CSE 2105

COMPUTER ARCHITECTURE

Md. Sabab Zulfiker,
Lecturer,
Department of CSE,
BSMRU, Kishoreganj.

Books

- Computer Organization and Design- The Hardware/Software Interface **By** David A. Patterson & John L. Hennessy
- Computer Organization and Design- The Hardware/Software Interface: RISC-V Edition **By** David A. Patterson & John L. Hennessy
- Structured Computer Organization **By** Andrew S. Tanenbaum

Computer Abstractions and Technology- I

What is computer architecture?

Computer architecture can be defined as a set of rules and methods that describe the functionality, management and implementation of computers. To be precise, it is nothing but rules by which a system performs and operates.

Classes of computing applications

Broadly speaking, computers are used in three different classes of applications.

- **Personal computer (PC)**: A computer designed for use by an individual, usually incorporating a graphics display, a keyboard, and a mouse. Personal computers emphasize delivery of good performance to single users at low cost and usually execute third-party software.

- **Server**: A computer used for running larger programs for multiple users, often simultaneously, and typically accessed only via a network. Servers are oriented to carrying large workloads, which may consist of either single complex applications—usually a scientific or engineering application—or handling many small jobs, such as would occur in building a large web server. These applications are usually based on software from another source.

Classes of computing applications

The low-end servers are typically used for file storage, small business applications, or simple web serving (see Section 6.10). On the other hand , the high end servers are known as **supercomputers**. **Supercomputer** is a class of computers with the highest performance and cost; they are configured as servers and typically cost tens to hundreds of millions of dollars. Supercomputers are usually used for high-end scientific and engineering calculations, such as weather forecasting, oil exploration, protein structure determination, and other large-scale problems.

- **Embedded computer:** A computer inside another device used for running one predetermined application or collection of software. Embedded computers are the largest class of computers and span the widest range of applications and performance. Embedded computers include the microprocessors found in your car, the computers in a television set, and the networks of processors that control a modern airplane or cargo ship.

PMDS, Cloud Computing, & SaaS

- **Personal mobile devices (PMDs)** are small wireless devices to connect to the Internet; they rely on batteries for power, and software is installed by downloading apps. Conventional examples are smart phones and tablets.
- **Cloud Computing** refers to large collections of servers that provide services over the Internet; some providers rent dynamically varying numbers of servers as a utility. Cloud Computing relies upon giant datacenters that are now known as Warehouse Scale Computers (WSCs). Companies like Amazon and Google build these WSCs containing 100,000 servers and then let companies rent portions of them so that they can provide software services to PMDs without having to build WSCs of their own.
- **Software as a Service (SaaS)** delivers software and data as a service over the Internet, usually via a thin program such as a browser that runs on local client devices, instead of binary code that must be installed, and runs wholly on that device. Examples include web search and social networking.

Understanding program performance

The performance of a program depends on a combination of the effectiveness of the algorithms used in the program, the software systems used to create and translate the program into machine instructions, and the effectiveness of the computer in executing those instructions, which may include input/output (I/O) operations. This table summarizes how the hardware and software affect performance.

Hardware or software component	**How this component affects performance**
Algorithm	Determines both the number of source-level statements and the number of I/O operations executed
Programming language, compiler, and architecture	Determines the number of computer instructions for each source-level statement
Processor and memory system	Determines how fast instructions can be executed
I/O system (hardware and operating system)	Determines how fast I/O operations may be executed

Moore’s law

- Moore’s Law states that integrated circuit resources double every 18–24 months. Moore’s Law resulted from a 1965 prediction of such growth in IC capacity made by Gordon Moore, one of the founders of Intel. As computer designs can take years, the resources available per chip can easily double or quadruple between the start and finish of the project. Like a skeet shooter, computer architects must anticipate where the technology will be when the design finishes rather than design for where it starts.

