high-volume home/personal computers



- How about mainstream 32-bit processors on a chip?
  - Squeeze 32-bit processors of mainframe (IBM 360) or minicomputer (VAX-11) onto the chip
    - Use CISC ISA for semantic gap & slow IC
  - Or, new architecture design strategy
    - RISC, pipelining, cache, … as IC becomes faster (but mostly developed in the 60’s)
    - OS, compiler, EDA tools, … allow new ISA



Importance of Technology Push

- Electronics technology continues to evolve
  - Increased capacity and performance
  - Reduced cost

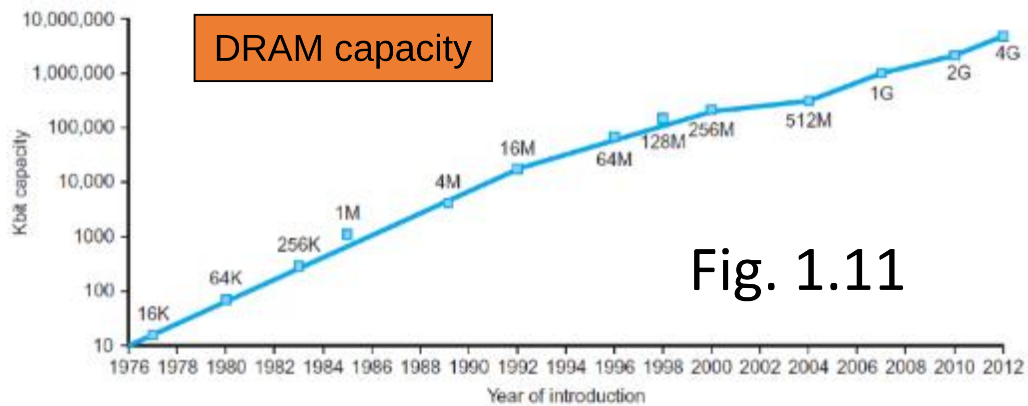
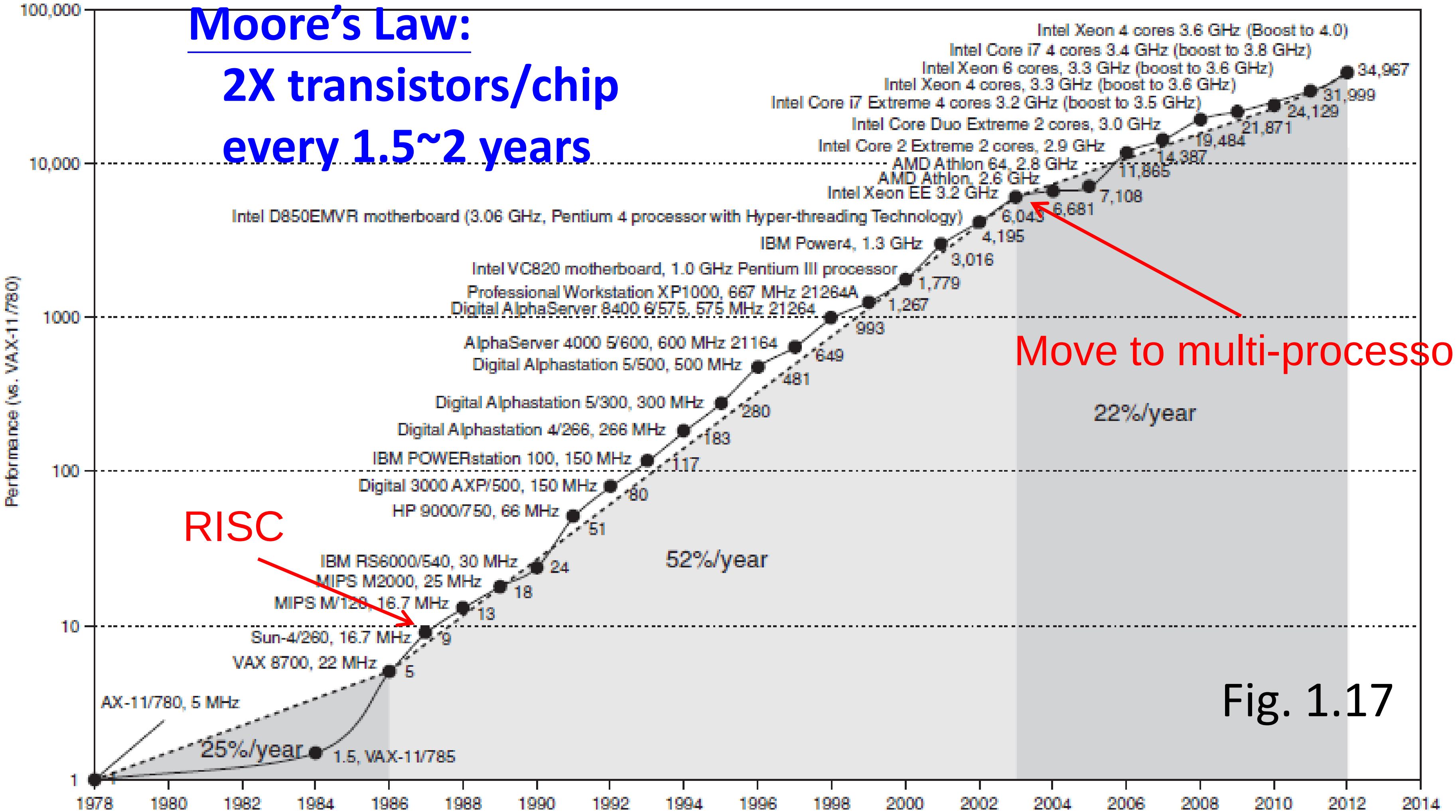


Fig. 1.11

Fig. 1.10

Year	Technology in computer	Relative performance/cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2013	Ultra large scale IC	250,000,000,000

Arch. Innovations Ride Technology Scaling



Technology Affects Software, Too!

