Introduction

Computer Architecture

Dr. Mahmoudi

January 26, 2025

Outline

- About Computer Architecture Course
- Computer Evolution (Optional).
- Computer Components: A review.
- Seven great ideas in computer architecture.
- How Processors Are Made.
- Measuring The Performance
- CISC Vs RISC

ensia

About Computer Architecture Course

Prerequisites

- Digital Systems,
- Data Structures and Algorithms,
- C++ Programming.

Evaluation

- Exam (60%)
- Midterm (20%)
- Tutorial Test (10%)
- Lab (10%) = Test (10 Pts) + Assignments (13×0.75)

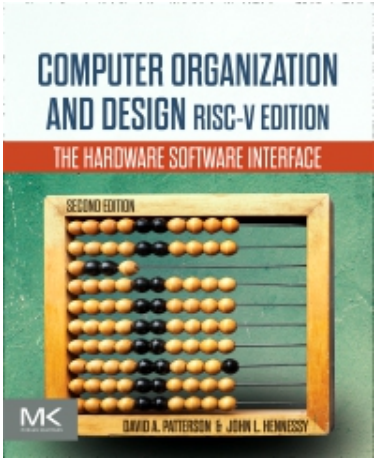
Motivation

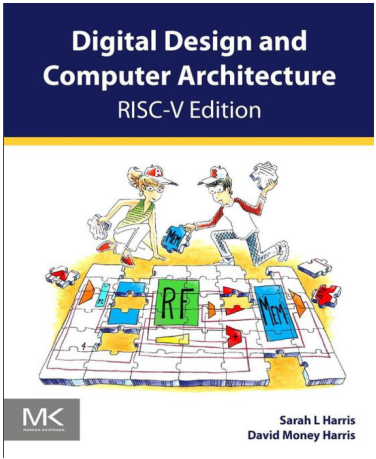
- How are computers made?
- How do computers work?
- Understanding computer architecture makes you a better programmer and CS engineer.
- What does computer architecture do with AI?

Course Content

- Introduction of RISC-V Architecture
- RISC-V Instruction Set
- RISC-V Programming Model
- RISC-V: Processor Design
- Pipelining
- Cache Management
- Virtual Memory
- I/O & interrupts
- Advanced Computer Architecture (Optional)

References


COMPUTER ORGANIZATION
AND DESIGN RISC-V EDITION
THE HARDWARE SOFTWARE INTERFACE
SECOND EDITION
DAVID A. PATTERSON & JOHN L. HENNESSY


Digital Design and
Computer Architecture
RISC-V Edition
RF
MEM
Sarah L Harris
David Money Harris

ensia

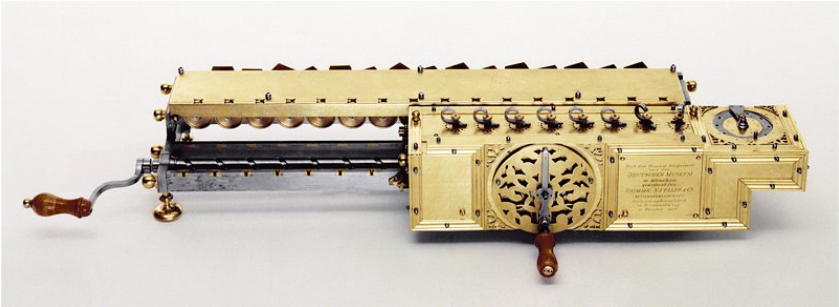
Computer Evolution

The Zeroth Generation—Mechanical Computers (1642–1945)

- The French scientist **Blaise Pascal** (1623–1662), in **1642** built the 1^{st} calculator.
- Powered by a hand-operated crank and does only **addition and subtraction**.
- The programming language Pascal is named in his honor.

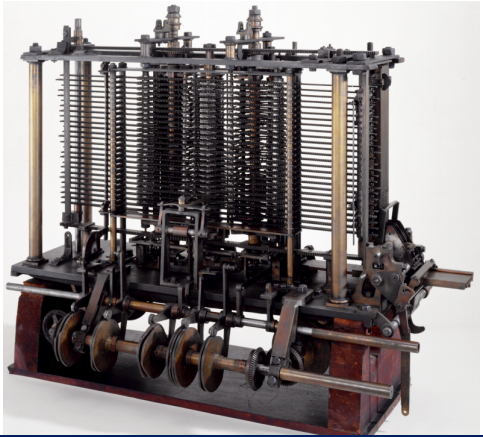
The Zeroth Generation—Mechanical Computers (1642–1945)

- Stepped reckoned by the German mathematician **Baron Gottfried Wilhelm von Leibniz** (1646–1716) **1673**.
- It could multiply and divide also.



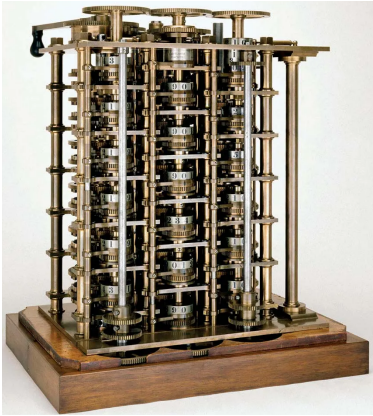
The Zeroth Generation—Mechanical Computers (1642–1945)

- **Charles Babbage** (1792–1871) built his **Difference Engine**.
- **Could run only one algorithm to add and subtract**
- It punched its results into a copper engraver's plate with a steel die (idea: CD-ROM).



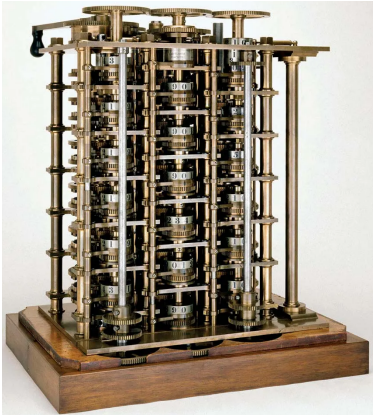
The Zeroth Generation—Mechanical Computers (1642–1945)

- **Charles Babbage's Analytical Engine**.
- It had four components:
  1. **Store** (memory): 1000 words of 50 decimal digits;
  2. **Mill** (computation unit): add, subtract, multiply, divide;
  3. **Punched-card reader**: data + instructions;
  4. **Punched and printed output** (output).



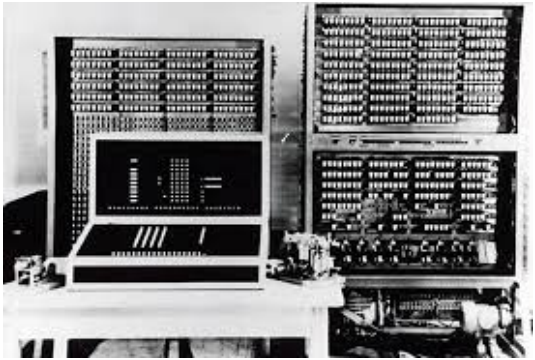
The Zeroth Generation—Mechanical Computers (1642–1945)

- Punching differents programs on the input cards.
- Babbage hired a young woman named **Augusta Ada Lovelace**.
- **The world's first computer programmer**.
- The programming language **Ada** is named in her honor.