Below your program

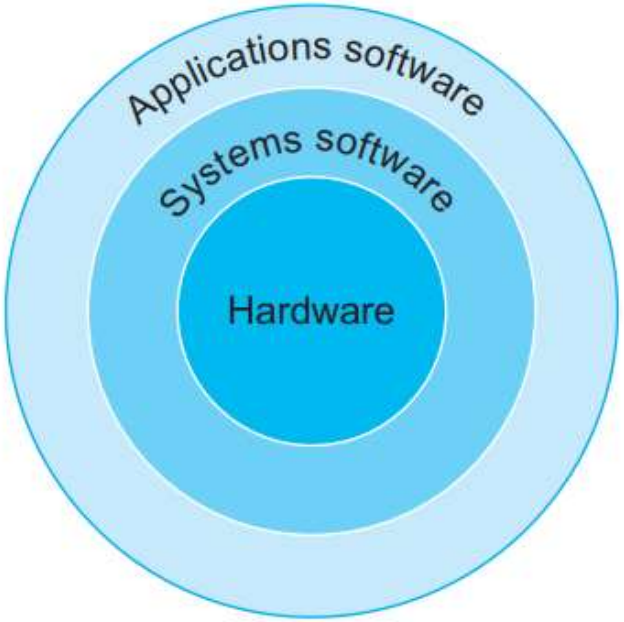


Fig. A simplified view of hardware and software as hierarchical layers, shown as concentric circles with hardware in the center and applications software outermost.

Below your program

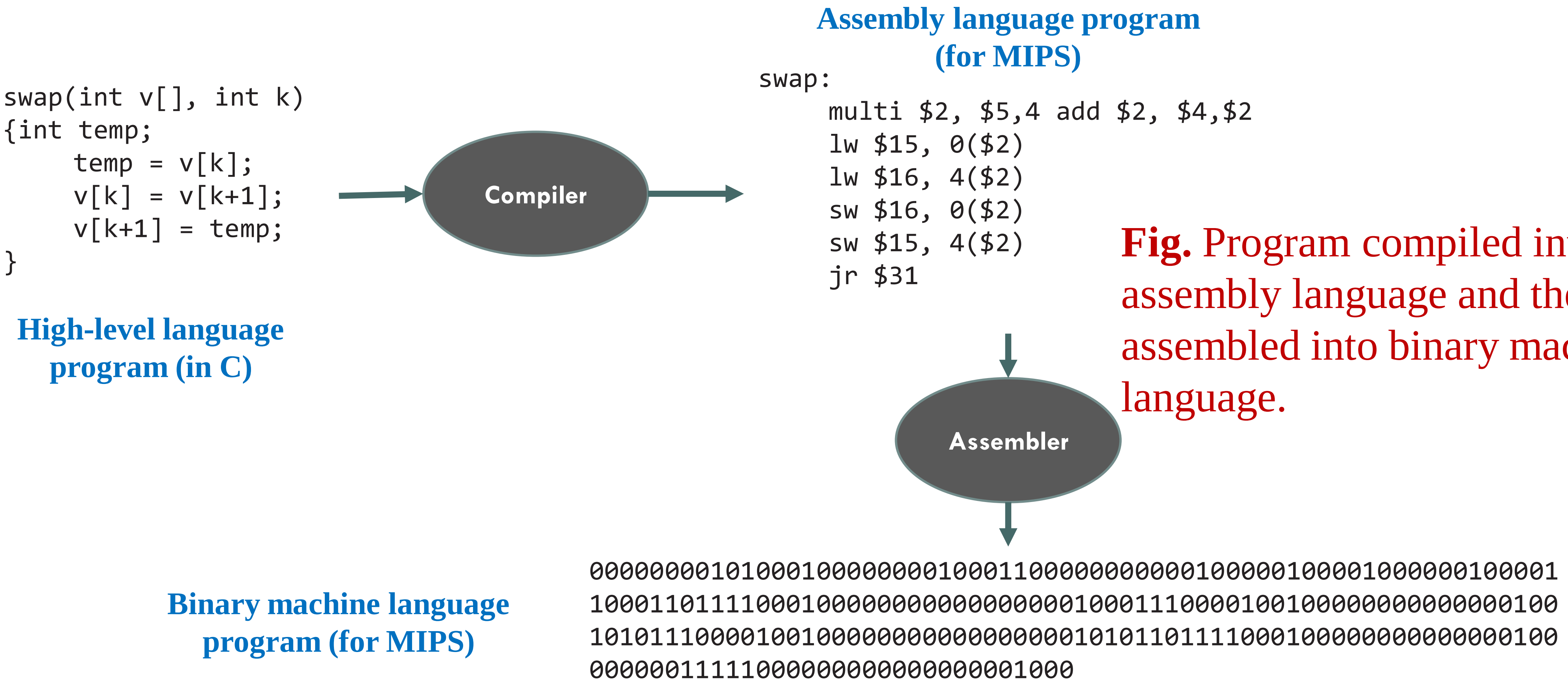
- **Systems software:** Software that provides services that are commonly useful, including operating systems, compilers, loaders, and assemblers. There are many types of systems software, but two types of systems software are central to every computer system today: an **operating system** and a **compiler**.
- An **operating system** interfaces between a user’s program and the hardware and provides a variety of services and supervisory functions. Among the most important functions are: Handling basic input and output operations, allocating storage and memory, etc.