Year	Logic	Storage	Prog. lang.	OS
54	vacuum tubes	core (8 ms)		
58	transistors (10 ms)		Fortran	
60			Algol, Cobol	
66	IC (100ns)			
67				Multiprog.
71	LSI (10ns)	1K DRAM	O.O.	V.M.
73	8-bit μP			
75	16-bit μP	4K DRAM		
78	VLSI (10ns)	16K DRAM		Networks
84	32-bit μP	256K DRAM	C++	
92	64-bit μP	16M DRAM	Fortran 90	

Comparison of Computers Since 1950

Year	Name	Size (cu. ft.)	Power (watts)	Performance (adds/sec)	Memory (KB)	Price	Price/ performance vs. UNIVAC	Adjusted price (2007 $)	Adjusted price/ performance vs. UNIVAC
1951	UNIVAC I	1,000	125,000	2,000	48	$1,000,000	1	$7,670,724	1
1964	IBM S/360 model 50	60	10,000	500,000	64	$1,000,000	263	$6,018,798	319
1965	PDP-8	8	500	330,000	4	$16,000	10,855	$94,685	13,367
1976	Cray-1	58	60,000	166,000,000	32,000	$4,000,000	21,842	$13,509,798	47,127
1981	IBM PC	1	150	240,000	256	$3,000	42,105	$6,859	134,208
1991	HP 9000/ model 750	2	500	50,000,000	16,384	$7,400	3,556,188	$11,807	16,241,889
1996	Intel PPro PC (200 MHz)	2	500	400,000,000	16,384	$4,400	47,846,890	$6,211	247,021,234
2003	Intel Pentium 4 PC (3.0 GHz)	2	500	6,000,000,000	262,144	$1,600	1,875,000,000	$2,009	11,451,750,000
2007	AMD Barcelona PC (2.5 GHz)	2	250	20,000,000,000	2,097,152	$800	12,500,000,000	$800	95,884,051,042

Fig. 1.12.7

Computers Are Now Pervasive

- Classes of Computers:
  - Personal computers
    - General purpose, variety of software
    - Subject to cost/performance tradeoff
  - Server computers
    - Network based, high capacity, performance, reliability
    - Range from small servers to building sized
  - Supercomputers
    - High-end scientific and engineering calculations
    - Highest capability but represent a small fraction of the overall computer market
  - Embedded computers
    - Hidden as components of systems, special purpose
    - Stringent power/performance/cost constraints

The PostPC Era

- Personal Mobile Device (PMD)
  - Battery operated
  - Connects to the Internet
  - Hundreds of dollars
  - Smart phones, tablets, electronic glasses
- 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
  - Amazon and Google

The PostPC Era

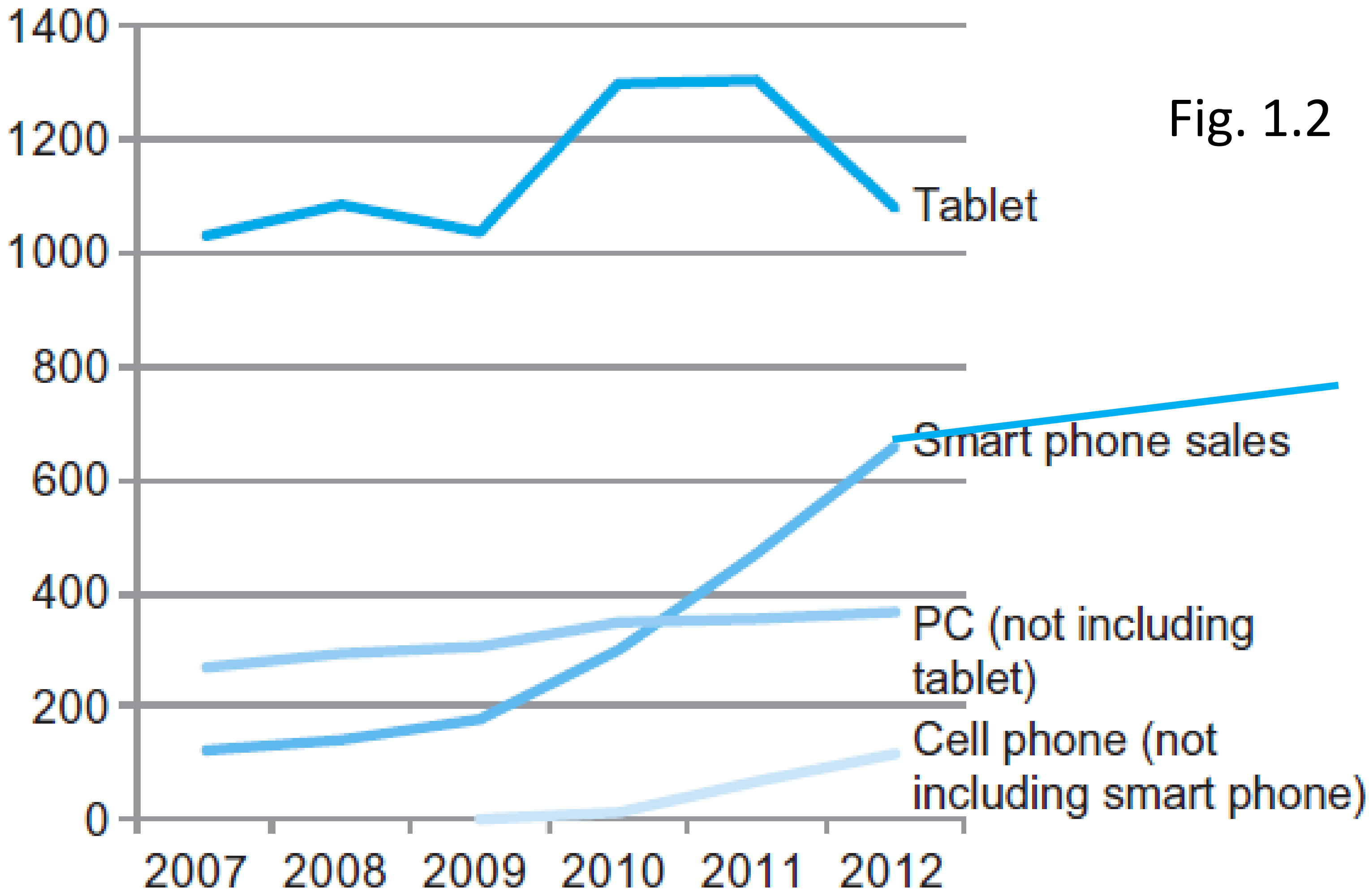


Fig. 1.2

Technology and Computer Summary

- Technology progresses:
  Mechanic → electro-mechanic → electronic
  (vacuum tube → transistor → integrated circuit)