The Zeroth Generation—Mechanical Computers (1642–1945)

- **Late 1930s**, a German engineering student **Konrad Zuse** built a series of automatic calculating machines using electromagnetic relays.
- Destroyed by the Allied bombing of Berlin in 1944 (WWII).



The Zeroth Generation—Mechanical Computers (1642–1945)

- **John Atanasoff** at Iowa State College and **George Stibbitz** at Bell Labs
- It used binary arithmetic and had capacitors for memory (DRAM ancestor).

The Zeroth Generation—Mechanical Computers (1642–1945)

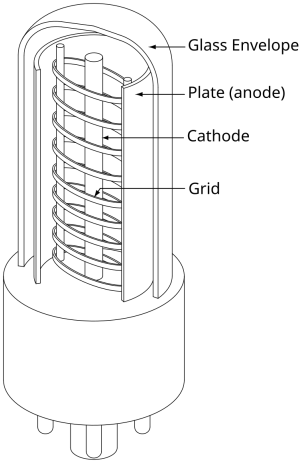
- **Howard Aiken** at Harvard built **Mark** I in **1944**.
- It had 72 words of 23 decimal digits each and had an instruction time of 6 sec.
- Input and output used punched paper tape.

The Zeroth Generation—Mechanical Computers (1642–1945)

- During WW II, the Germans built a machine called **ENIGMA** to encrypt the messages sent from the German admirals in Berlin to the submarines by radio.

The First Generation—Vacuum Tubes (1945–1955)

- A **vacuum tube** (**valve** in British English) is an electronic device used to control electric current flow.
- **Thermionic emission**: The cathode is heated to emit electrons.
- **Anode**: the part that accepts the emitted electrons.



The First Generation—Vacuum Tubes (1945–1955)

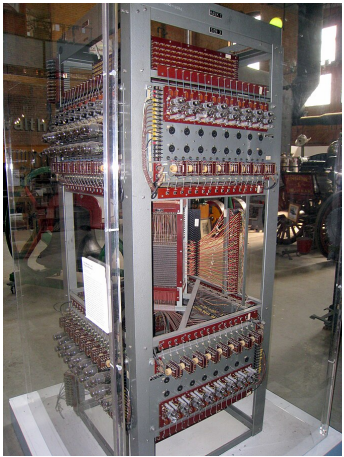
- To decode **Enigma** messages, the British built an electronic computer called the **COLOSSUS, in 1943**.
- The famous British mathematician **Alan Turing** helped design this machine.
- First electronic computer.

The First Generation—Vacuum Tubes (1945–1955)

- **1943-1946, John Mauchley and J. Presper Eckert** built the **ENIAC (Electronic Numerical Integrator And Computer)**.
- 18,000 vacuum tubes, 20 registers holding a 10-digit decimal number.
- **EDSAC (1949)**: The first operational electronic computers built at the University of Cambridge by **Maurice Wilkes**.
- Then, **EDVAC (Electronic Discrete Variable Automatic Computer)** by Eckert & Mauchley.

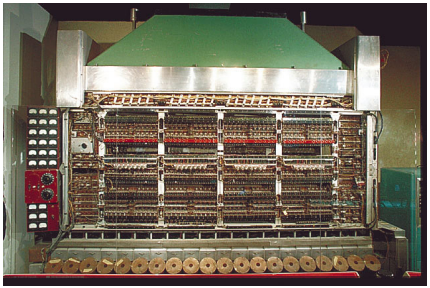
The First Generation—Vacuum Tubes (1945–1955)

- In **1951**, M.I.T. researchers built **Whirlwind I**
- 16-bit word and was designed for real-time control.
- Led to the invention of the magnetic core memory by **Jay Forrester**
- First commercial minicomputer.



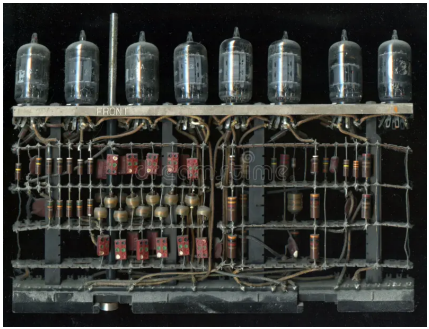
The First Generation—Vacuum Tubes (1945–1955)

- **John von Neumann**, went to Princeton's Institute of Advanced Studies to build his own version of the EDVAC, the **IAS machine**.
- Stored-program computer.
- The memory = 4096 words of 40 bits.
- Each word held either two 20-bit instructions or a 40-bit signed integer.
- Instruction = 8 bits for type and 12 bits for address.



The First Generation—Vacuum Tubes (1945–1955)

- **1953, IBM** built the **701** with **2048 36-bit words**, with two instructions per word.
- **1956, 704** with 4096 words and 36-bit instructions.
- **1958** last vacuum-tube machine, the **709**.



The First Generation—Vacuum Tubes (1945–1955)

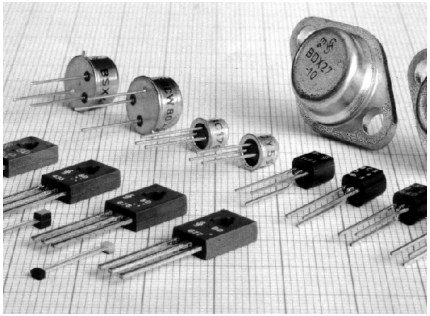
Year	Name	Made by	Comments
1943	COLOSSUS	British government	First electronic computer
1946	ENIAC	Eckert/Mauchley	Modern computer history starts here
1949	EDSAC	Wilkes	First stored-program computer
1951	Whirlwind I	M.I.T.	First real-time computer
1952	IAS	Von Neumann	Most current machines use this design

Table: First Generation Milestones

For a list of vacuum-tube computers go **here**.

The Second Generation—Transistors (1955–1965)

- **Transistor** An on/off switch controlled by an electric signal.
- A semiconductor device used to amplify or switch electrical signals and power.
- Invented at **Bell Labs** in **1948** by **John Bardeen, Walter Brattain, and William Shockley** (1956 Nobel Prize in physics).



The Second Generation—Transistors (1955–1965)

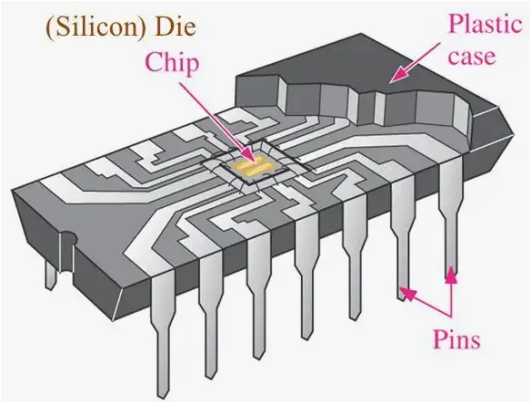
- **DEC** (Digital Equipment Corporation), built **PDP-1** in **1961**, with **4096 18-bit words** and could execute **200,000** instructions/sec.
- Visual display plots points on 512-by-512 pixel screen.
- M.I.T students programmed **Spacewar** the world had its first video game.
- **PDP-8** (12-bit) with a major innovation: a single bus (**omnibus**).

