Moore’s Law and Next Steps for Silicon

SOCIAL CAPITAL_

Table of Contents

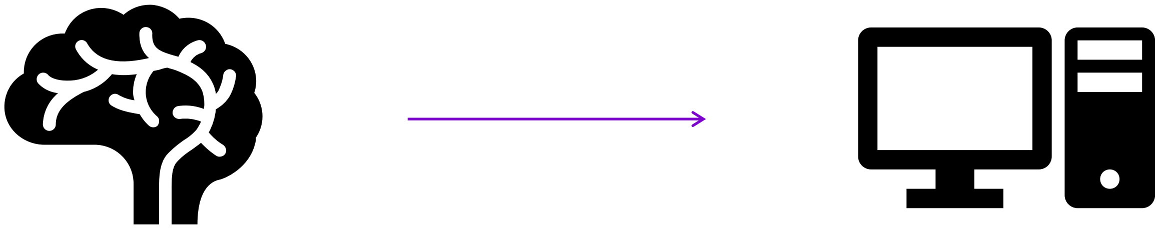
How to Read This Presentation

- This presentation aims to provide a common foundation on semiconductors, covering everything from the basic physics of the p-n junction to new types of chips built for specific tasks like AI training and inference.
- Each section of this presentation builds on the prior and assumes no prior knowledge about the discussed topic. You should read this presentation in chronological order like a book.
- At the end of each section, there will be a slide with links to further short readings and YouTube videos to reinforce and enhance your learning.
- By the end of this presentation, you should have a good understanding of how semiconductor devices work, how they are designed and built, and what new types of silicon are being developed to overcome the slowing of Moore's Law.

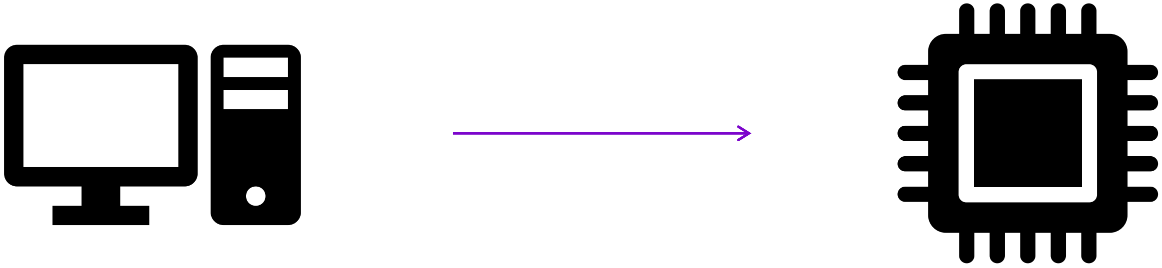
CHAPTER 01

Introduction to Semiconductors

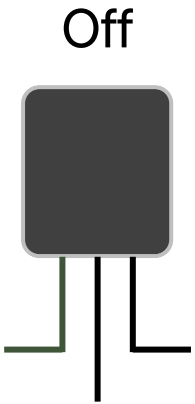
The original goal of building a computer was to create a digital brain that could process information and instructions faster and more reliably than the human brain



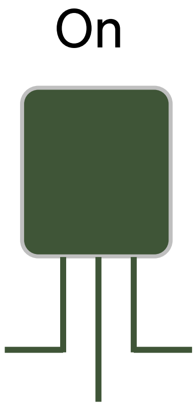
At the core of every digital brain is a microprocessor, which is created from a type of material called a ‘semiconductor’, that can process information and instructions



Microprocessors themselves are built using 'transistors', which are tiny electric switches that switch between on and off states by applying an electric voltage