  0-1 switch

  Moore's Law
  - Size ↓
  - Switching speed ↑
  - Reliability ↑
  - Power ↓
  - Cost ↓

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

- Also requires innovative architectural ideas to ride the technology scaling for computer advances

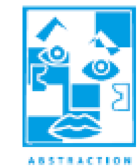
Outline

- Computer: a historical perspective (Sec. 1.1, 1.5)
  - Technology affects computer development
  - Great ideas in computer architecture require technology
- Great ideas in computer architecture (Sec. 1.2)
- Below your program (Sec. 1.3)
- Under the covers (Sec. 1.4)
- Technologies for building processors and memory (Sec. 1.5)
- Performance (Sec. 1.6)
- The power wall (Sec. 1.7)
- From uniprocessors to multiprocessors (Sec. 1.8)

Eight Great Ideas in Computer Architecture

- Design for ***Moore's Law***
  - Very long computer design time → must anticipate the technology when the design finishes rather than starts

- Use ***abstraction*** to simplify design (next page)
  - To improve productivity → represent design with different levels and hide lower-level details with simpler models

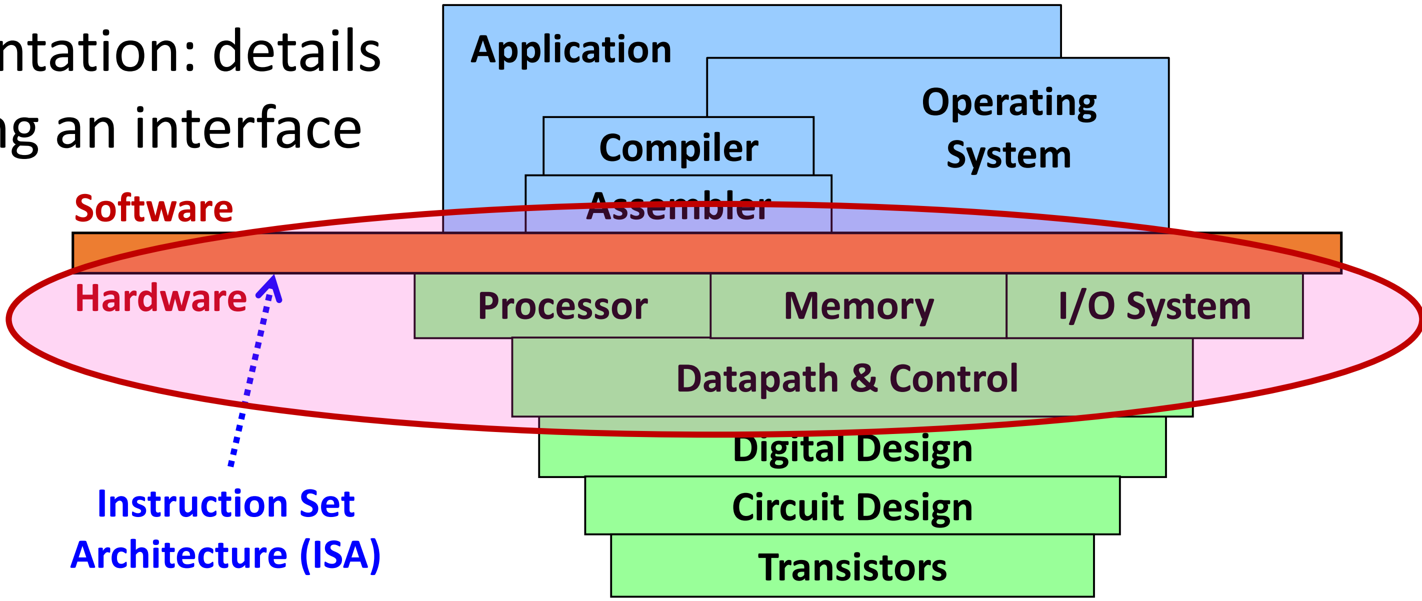


- Make the ***common case fast***
  - To make the most of limited resources, but need to know what the common case is → qualitative approach

- Performance via ***parallelism***
  - To improve throughput by doing things in parallel, but need to throw in multiple resources

An Example of Abstraction

- Abstraction helps us deal with complexity
  - Hide lower-level detail
- Recall instruction set architecture (ISA)
  - ISA: the hardware/software interface
  - Application binary interface: ISA + system SW interface
  - Implementation: details underlying an interface



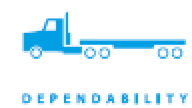
Eight Great Ideas in Computer Architecture

- Performance via ***pipelining***
  - To improve throughput by subdividing a task and doing subtasks in cascade; requiring minimal resources

- Performance via ***prediction***
  - To get head start, if can predict correctly most of time and cheap to recover from misprediction

- Hierarchy of ***memories***
  - To provide an illusion of large, cheap, fast memory by leveraging program/data locality

- Dependability via ***redundancy***
  - Include lower-cost, redundant components that can take over when a failure occurs and to help detect failures



Pipelining

- Factory assembly line vs assembly in one station

Outline

- Computer: a historical perspective
  - Technology affects computer development
  - Great ideas in computer architecture require technology
- Great ideas in computer architecture (Sec. 1.2)
- Below your program (Sec. 1.3)
- Under the covers (Sec. 1.4)
- Technologies for building processors and memory (Sec. 1.5)
- Performance (Sec. 1.6)
- The power wall (Sec. 1.7)
- From uniprocessors to multiprocessors (Sec. 1.8)

Below Your Program

- Application software
  - Written in high-level language
- System software
  - Compiler: translates HLL code to machine code
  - Operating System: service code
    - Handling input/output
    - Managing memory and storage
    - Scheduling tasks & sharing resources
- Hardware
  - Processor, memory, I/O controllers