The Second Generation—Transistors (1955–1965)

- **IBM** built a transistorized version of the 709, the **7090**and later the **7094**.
- The 7094 had a cycle time of **2 micro secs** and a core memory of **32,768 words of 36 bits**.
- IBM became a major force in scientific computing with the 7094 and small businesses with a machine called the **1401**.

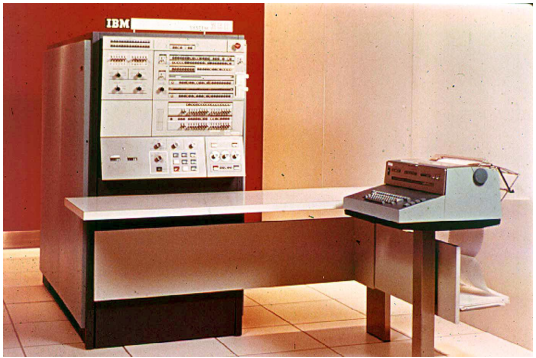
The Third Generation—Integrated Circuits (1965–1980)

- **Integrated circuit**, nicknamed **chips**, combined dozens to hundreds of transistors into a single chip.
- Invented by **Jack Kilby** and **Robert Noyce** (working independently) in **1958**.
- Computers became smaller, faster, and cheaper.



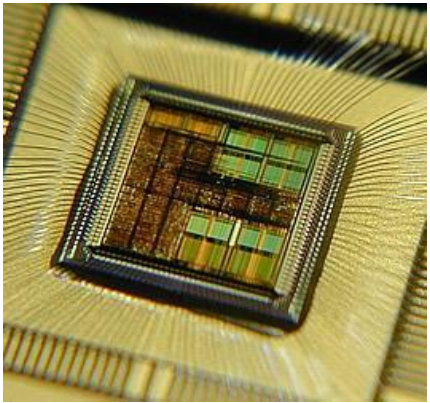
The Third Generation—Integrated Circuits (1965–1980)

- **System/360** by **IBM** was a family of computers with the same assembly language, increasing size and power.
- **multiprogramming**: several programs in memory at once.
- later followed by the 370, 4300, 3080, 3090, 390 and z series.
- **DEC**'s PDP-11



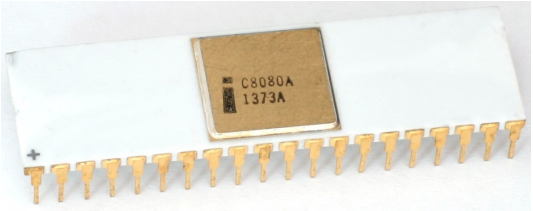
The Fourth Generation—Very Large Scale Integration (1980–?)

- **very large-scale integrated (VLSI) circuit** A device containing hundreds of thousands to millions of transistors.
- The personal computer era had begun.



The Fourth Generation—Very Large Scale Integration (1980–?)

- **Intel 8080**: First general-purpose computer on a chip.
- **CP/M** by **Gary Kildall** (floppy) disk operating system for 8080, with a file system, and user commands typed in from the keyboard to a command processor (shell).
- **Apple** and later the **Apple II**, designed by **Steve Jobs** and **Steve Wozniak**.



The Fourth Generation—Very Large Scale Integration (1980–?)

Year	Name	Made by	Comments
1974	8080	Intel	First general-purpose 8-bit computer on a chip
1981	IBM PC	IBM	Started the modern personal computer era
1981	Osborne-1	Osborne	First portable computer
1983	Lisa	Apple	First personal computer with a GUI
1985	386	Intel	First 32-bit ancestor of the Pentium line
1985	MIPS	MIPS	First commercial RISC machine
1987	SPARC	Sun	First SPARC-based RISC workstation
1992	Alpha	DEC	First 64-bit personal computer
2001	POWER4	IBM	First dual-core chip multiprocessor

Table: Fourth Generation Milestones

The Fifth Generation—Low-Power and Invisible Computers

- In **1989, Grid Systems** released the first tablet computer, the **GridPad**.
- The **Apple Newton**, released in **1993**.
- **1992, IBM** built first smartphone called **Simon**.
- These **PDAs (Personal Digital Assistants)** evolved into **smartphones**.

The Fifth Generation—Low-Power and Invisible Computers

- **invisible** computers are embedded into appliances, watches, bank cards, and numerous other devices.
- Hardware and software are often **co designed**.
- the real fifth generation is more a paradigm shift than a specific new architecture.
- This model, devised by **Mark Weiser**, was originally called **ubiquitous computing**, but the term **pervasive computing** is also used frequently now.

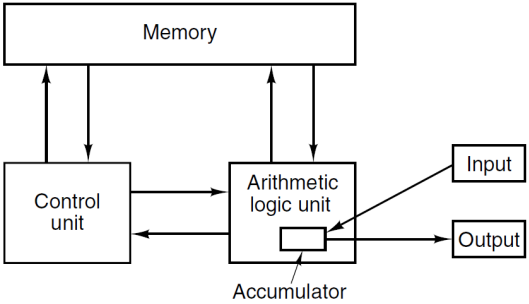
ensia

Computer Components

A Review

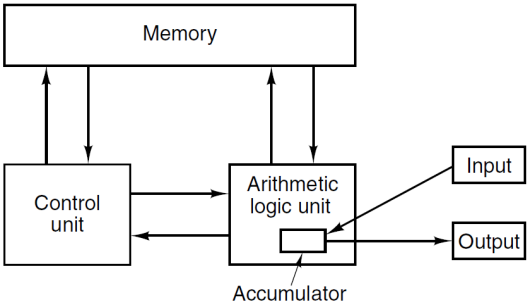
Von Neumann Architecture

- In 1945, John Von Neumann published an idea for a new computer, the EDVAC (Electronic Discrete Variable Computer).
- It was about the design of a new stored-program computer, referred to as the IAS computer (Institute for Advanced Study).



Von Neumann Architecture

- **Memory**, which stores both data and instructions.
- **Arithmetic and logic unit (ALU)** (CA: Central Arithmetical) capable of operating on binary data.
- **Control unit**, which interprets the instructions in memory and causes them to be executed.
- **Input–output (I/O)** equipment operated by the control unit.

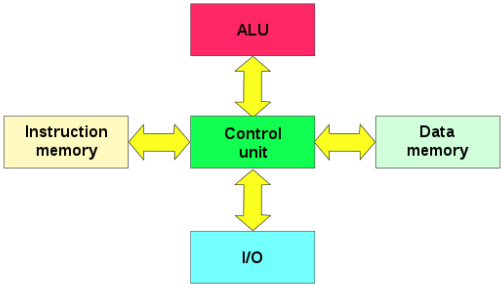


The von Neumann bottleneck

- The shared bus between the program memory and data memory leads to the von Neumann bottleneck.
- Because the single bus can only access one of the two classes of memory at a time, throughput is lower than the rate at which the CPU can work.
- This seriously limits the effective processing speed when the CPU is required to perform minimal processing on large amounts of data.
- The CPU is continually forced to wait for needed data to move to or from memory.
- Since CPU speed and memory size have increased much faster than the throughput between them, the bottleneck has become more of a problem, a problem whose severity increases with every new generation of CPU.