From a high-level language to the language of hardware

- **High-level programming language:** A portable language such as C, C++, Java, or Visual Basic that is composed of words and algebraic notation that can be translated by a compiler into assembly language.
- **Assembly language:** A symbolic representation of machine instructions.
- **Machine language:** A binary representation of machine instructions.
- **Compiler:** A program that translates high-level language statements into assembly language statements.
- **Assembler:** A program that translates a symbolic version of instructions into the binary version.

Memorize the diagram with simpler example

From a high-level language to the language of hardware



```
swap(int v[], int k)
{int temp;
     temp = v[k];
     v[k] = v[k+1];
     v[k+1] = temp;
}
```

High-level language program (in C)

Assembly language program (for MIPS)

```
swap:
     multi $2, $5,4 add $2, $4,$2
     lw $15, 0($2)
     lw $16, 4($2)
     sw $16, 0($2)
     sw $15, 4($2)
     jr $31
```

Binary machine language program (for MIPS)

```
0000000010100010000000010001100000000000100000100001000000100001
1000110111100010000000000000000010001110000100100000000000000100
1010111000010010000000000000000010101101111000100000000000000100
00000011111000000000000000001000
```

Fig. Program compiled into assembly language and then assembled into binary machine language.

Organization of a computer

The five classic components of a computer are input, output, memory, datapath, and control, with the last two sometimes combined and called the processor. The processor gets instructions and data from memory. Input writes data to memory, and output reads data from memory. Control sends the signals that determine the operations of the datapath, memory, input, and output.

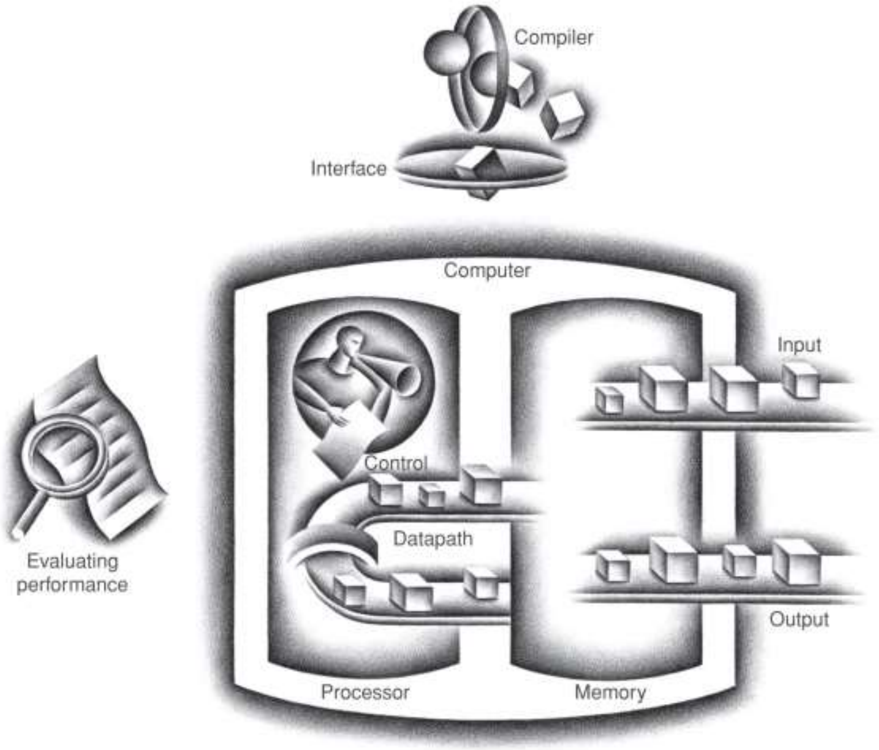


Fig. The organization of a computer, showing the five classic components.