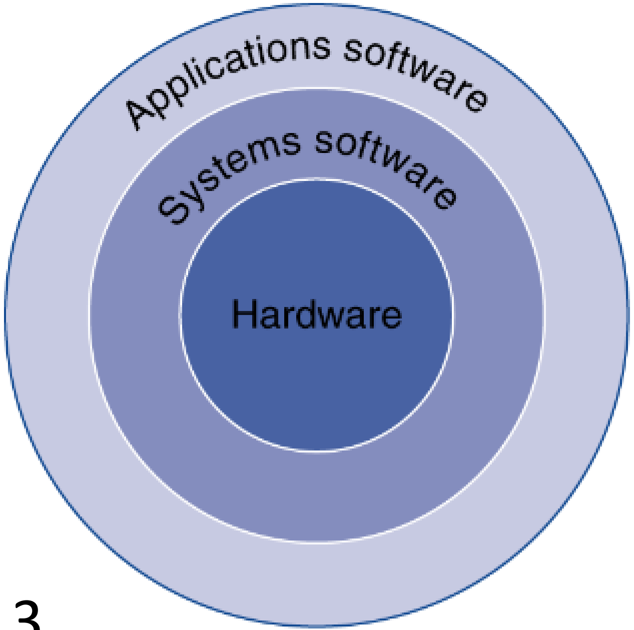


Fig. 1.3

Levels of Program Code

- High-level language
  - Level of abstraction closer to problem domain
  - Provides for productivity and portability
- Assembly language
  - Textual representation of instructions
- Hardware representation
  - Binary digits (bits)
  - Encoded <u>instructions</u> and <u>data</u>

Main focus of ISA

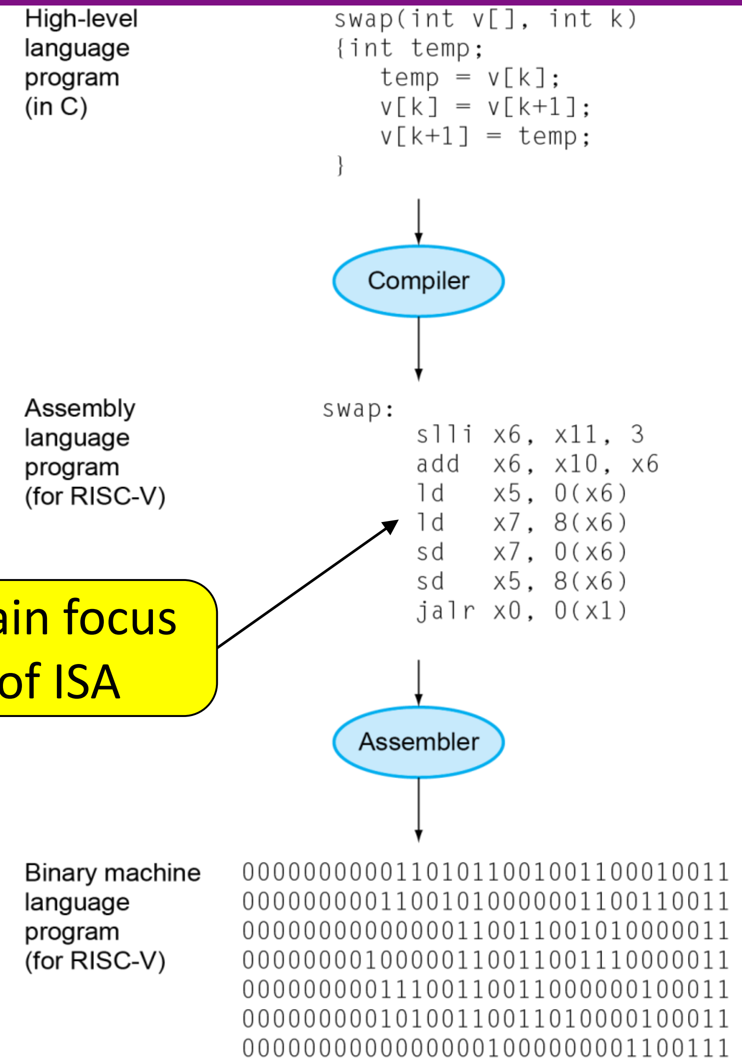


Fig. 1.4

Understanding Performance

- 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

Outline

- Computer: a historical perspective
  - Technology affects computer development
  - Great ideas in computer architecture require technology
- Great ideas in computer architecture (Sec. 1.2)
- Below your program (Sec. 1.3)
- Under the covers (Sec. 1.4)
- Technologies for building processors and memory (Sec. 1.5)
- Performance (Sec. 1.6)
- The power wall (Sec. 1.7)
- From uniprocessors to multiprocessors (Sec. 1.8)

Components of a Computer

- Same components for all kinds of computer
  - Desktop, server, embedded
- Input/output includes
  - User-interface devices
    - Display, keyboard, mouse
  - Storage devices
    - Hard disk, CD/DVD, flash
  - Network adapters
    - For communicating with other computers