Harvard Architecture

- The Harvard architecture is a computer architecture with separate storage and signal pathways for instructions and data.
- It is often contrasted with the von Neumann architecture.
- The term is often stated as having originated from the Harvard Mark I relay-based computer.

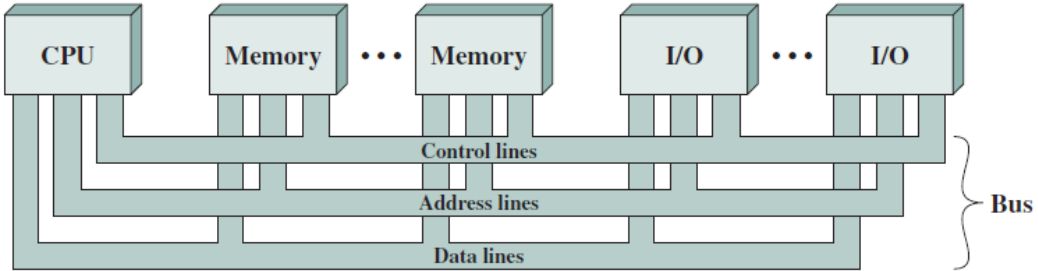


Central Processing Unit (CPU)

- The **central processing unit (CPU)**, also called **processor**, is the active part of the computer.
- The processor logically comprises two main components: **datapath** and **control** (**brawn and brain**).
  1. **Datapath** Components of the processor that perform arithmetic operations. It includes the memory, registers, adders, ALU, and communication buses. It is called data path because data follows a specific path when executing an instruction.
  2. **Control** The component of the processor that commands the datapath according to the instructions of the program. It sets up dataflow directions on communication buses and selects ALU and memory functions. Control signals are generated by a control unit of one or more finite-state machines.

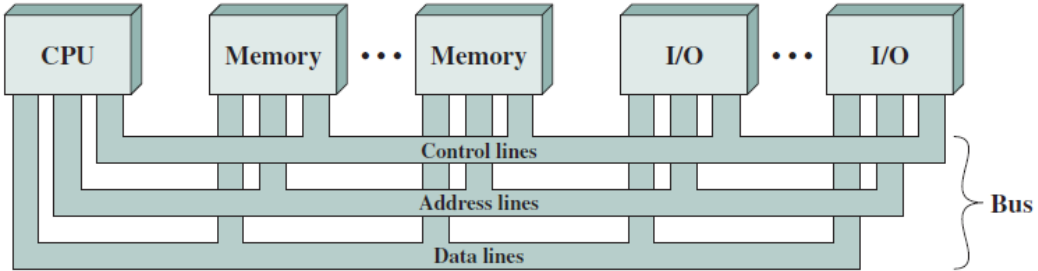
Bus Interconnection

- A bus is a communication pathway connecting two or more devices.
- A shared transmission medium.
- If two devices transmit during the same time period, their signals will overlap and become garbled.



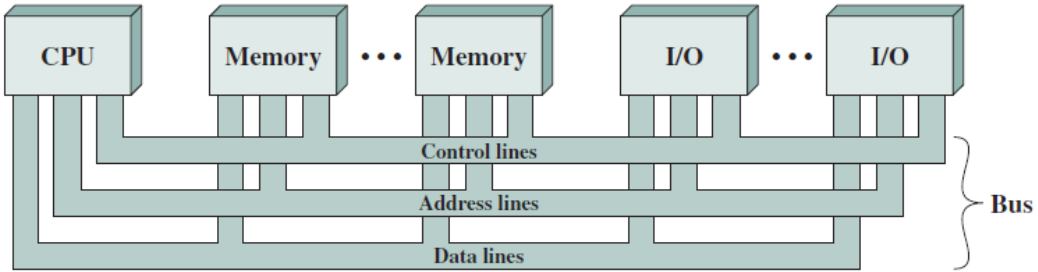
Bus Interconnection

- The **data lines** provide a path for moving data among system modules.
- These lines, collectively, are called the **data bus**.
- The number of lines is referred to as the width of the data bus.



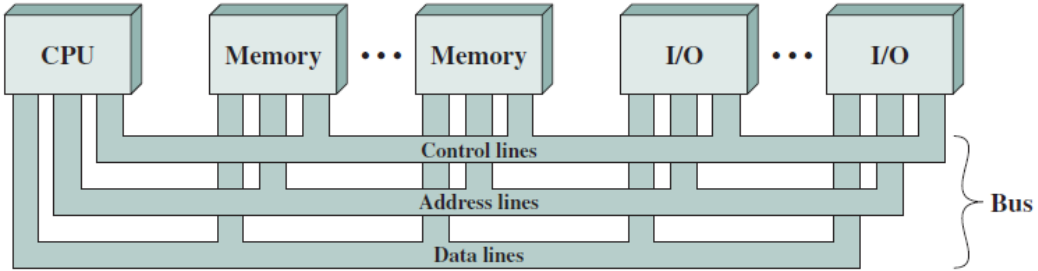
Bus Interconnection

- The **address lines** are used to designate the source or destination of the data on the data bus.
- The width determines the maximum possible memory capacity of the system.
- The higher-order bits are used to select a particular module on the bus, and the lower-order bits select a memory location or I/O port within the module.



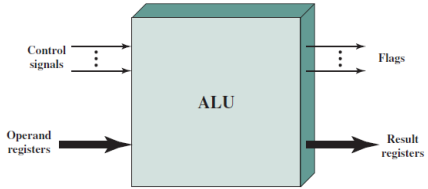
Bus Interconnection

- The **control lines** are used to control the access to and the use of the data and address lines.
- They transmit both command and timing information among system modules.
- Timing signals indicate the validity of data and address information.
- Command signals specify operations to be performed.



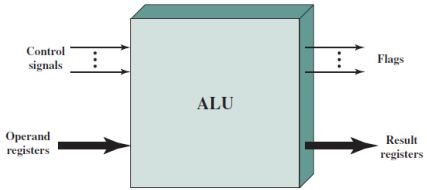
Arithmetic and Logic Unit (ALU)

- Operands for arithmetic and logic operations are presented to the ALU in registers, and the results of an operation are stored in registers.
- These registers are temporary storage locations within the processor that are connected by signal paths to the ALU .
- The ALU may also set flags as the result of an operation.
- For example, an overflow flag is set to 1 if the result of a computation exceeds the length of the register into which it is to be stored.



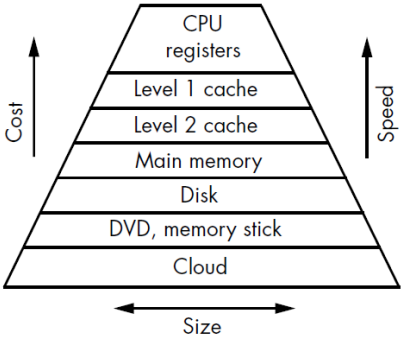
Arithmetic and Logic Unit (ALU)

- The flag values are also stored in registers within the processor.
- The processor provides signals that control the operation of the ALU and the movement of the data into and out of the ALU.