Components of the Apple iPad

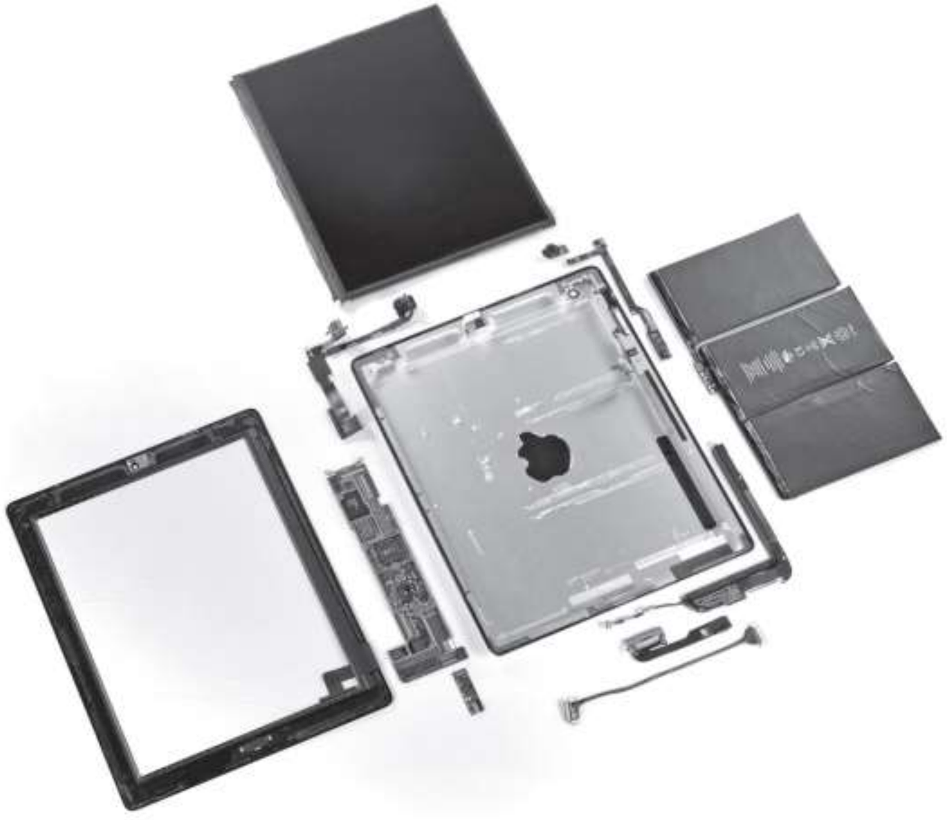


Fig. Components of the Apple iPad 2 A1395.