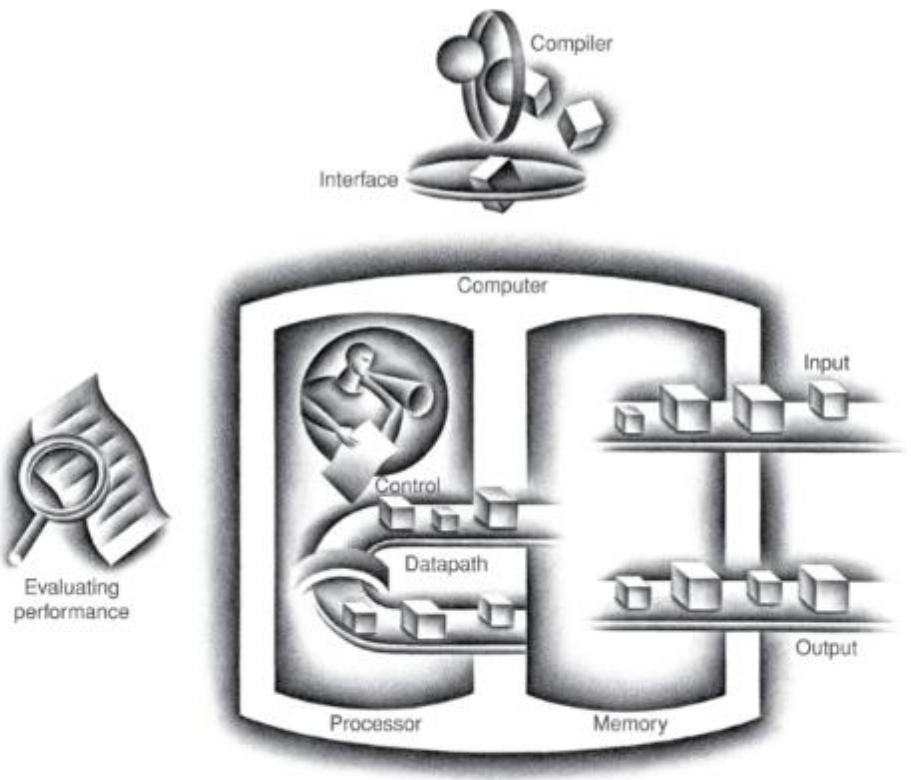
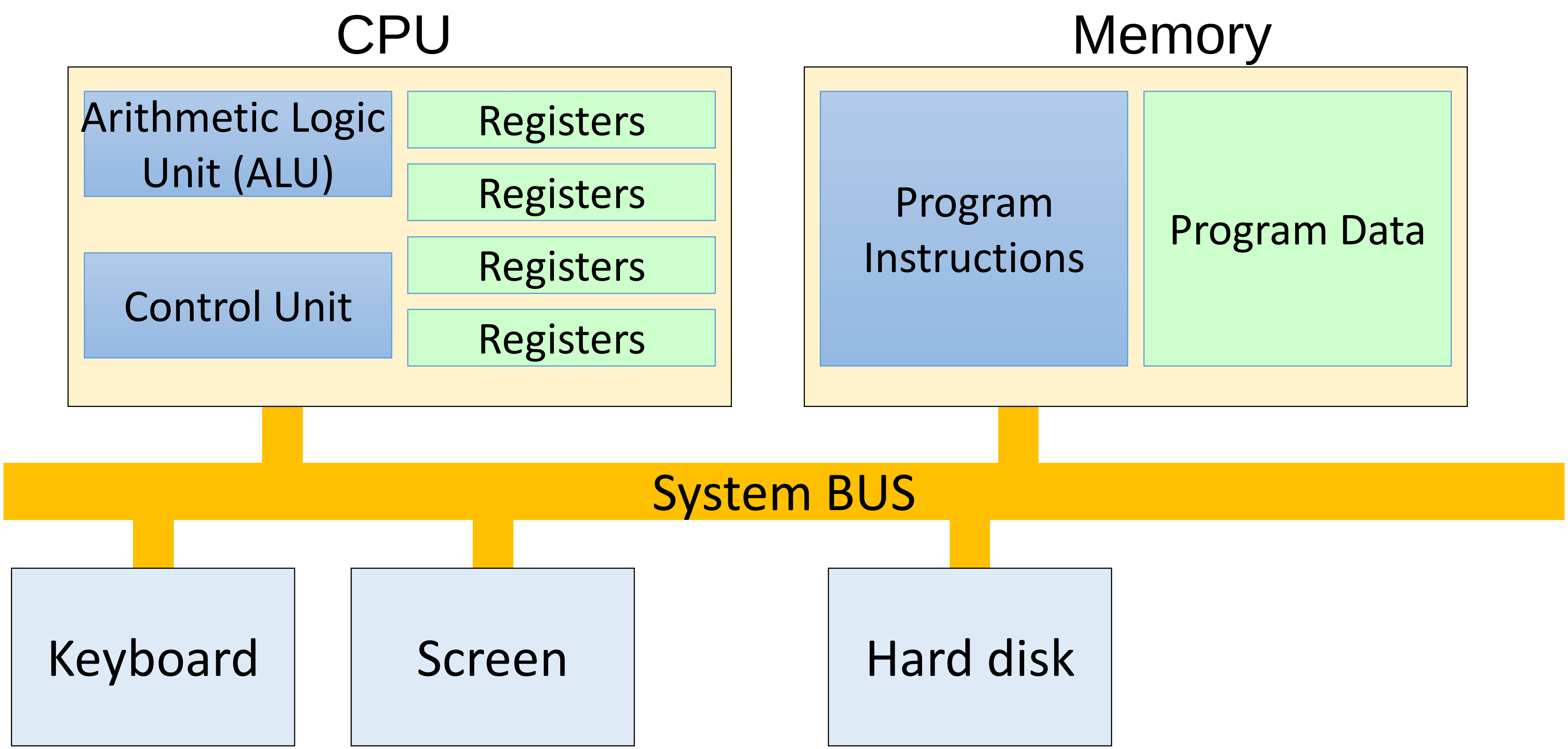


Fig. 1.5

Inside a Computer



(Prof. Jing-Jia Liou)

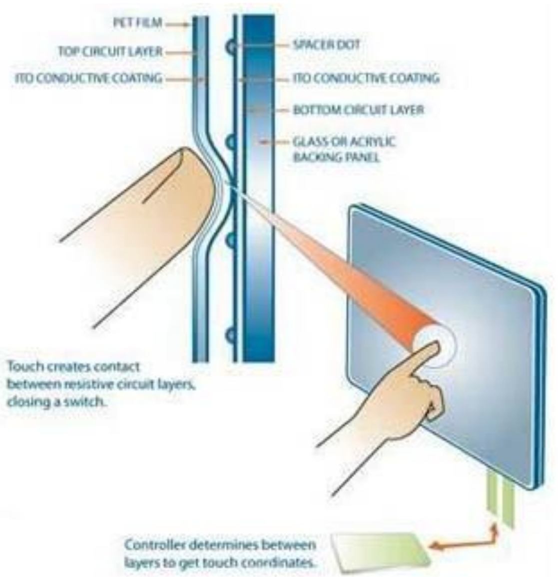
Central Processing Unit (CPU)

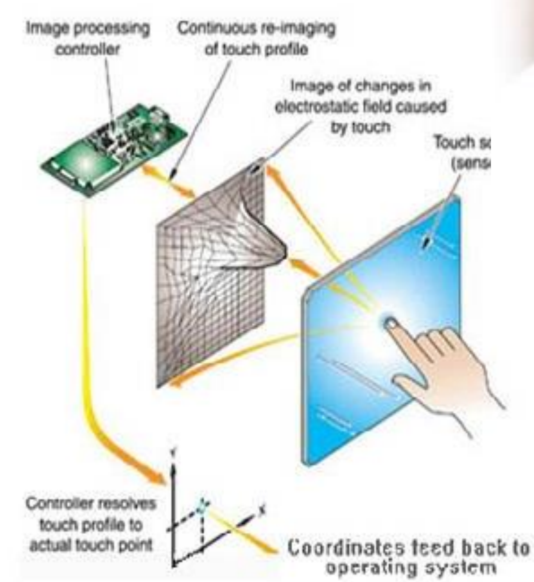
- Control Unit
  - A finite state machine (FSM)
  - Retrieves and decodes program instructions
  - Generate signals to coordinate computer operations: load/store registers, perform ALU functions, take branches, etc…
- Arithmetic & Logic Unit (ALU)
  - Performs mathematical and logical operations