The Memory Hierarchy

- In general, the closer memory is to the CPU, the faster and more expensive it is.
- The slowest memory is the cloud. It's also the least expensive.
- Memory within the CPU runs at the same speed as the CPU but is relatively expensive.
- The top three layers are typically included in the CPU chip in modern computers.
- There may be one or two more levels of cache before getting to the main memory.



Registers

- The fastest memory is within the CPU itself: the registers.
- Registers typically provide about 1KB of storage and are accessed at the same speed as the CPU.
- They’re mainly used for numerical computations, logical operations, temporary data storage, and similar short-term operations.
- Many registers are directly accessible by the programmer, while others are hidden.
- Some are used in the hardware that serves to interface between the CPU and I/O devices.
- The organization of registers in the CPU is very specific to the particular CPU architecture

Registers

- **Program Counter** A program counter (PC) is a CPU register in the computer processor that has the address of the next instruction to be executed from memory.
- As each instruction gets fetched, the program counter increases its stored value by 1.
- It is a digital counter needed for faster execution of tasks as well as for tracking the current execution point.
- **Instruction Register (IR)** is the part of a CPU's control unit that holds the instruction currently being executed or decoded.
- The instruction register specifically holds the instruction and provides it to the instruction decoder circuit.

Registers

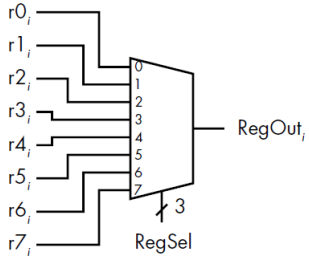
- **Memory Address Register (MAR)** is the CPU register that either stores the memory address from which data will be fetched from the CPU, or the address to which data will be sent and stored.
- It is a temporary storage component in the CPU that temporarily stores the address (location) of the data sent by the memory unit until the instruction for the particular data is executed.

Registers

- **Memory Data Register (MDR)** is the register that stores the data being transferred to and from the immediate access storage.
- Memory data register (MDR) is also known as memory buffer register (MBR).
- **General Purpose Register** General-purpose registers are used to store temporary data within the microprocessor.
- They can be used either by a programmer or by a user.

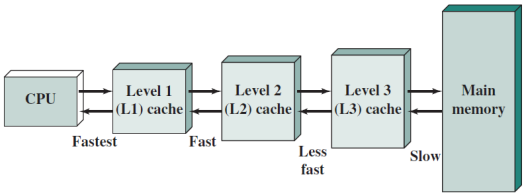
Register File

- The registers in the CPU that are used for similar operations are grouped into a register file.
- We connect an 8-to-3 multiplexer to each corresponding bit of the eight registers.
- The inputs to the multiplexer, $r0_i$–$r7_i$, are the i^{th} bits from each of eight registers, $r0$–$r7$.
- A four-bit register would need four of these multiplexer output circuits.
- The same RegSel would be applied to all four multiplexers simultaneously to output all four bits of the same register.



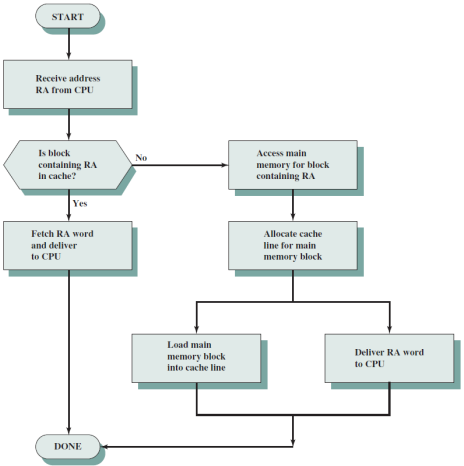
Cache Memory

- Most modern computers include very fast **cache memory** between the main memory and the CPU.
- It provides a much faster location for the instructions and variables currently being processed by the program.
- Cache memory is organized in levels, with Level 1 being the closest to the CPU, and the smallest.
- The amount of memory copied into a cache at a time is called a line.



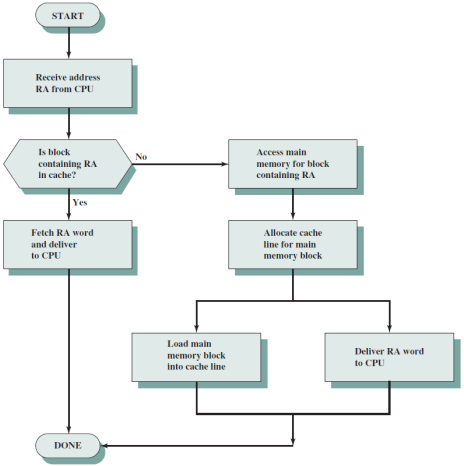
Cache Memory

- When a program needs to access an instruction or data item, the hardware first checks if it is located in the L1 cache.
- If not, it checks the Level 2 cache.
- If it is there, the hardware copies a block of memory that includes it into L1 cache and then into the CPU.



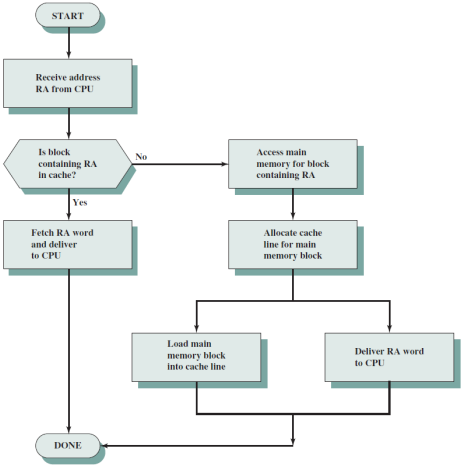
Cache Memory

- If it is not in the L2 cache, the hardware checks the L3 cache.
- If it finds it, it copies it into Level 2, then Level 1, and from there into the CPU.
- If the data is not in L3, the hardware checks the main memory.



Cache Memory

- In this way, the hardware makes a copy of the portion of the program within the L3 cache, a smaller portion of what it's working on in the Level 2 cache, and an even smaller portion in the Level 1 cache.
- When data is written to main memory, it starts with Level 1 cache, then the next cache levels, and finally into main memory.



Main Memory

- Main memory communicates with the CPU using the data, address, and control buses.
- Usually, the entire program and dataset are not loaded into main memory.
- Only the portion currently being worked on is loaded by the operating system from mass storage into main memory.
- Most mass storage devices in modern computers can be accessed only in blocks of predetermined size.

Main Memory

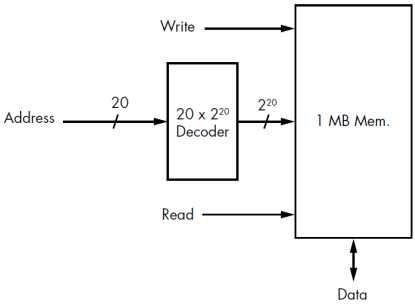
- When a needed instruction or data item is loaded into main memory, the computer loads the whole block of instructions or data that includes the needed item into memory.
- Chances are good that the nearby parts of the program (instructions or data) will be needed soon.
- Since they’re already in main memory, the operating system doesn’t need to access the mass storage device again, thus speeding up program execution

Dynamic RAM (DRAM)

- A dynamic RAM (DRAM) is made with cells that store data as a charge on capacitors.
- The presence or absence of charge in a capacitor is interpreted as a binary 1 or 0.
- Because capacitors have a natural tendency to discharge, dynamic RAMs require periodic charge refreshing to maintain data storage.
- The term dynamic refers to this tendency of the stored charge to leak away, even with power continuously applied.