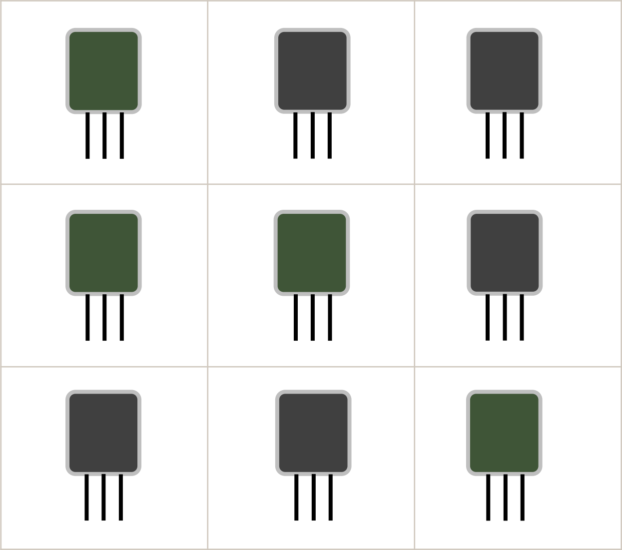


Transistor is off because no voltage is applied



Transistor is on after a voltage is applied

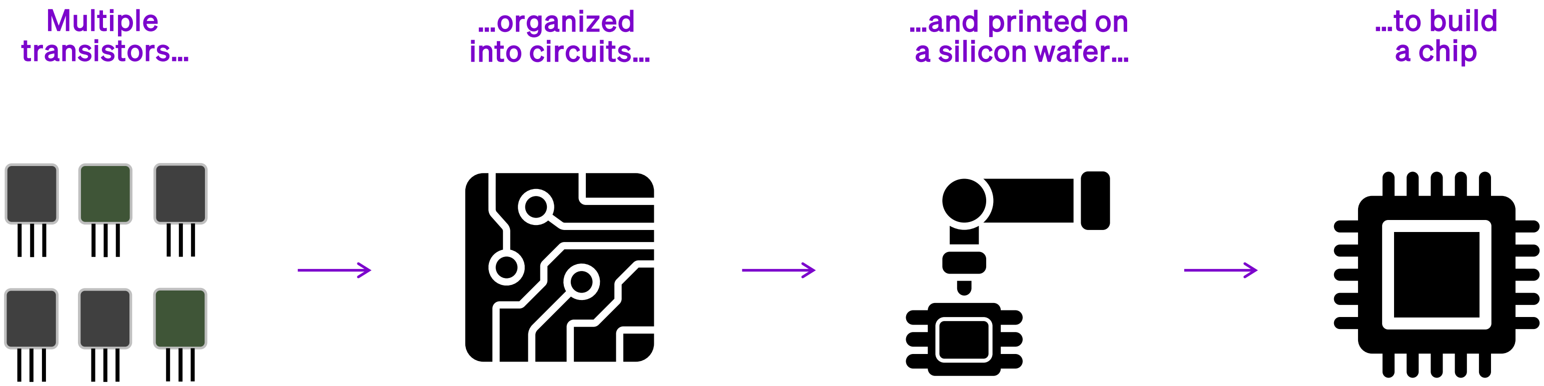
The off and on states of transistors represent the 0s and 1s used in binary code, the basic language that computers use to represent data and execute instructions