(Prof. Jing-Jia Liou)

Touchscreen

- PostPC device
- Supersedes keyboard and mouse
- Resistive and Capacitive types
  - Tablets, smartphones use capacitive
  - Capacitive allows multi-touch controls





(http://4.bp.blogspot.com/-DFIwFWSWuCc/UO0mRtFp3BI/AAAAAAAAC8Q/nRsDW1I0nBQ/s1600/resistive-touch-screen.jpg)

Through the Looking Glass

- Liquid Crystal Display (LCD): picture elements (*pixels*)
  - Mirrors content of *frame buffer* memory

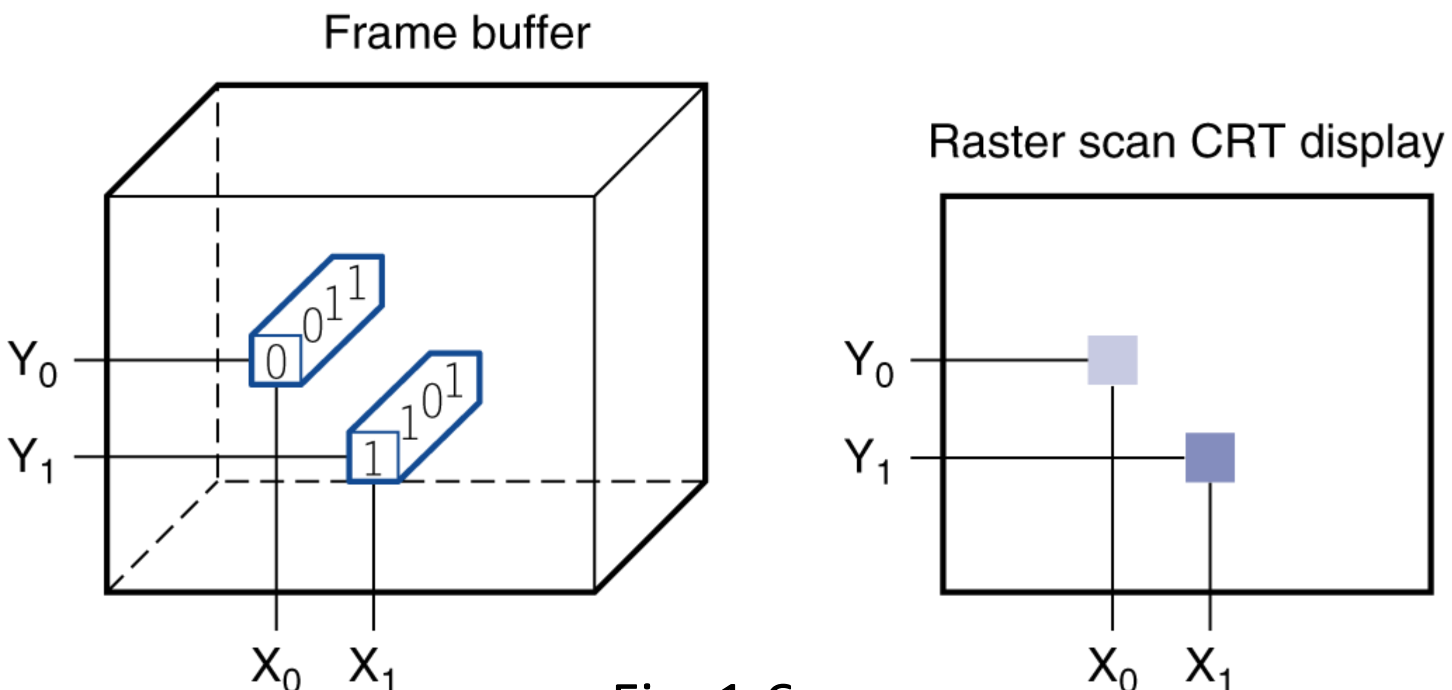


Fig. 1.6

(LCD working principle: http://www.npeducations.com/2012/07/liquid-crystal-display-and-its-internal.html)

Opening the Box (Apple iPad 2)

Fig. 1.7

Capacitive multitouch LCD screen

3.8 V, 25 Watt-hour battery

Computer board:

- Red box: Apple A5
- Orange box: 32 GB flash memory
- Yellow box: power controller and I/O controller chips