Static Random-Access Memory

- The memory that uses flip-flops is called static random-access memory (SRAM).
- It's called static because it maintains its values so long as power is maintained.
- It's called random because it takes the same amount of time to access any (random) byte in this memory.
- SRAM is commonly used for cache memory.

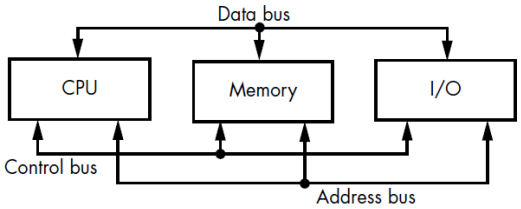


SRAM versus DRAM

- Both static and dynamic RAMs are volatile;
- A dynamic memory cell is simpler and smaller than a static memory cell. Thus, a DRAM is more dense (smaller cells = more cells per unit area) and less expensive than a corresponding SRAM.
- On the other hand, a DRAM requires the supporting refresh circuitry.
- For larger memories, the fixed cost of the refresh circuitry is more than compensated for by the smaller variable cost of DRAM cells. Thus, DRAMs tend to be favored for large memory requirements.
- A final point is that SRAMs are somewhat faster than DRAMs. Because of these relative characteristics, SRAM is used for cache memory, and DRAM is used for main memory.

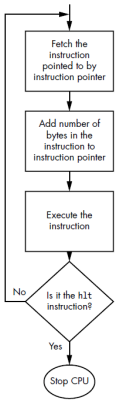
Mass Storage

- Mass storage is used for keeping programs and large amounts of data in a machine-readable format.
- This includes hard disks, solid-state drives, memory sticks, optical disks, etc.
- Their contents are nonvolatile,



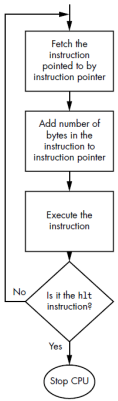
Instruction Execution Cycle

- A sequence of instructions is stored in memory.
- The memory address where the first instruction is located is copied to the PC.
- The CPU sends the address in the PC to memory on the address bus.
- The CPU sends a “read” signal on the control bus.



Instruction Execution Cycle

- The memory responds by sending a copy of the state of the bits at that memory location on the data bus, which the CPU then copies into its instruction register.
- The PC is automatically incremented to contain the address of the next instruction in memory.
- The CPU executes the instruction in the instruction register.
- Go back to step 3.



Seven great ideas in computer architecture

- **Use Abstraction to Simplify Design:** Lower-level details are hidden to offer a simpler model at higher levels.
- **Make the Common Case Fast**
- **Performance via Parallelism**
- **Performance via Pipelining:** A particular pattern of parallelism.
- **Performance via Prediction:** ”It can be better to ask for forgiveness than to ask for permission”.
- **Hierarchy of Memories**
- **Dependability via Redundancy:** we make systems dependable by including redundant components that can take over when a failure occurs and to help detect failures

ensia

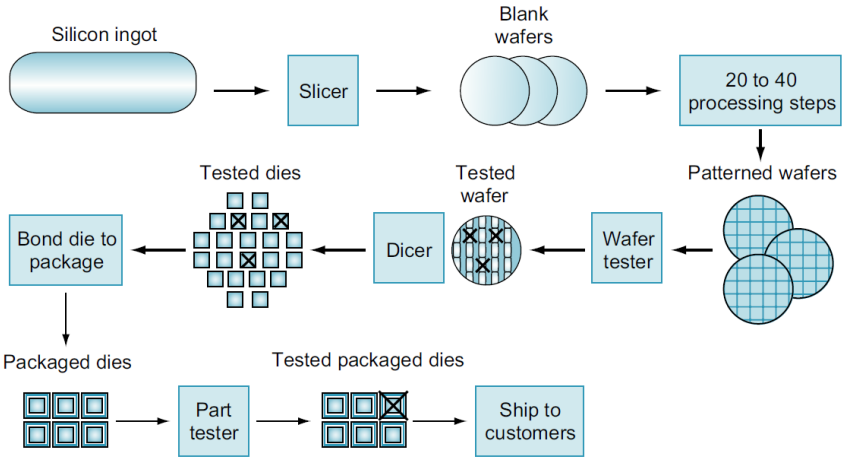
How Processors Are Made

Making a Chip

- The chip manufacture begins with silicon, a substance found in sand.
- Because silicon does not conduct electricity well, it is called a semiconductor.
- With a special chemical process, it is possible to add materials to silicon that allow tiny areas to transform into one of three devices:
  1. Excellent conductors of electricity (using either microscopic copper or aluminum wire);
  2. Excellent insulators from electricity (like plastic sheathing or glass);
  3. Areas that can conduct or insulate under specific conditions (transistors)
- A VLSI circuit is billions of combinations of conductors, insulators, and switches manufactured in a single small package.

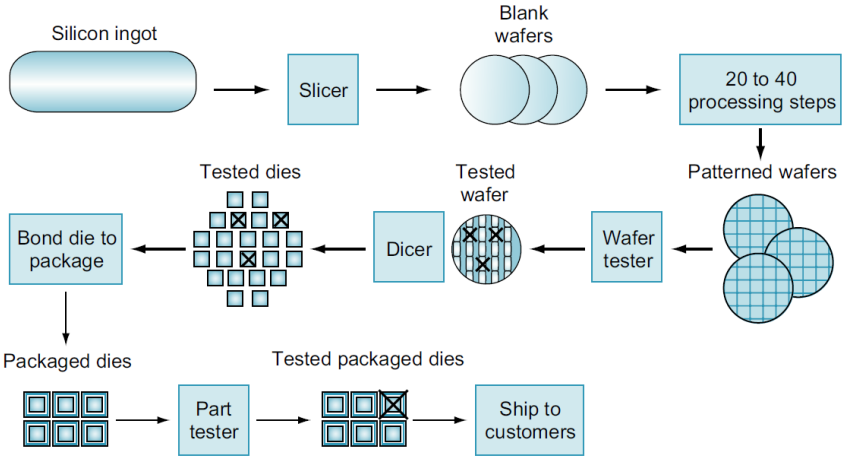
Making a Chip

- **Silicon crystal ingot** A rod composed of a silicon crystal that is between 8 and 12 inches in diameter and about 12 to 24 inches long.