Apple iPad logic board

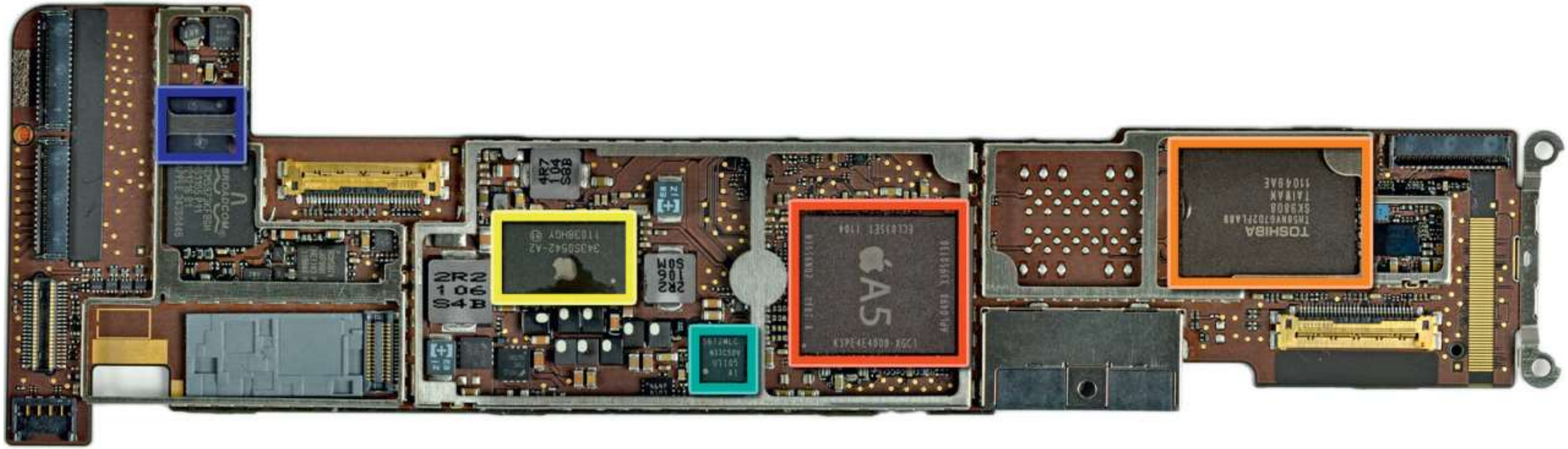


Fig. The logic board of Apple iPad 2

IC, CPU, Datapath, & Control

- **Integrated circuit (IC):** Also called a chip. A device combining dozens to millions of transistors.
- **Central processor unit (CPU):** Also called processor. The active part of the computer, which contains the datapath and control and which adds numbers, tests numbers, signals I/O devices to activate, and so on.
- The processor logically comprises two main components: datapath and control.
- **Datapath:** A datapath is a collection of functional units such as arithmetic logic units (ALUs) or multipliers that perform data processing operations, registers, and buses.

Processor chip inside the A5 package

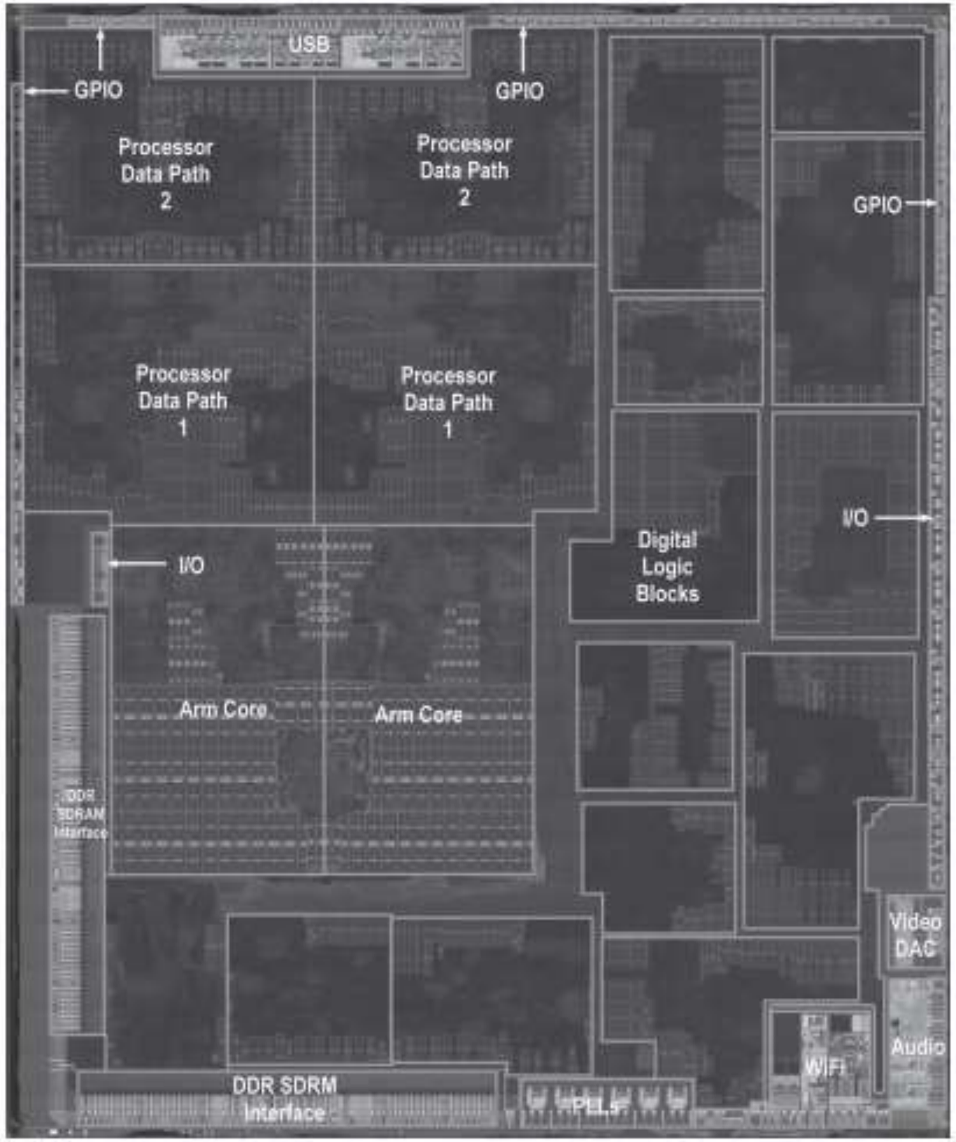


Fig. The processor integrated circuit inside the A5 package.