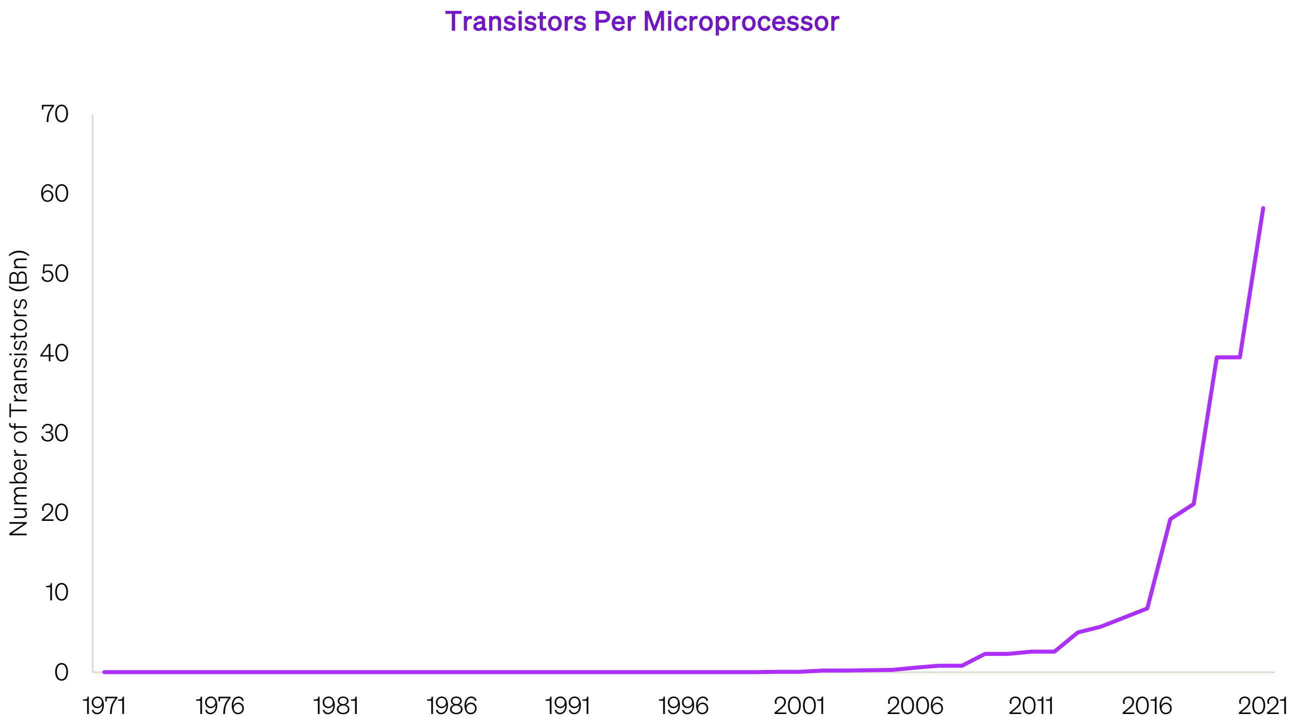
=

1	0	0
1	1	0
0	0	1

Multiple transistors can be organized into different circuits that carry out logical and mathematical functions, which are printed on a sheet of silicon to build a chip

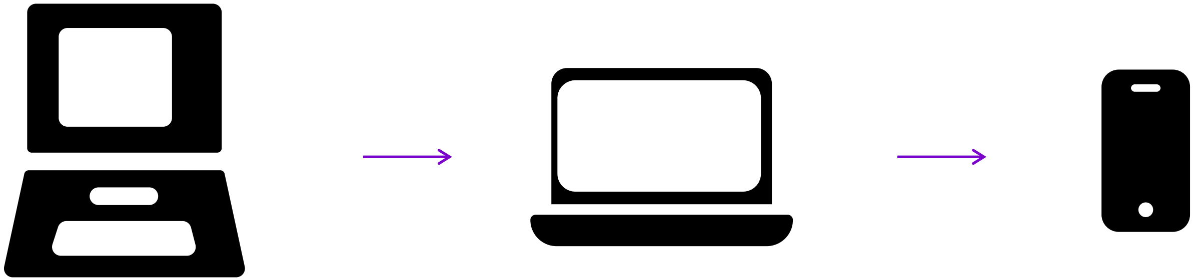


Since the 1960s, the number of transistors that can fit on a single chip has roughly doubled every two years, predicted by a famous relationship called 'Moore's Law'