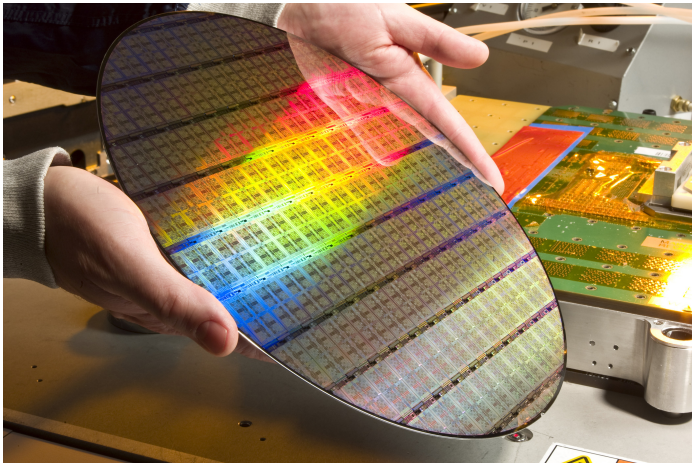
Making a Chip

- **Wafer** A slice from a silicon ingot no more than 0.1 inches thick used to create chips.



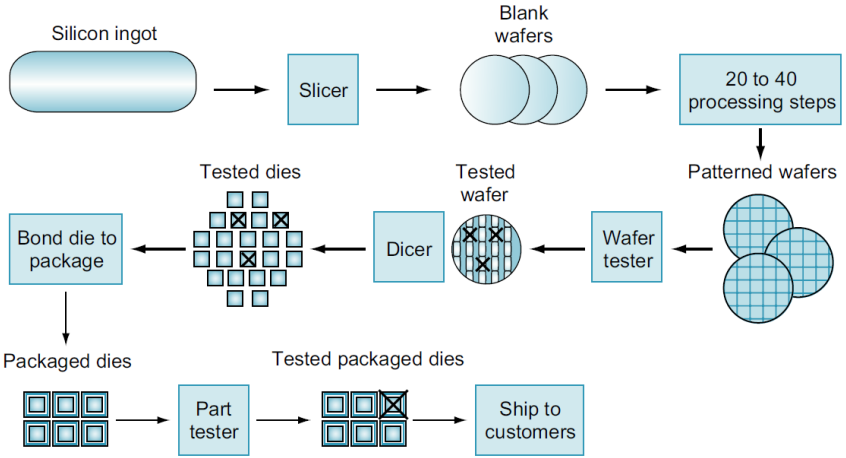
Making a Chip

- **Wafer** A slice from a silicon ingot no more than 0.1 inches thick used to create chips.



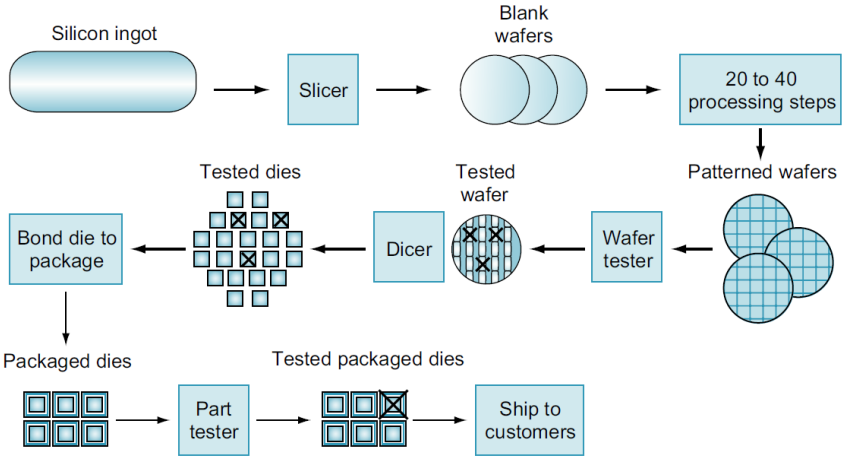
Making a Chip

- Patterns of chemicals are placed on each wafer, creating the transistors, conductors, and insulators.



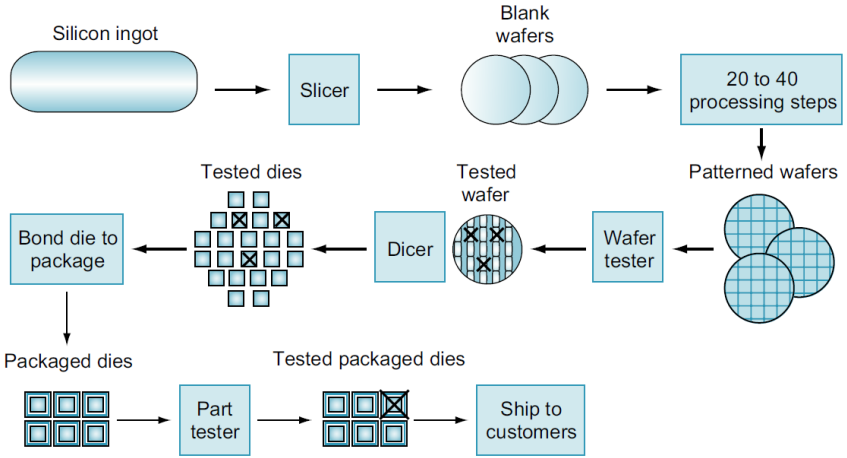
Making a Chip

- **Defect** A microscopic flaw in a wafer or in patterning steps that can result in the failure of the die containing that defect.



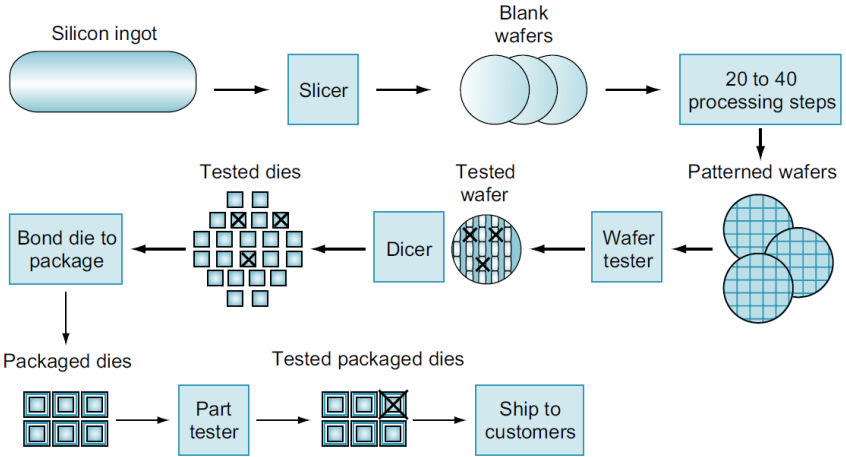
Making a Chip

- **Die** The individual rectangular sections that are cut from a wafer, more informally known as **chips**.



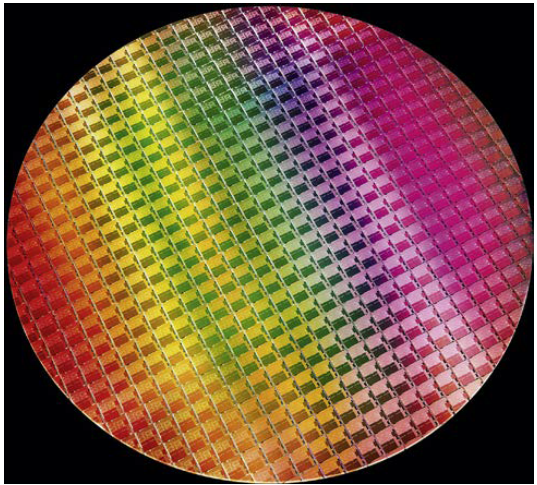
Making a Chip

- **Yield** The percentage of good dies from the total number of dies on the wafer.