Memory

- The **memory** is where the programs are kept when they are running; it also contains the data needed by the running programs. The memory is built from **DRAM** chips. DRAM stands for **dynamic random access memory**. Multiple DRAMs are used together to contain the instructions and data of a program.

- Inside the processor is another type of memory—cache memory.

- **Cache memory** consists of a small, fast memory that acts as a buffer for the DRAM memory. (The nontechnical definition of cache is a safe place for hiding things). Cache is built using a different memory technology, **static random access memory (SRAM)**. SRAM is faster but less dense, and hence more expensive, than DRAM.

Memory

- **SRAM (static RAM)** is a type of random access memory (RAM) that retains data bits in its memory as long as power is being supplied. Unlike **dynamic RAM (DRAM)**, which must be continuously refreshed, SRAM does not have this requirement, resulting in better performance and lower power usage.

- **Volatile memory:** Storage, such as DRAM, that retains data only if it is receiving power.

- **Nonvolatile memory:** A form of memory that retains data even in the absence of a power source and that is used to store programs between runs. A DVD disk is nonvolatile.

- To distinguish between the volatile memory used to hold data and programs while they are running and this nonvolatile memory used to store data and programs between runs, the term **main memory or primary memory** is used for the former, and **secondary memory** for the latter.

Memory

- Example of Secondary Memory: Magnetic Disks.
- Because of their size and form factor, personal Mobile Devices use flash memory, a nonvolatile semiconductor memory, instead of disks.

SAM vs RAM

- In computing, sequential access memory (SAM) is a class of data storage devices that read stored data in a sequence. This is in contrast to random access memory (RAM) where data can be accessed in any order. Sequential access devices are usually a form of magnetic storage or optical storage. They are typically used for secondary storage in general-purpose computers.

Instruction set architecture

- Also called **architecture**. An abstract interface between the hardware and the lowest-level software that encompasses all the information necessary to write a machine language program that will run correctly, including instructions, registers, memory access, I/O, and so on.

Transistor

- A **transistor** is simply an on/off switch controlled by electricity. The *integrated circuit* (IC) combined dozens to hundreds of transistors into a single chip. When Gordon Moore predicted the continuous doubling of resources, he was predicting the growth rate of the number of transistors per chip.

Chip manufacturing process

- The manufacture of a chip 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:

- Excellent conductors of electricity
- Excellent insulators from electricity
- Areas that can conduct or insulate under special conditions (as a switch)

- The process starts with a silicon crystal ingot, which looks like a giant sausage. Today, ingots are 8–12 inches in diameter and about 12–24 inches long. An ingot is finely sliced into wafers no more than 0.1 inches thick.

Chip manufacturing process

- After being sliced from the silicon ingot, blank wafers are put through 20 to 40 steps to create patterned wafers, forming the transistors, conductors, and insulators discussed earlier.
- These patterned wafers are then tested with a wafer tester, and a map of the good parts is made.
- Then, the wafers are diced into dies.
- In this figure, one wafer produced 20 dies, of which 17 passed testing. (X means the die is bad.) The yield of good dies in this case was 17/20, or 85%.
- These good dies are then bonded into packages and tested one more time before shipping the packaged parts to customers. One bad packaged part was found in this final test.