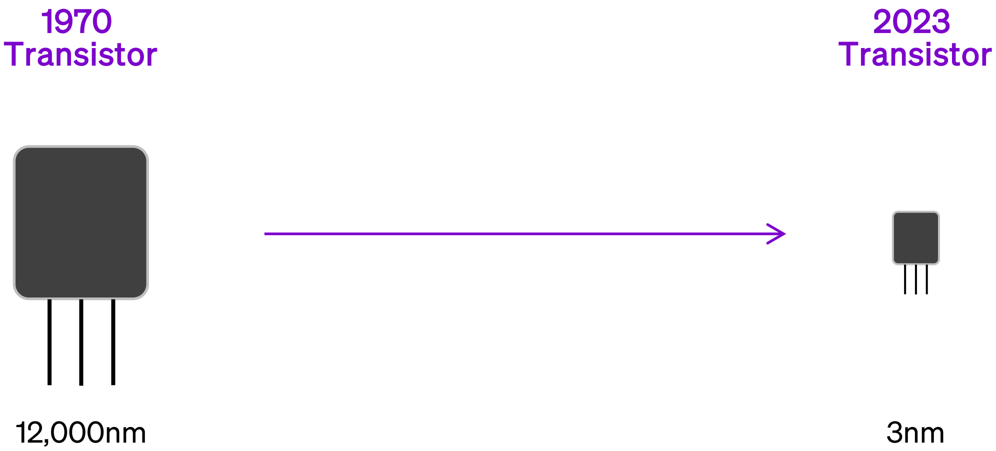
Source: Our World in Data

This has led to an exponential increase in compute power, and allowed us to build smarter and more compact devices



How did we achieve this?

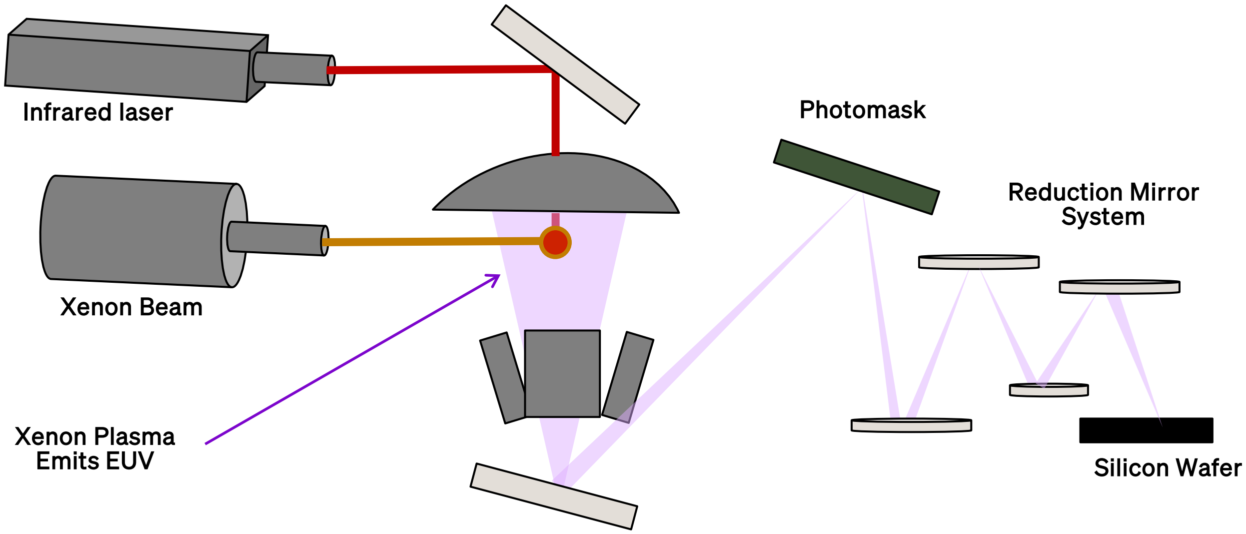
To fit more transistors onto a single chip, we shrunk the size of a transistor to thousands of times smaller than the width of a human hair



Source: New York Times

We achieved this by developing new methods of printing chips, which use ‘extreme ultraviolet light’ to etch smaller and smaller transistors onto a sheet of silicon

How Extreme Ultraviolet Light (EUV) is Generated



Smaller, more densely packed transistors have enabled the development of higher-performance chips that are more area and power efficient

Performance

More transistors allow a chip to process more complicated instructions, and more tightly-packed transistors lead to shorter travel time for electrons