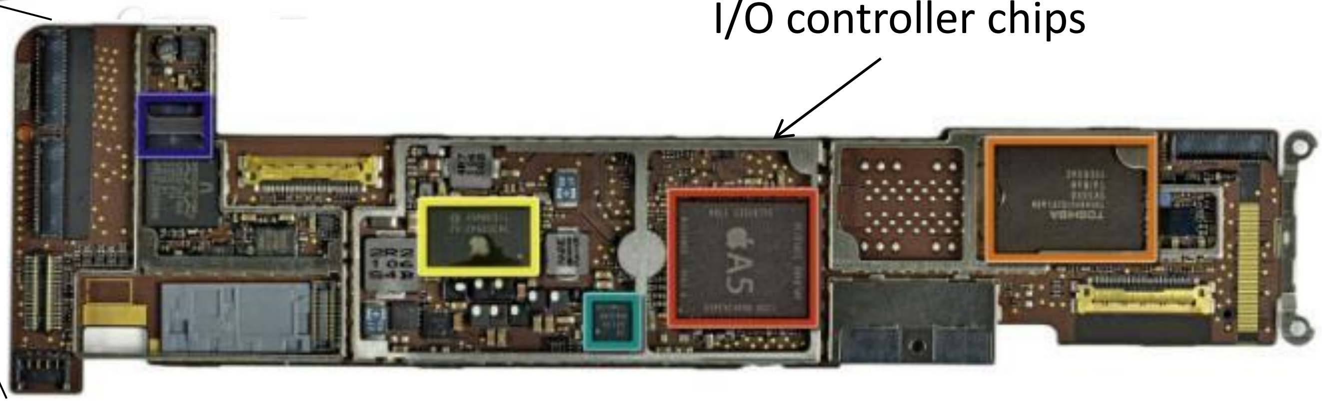
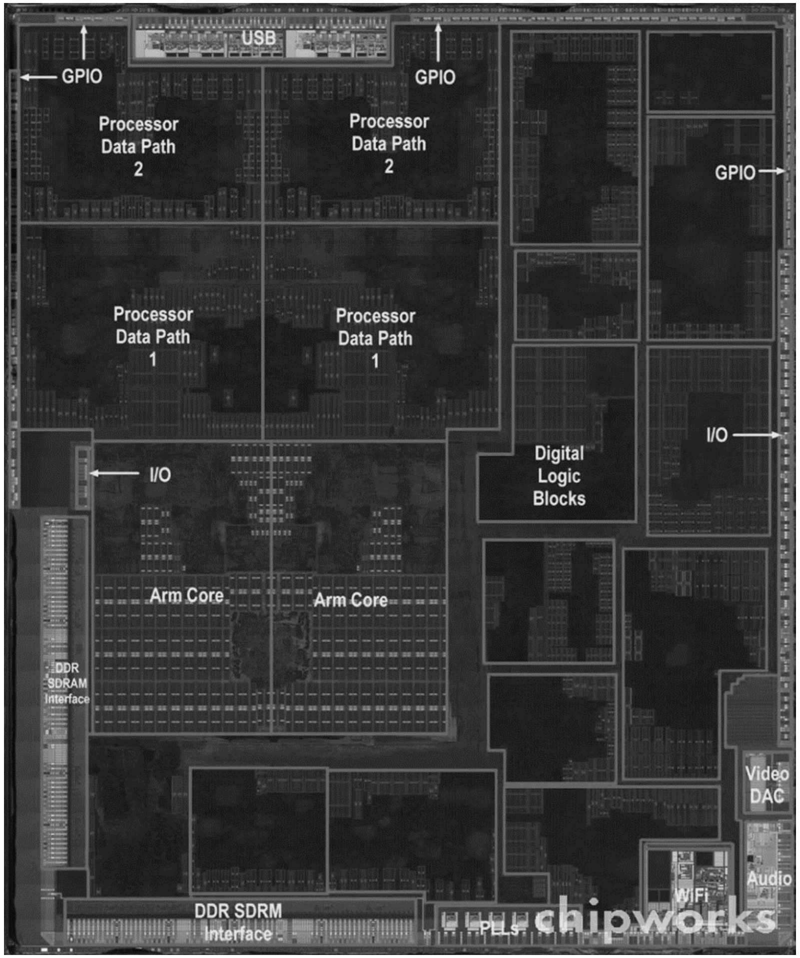


Fig. 1.8

Inside the Processor (CPU)

- Dual ARM cores + GPU
  - Datapath: performs operations on data
  - Control: sequences datapath, memory, ...
- Cache memory
  - Small fast SRAM memory for immediate access to data
- DRAM
  - Dynamic random access memory

Fig. 1.9 Apple A5

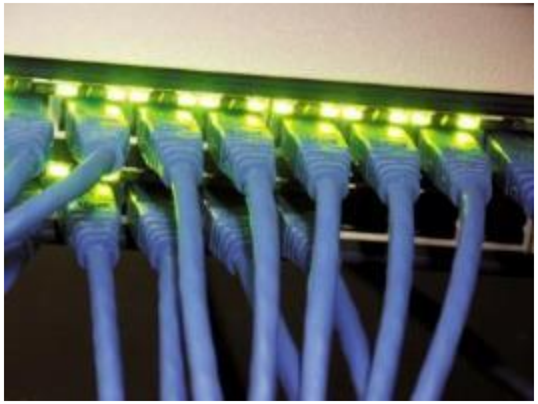


A Safe Place for Data

- Volatile main memory (DRAM)
  - Loses instructions and data when power off
- Non-volatile secondary memory
  - Magnetic disk
  - Flash memory
  - Optical disk (CDROM, DVD)

Networks

- Communication, resource sharing, nonlocal access
- Local area network (LAN): Ethernet
- Wide area network (WAN): the Internet
- Wireless network: WiFi, Bluetooth, wireless broadband



Outline

- Computer: a historical perspective
  - Technology affects computer development
  - Great ideas in computer architecture require technology
- Great ideas in computer architecture (Sec. 1.2)
- Below your program (Sec. 1.3)
- Under the covers (Sec. 1.4)
- Technologies for building processors and memory (Sec. 1.5)
- Performance (Sec. 1.6)
- The power wall (Sec. 1.7)
- From uniprocessors to multiprocessors (Sec. 1.8)

Semiconductor Technology

- IC technology affects computer designs
  - Shapes what computers will be able to do and how quickly they will evolve
- How computer design affects IC implementation?
- Need to understand chip manufacturing
  - **Silicon**: basic element, does not conduct electricity well → **semiconductor**
  - Can add materials to allow tiny areas to transform into one of three devices: conductors, insulators, switches
  - IC manufacturing process is critical to the cost of the chips and hence important to computer designers