Memorize the diagram

Chip manufacturing process

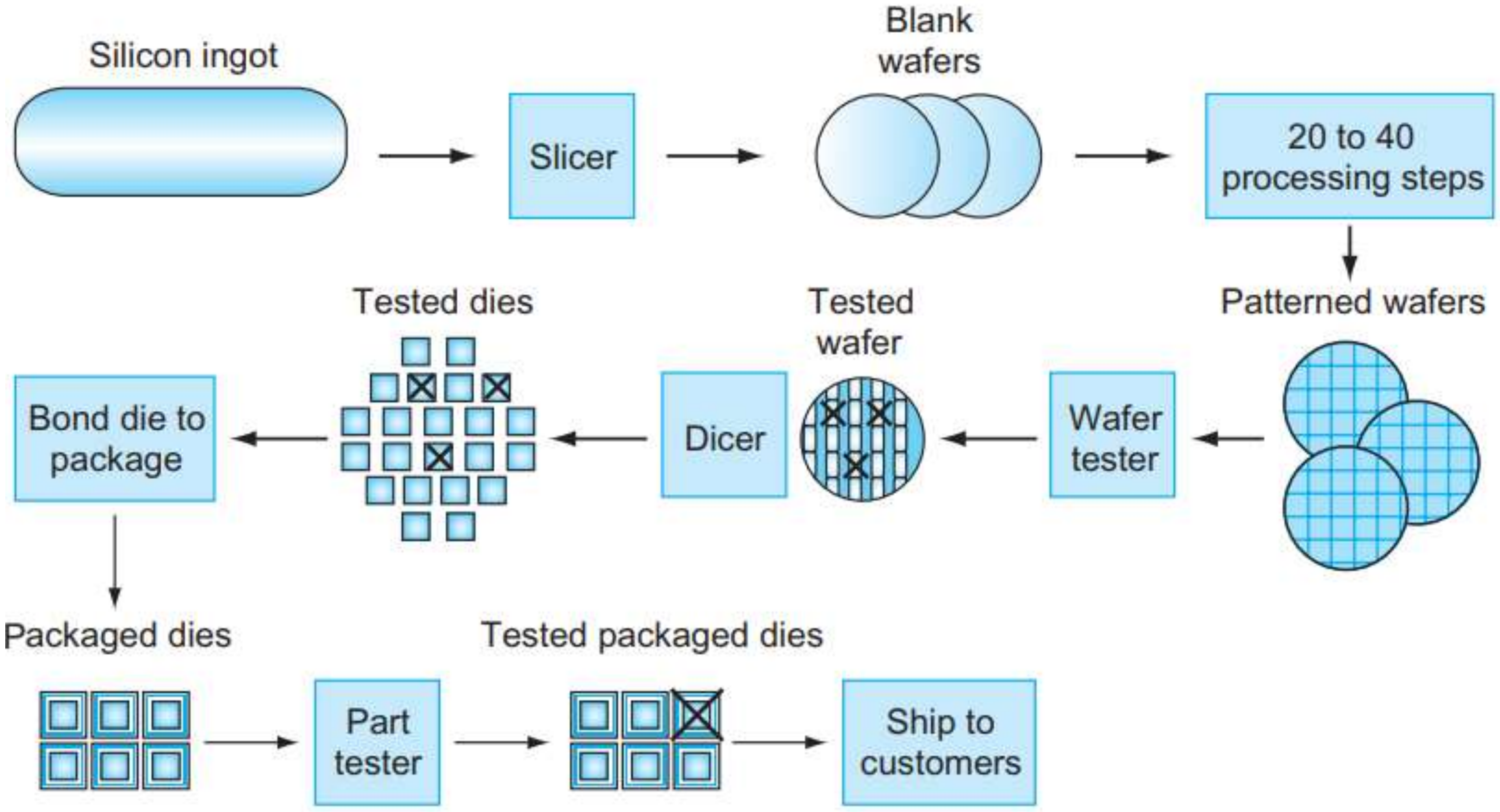


Fig. The chip manufacturing process

Chip manufacturing process

- **Defect:** A single microscopic flaw in the wafer itself or in one of the dozens of patterning steps can result in that area of the wafer failing. These **defects**, as they are called, make it virtually impossible to manufacture a perfect wafer.
- **Yield:** The percentage of good dies from the total number of dies on the wafer.

Fig. A 12-inch (300 mm) wafer of Intel Core i7

Memorize all the formulas

Cost of an Integrated Circuit

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

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

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

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

Cost of an Integrated Circuit

Assume a 15 cm diameter wafer has a cost of 12, contains 84 dies, and has 0.020 defects/cm2.

I. Find the yield for the wafer.

II. Find the cost per die for the wafer.

III. If the number of dies per wafer is increased by 10% and the defects per area unit increases by 15%, find the die area and yield.

IV. Assume a fabrication process improves the yield from 0.92 to 0.95. Find the defects per area unit for each version of the technology given a die area of 200 mm2.

Thank You