Size

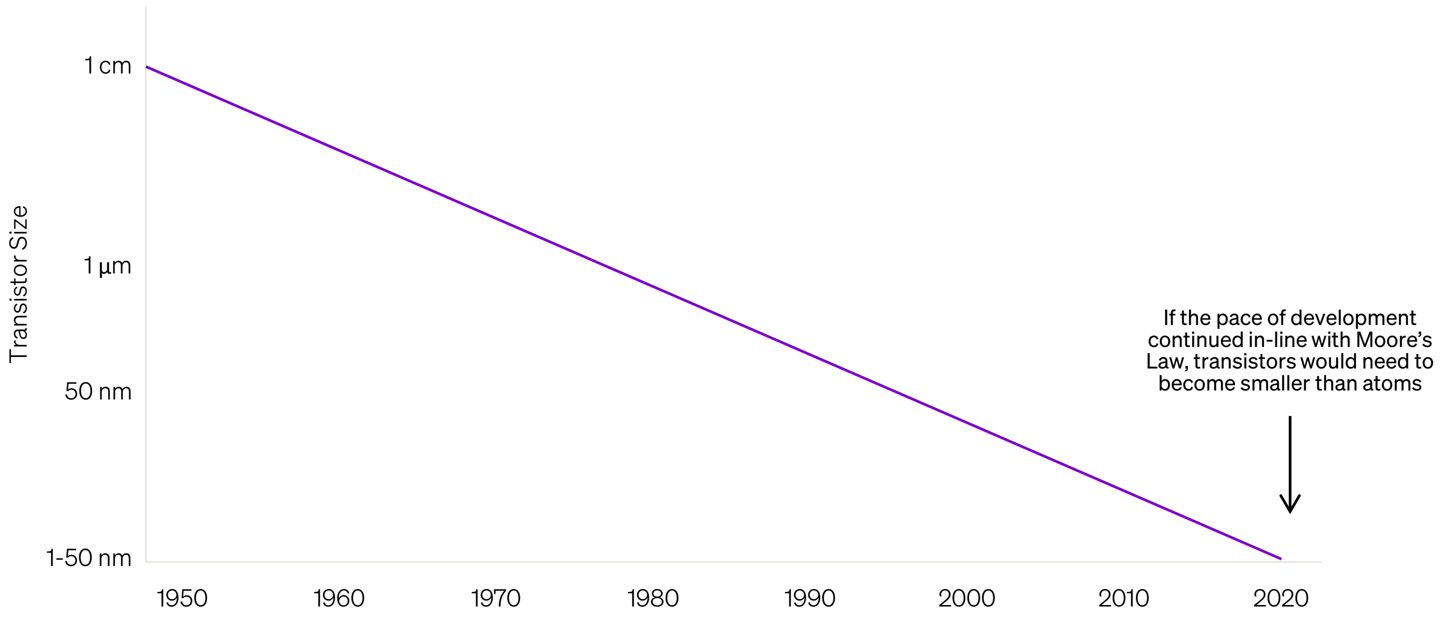
Smaller, more densely packed transistors allow the same integrated circuit to be printed on a smaller chip, which is useful for smartphones and other devices

Power-Efficiency

Smaller, more densely packed transistors result in less travel time for electrons between transistors as well as lower voltage requirements to switch on and off

But as transistors get smaller and smaller, they approach the size of an atom, meaning we have struggled to continue to increase transistor count in-line with Moore's Law

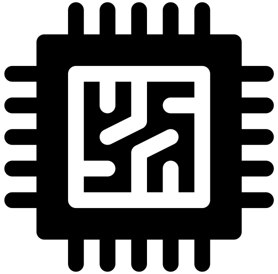
Transistor Size Trend Line According to Moore's Law



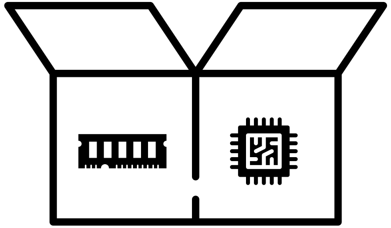
Source: Pokropivny et al. (2007), Introduction to Nanomaterials and Nanotechnology

So, to increase performance, companies have turned to developing new types of custom semiconductors and packaging them more closely with other components