Manufacturing ICs

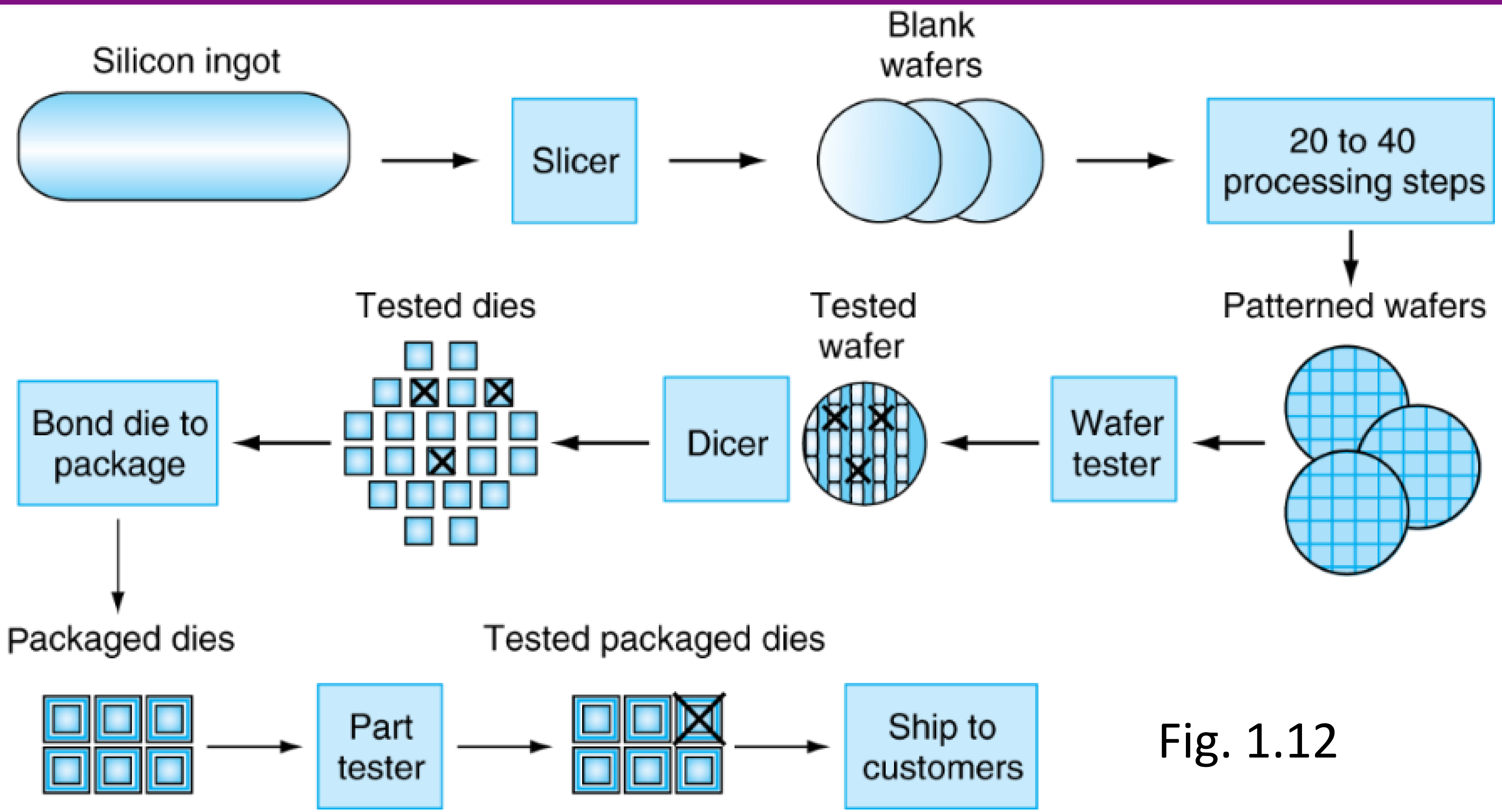


Fig. 1.12

Yield: proportion of working dies per wafer

Intel Core i7 Wafer

- 12-inch (300mm) wafer, 280 chips, 32nm technology
- Each chip is 20.7 x 10.5 mm

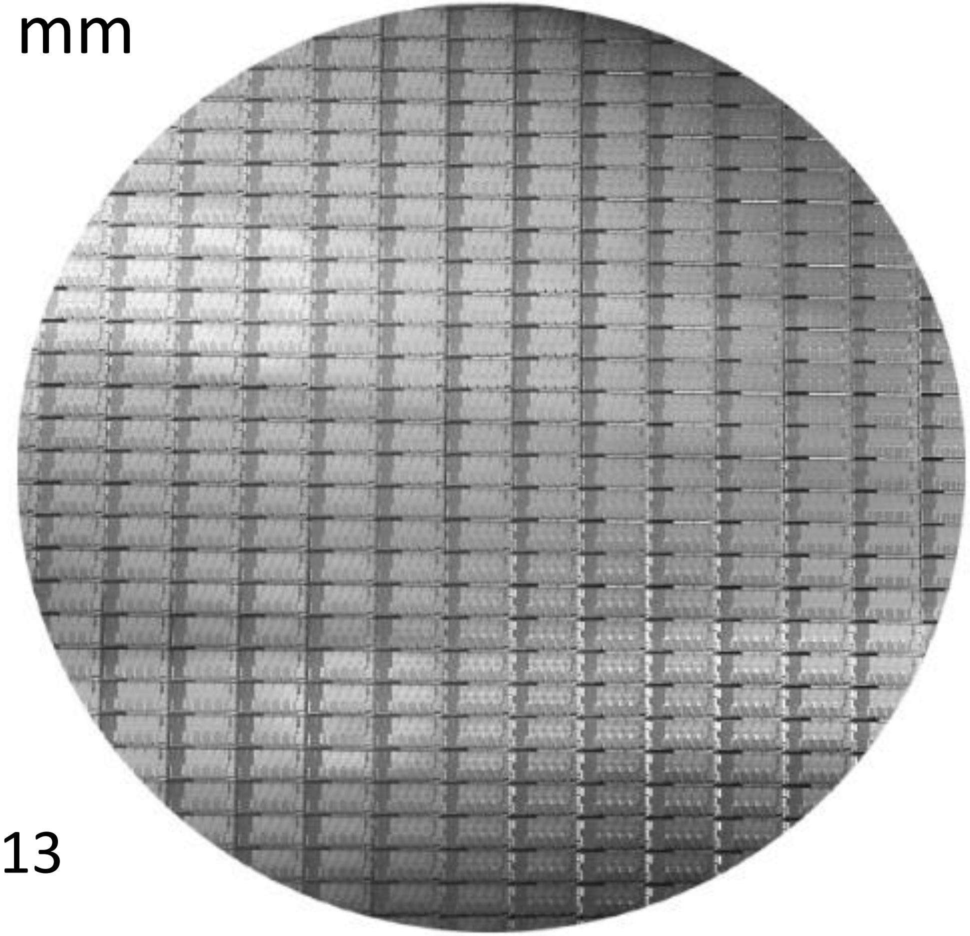


Fig. 1.13

Integrated Circuit Cost

$$\text{Cost per die} = \frac{\text{Cost per wafer}}{\text{Dies per wafer} \times \text{Yield}}$$

$$\text{Dies per wafer} \approx \text{Wafer area/Die area}$$

$$\text{Yield} = \frac{\text{\# of good dies}}{\text{\# of total dies}} = \frac{1}{(1 + (\text{Defects per area} \times \text{Die area/2}))^2}$$

- Nonlinear relation to area and defect rate
  - Wafer cost and area are fixed
  - Defect rate determined by manufacturing process
  - Die area is determined by **architecture** and **circuit design**

Big Picture

Abstraction Layers	Application
	Algorithm
	Software
	ISA
	Microarchitecture
	Circuit
	Transistor

(Prof. Jing-Jia Liou)

Abstractions

- 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 and interface

(Prof. Jing-Jia Liou)