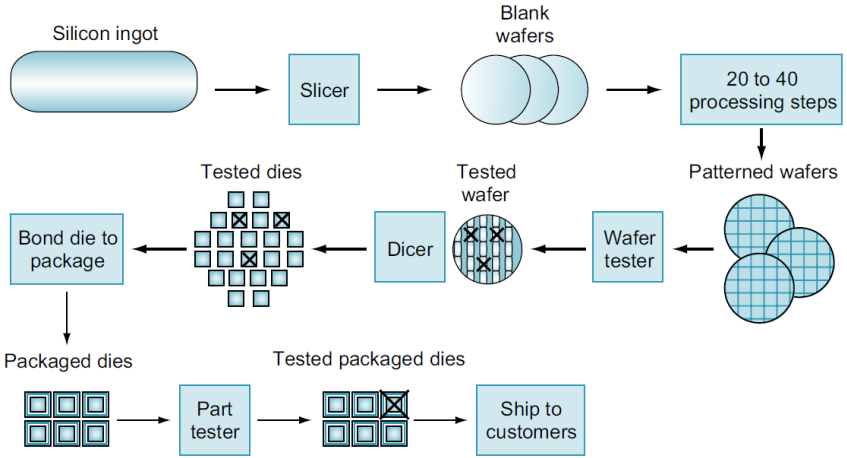
Making a Chip

- This 300mm by 10nm wafer contains 10^{th} Gen $Intel^{®}I7Core^{TM}$ processors, code-named **“Ice Lake”**. The number of dies on this wafer at 100% yield is 506.



Making a Chip

- **Bonding** good dies are connected to the in-put/output pins of a package.



Making a Chip

The cost of an integrated circuit can be expressed in three simple equations:

$$Cost\quad per\quad die = \frac{Cost\quad per\quad wafer}{Dies\quad per\quad wafer\quad \times\quad yield} \tag{1}$$

$$Dies\quad per\quad wafer \approx \frac{Wafer\quad area}{Die\quad area}, \tag{2}$$

since it does not subtract the area near the border of the round wafer that cannot accommodate the rectangular dies.

$$Yeld = \frac{1}{(1 + (Defects\quad per\quad area \times Die\quad area))^N}, \tag{3}$$

based on empirical observations of yields at integrated circuit factories, with the exponent related to the number of critical processing steps.

ensia

Measuring The Performance

What do we mean by performance?

- **Response time** Also called **execution time, wall clock time, or elapsed time**: The total time required for the computer to complete a task, including disk accesses, memory accesses, I/O activities, operating system overhead, CPU execution time, etc.
- **Throughput** Also called **bandwidth**: The number of tasks completed per unit time.
- **Example:** Personal mobile devices are more focused on response time, versus servers, which are more focused on throughput.

What do we mean by performance?

$$Performance = \frac{1}{Execution\quad time}$$

$$Performance_X > Performance_Y$$

$$\frac{1}{Execution\quad time_X} > \frac{1}{Execution\quad time_Y} \tag{4}$$

$$Execution\quad time_Y > Execution\quad time_X$$

Relative Performance

If X is n times as fast as Y, then the execution time on Y is n times as long as it is on X:

$$\frac{Performance_X}{Performance_Y} = \frac{Execution_Y}{Execution_X} \tag{5}$$

If computer A runs a program in 10 seconds and computer B runs the same program in 15 seconds, how much faster is A than B?

$$\begin{aligned} \frac{Performance_A}{Performance_B} = \frac{Execution_B}{Execution_A} = n \\ \frac{15}{10} = 1.5 \end{aligned} \tag{6}$$

A is 1.5 times as fast as B, or B is 1.5 times slower than A,

Measuring Performance

- A processor executes several programs simultaneously. Therefore, the system may optimize throughput rather than the elapsed time for one program.
- **CPU execution time** Also called CPU time. The actual time the CPU spends computing for a specific task and does not include time spent waiting for I/O or running other programs.
- **User CPU time** The CPU time spent in a program itself.
- **System CPU time** The CPU time spent in the operating system performing tasks on behalf of the program.

Measuring Performance

- **Clock cycle** Also called **tick, clock tick, clock period, clock, or cycle**: The time for one clock period, usually of the processor clock, which runs at a constant rate.
- **Clock period**: The length of each clock cycle.
- We compute the CPU execution time for a program as follows:

$$CPU \quad execution \quad time = CPU \quad clock \quad cycles \times Clock \quad cycle \quad time \tag{7}$$

, Or

$$CPU \quad execution \quad time = \frac{CPU \quad clock \quad cycles}{Clock \quad rate} \tag{8}$$

Improving Performance

- We improve performance by reducing the number of clock cycles required for a program or the length of the clock cycle.

Instruction Performance

$$CPU \quad time = IC \times CPI \times Clock \quad Cycle \quad Time \tag{9}$$

- **clock cycles per instruction (CPI)** Average number of clock cycles per instruction for a program or program fragment.
- **Instruction Count (IC)** The number of instructions executed by the program.

$$CPU \quad time = \frac{IC \times CPI}{Clock \quad rate} \tag{10}$$

Program Performance

Hardware/ software component	Affects what?	How?
Algorithm	IC, CPI	**IC**: # program instructions -> # processor instructions **CPI**: Complex instruction -> High CPI
Language	IC, CPI	Abstraction level: High level Vs Low level languages
Compiler	IC, CPI	Translation efficiency of different compilers
Instruction Set Architecture	IC, CPI, Clock rate	See above

ensia

CISC Vs RISC

CISC Vs RISC

There are two main families of microprocessor architecture, namely:

- **CISC architecture (Complex Instruction Set Computer):**
  1. It is an architecture where the instruction format is variable and the instructions are complex.
  2. It allows Multiple addressing modes.
  3. Commercial CISC processors:
     Intel P4 and AMD Athlon

CISC Vs RISC

There are two main families of microprocessor architecture, namely:

- **RISC architecture (Reduced Instruction Set Computer):**
  1. It is an architecture where the instruction format is fixed (A single instruction size: 32 bits) and the instruction set is reduced.
  2. Its architecture supports very few addressing modes (absence of indirect addressing) on the other hand the high number of registers (32 registers)
  3. Commercial RISC processors: Silicon Graphics: MIPS 1986, Hewlett-Packard: HP-PA 1986, Sun: SPARC 1987, Apple-IBM-Motorola: Power PC 1990, DEC: Alpha 1992

CISC Vs RISC

CISC	RISC
Emphasis on hardware	Emphasis on software
Multiple instruction sizes and formats	Instruction of the same size with few formats
Less registers	Uses more registers
More addressing modes	Fewer addressing modes
Extensive use of micro-programming	Complexity in compiler
Instructions take a varying amount of cycle time	Instructions take one cycle time
Pipelining is difficult	Pipelining is easy

Table: Comparison between CISC and RISC

ensia

Thank you!