New types of custom chips can be optimized for specific tasks



Chips can be packaged closely with components like memory to improve performance

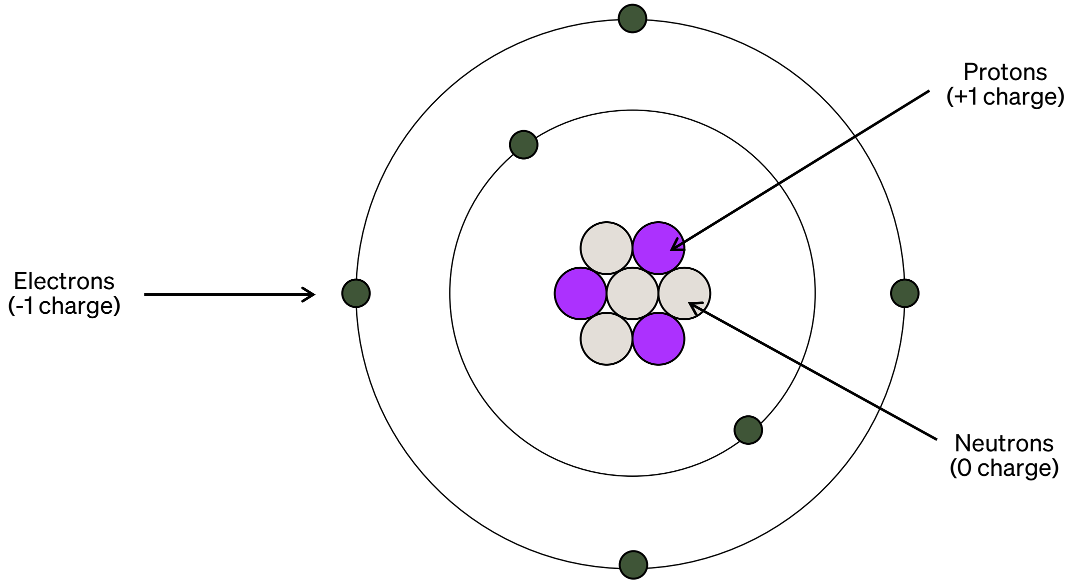


To understand how this all works, we need to go back to the basic physics behind electricity and transistors

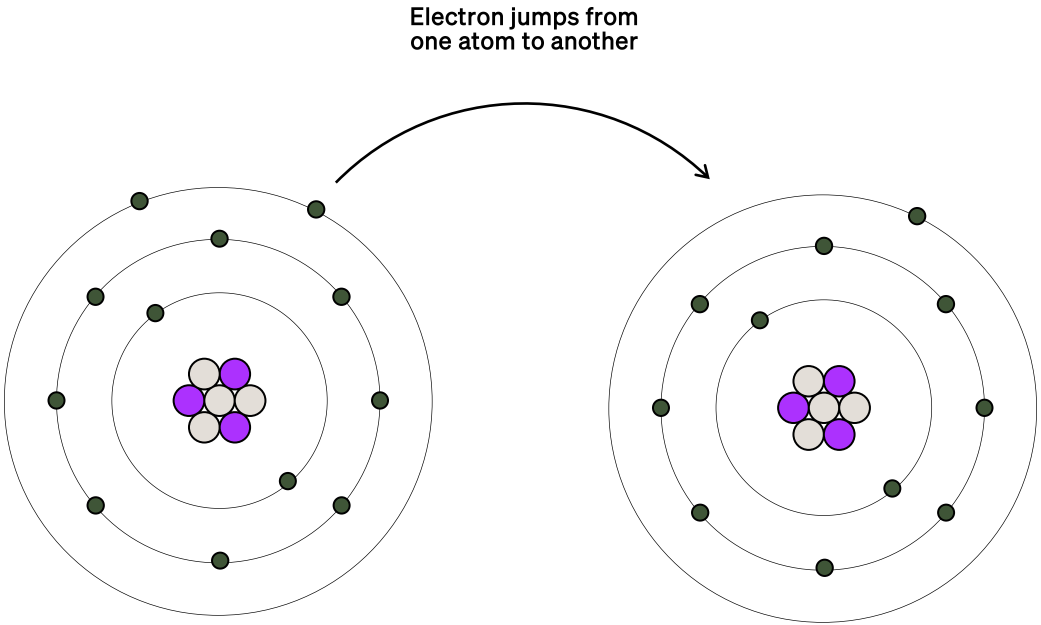
CHAPTER 02

Semiconductor Physics 101

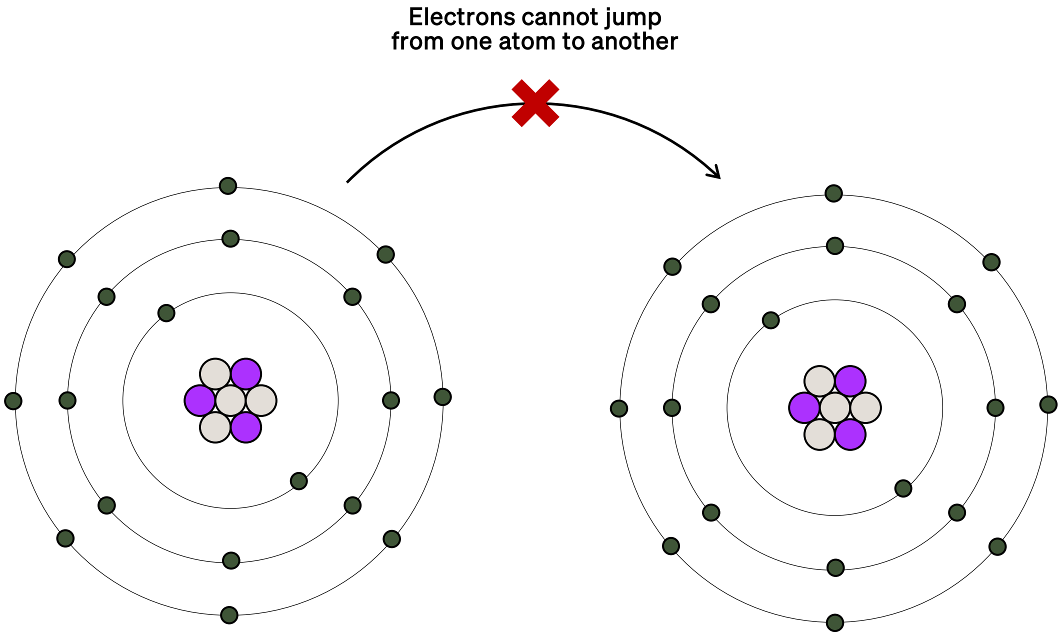
An atom is made up of tiny particles called protons, neutrons and electrons which are organized in 'shells' and each carry a different electrical charge



For certain materials like metals, which are conductors, the electrons in the outermost shell are free to jump between one atom and another



Other materials, called insulators, have very tightly-bound electrons which are not free to jump between neighboring atoms



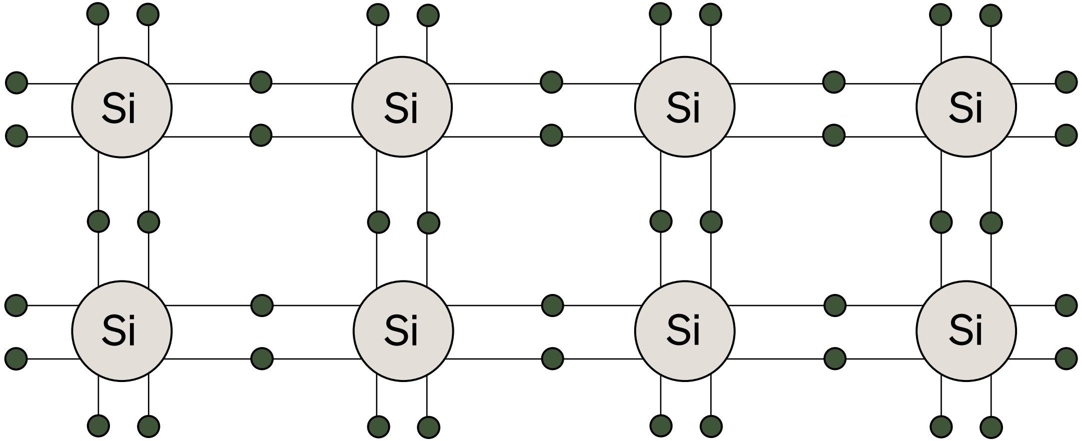
Semiconductors are materials that sit between conductors and insulators and only conduct electricity in specific circumstances

By controlling when they do or do not conduct electricity, we can make tiny electronic switches called transistors

How do transistors work?

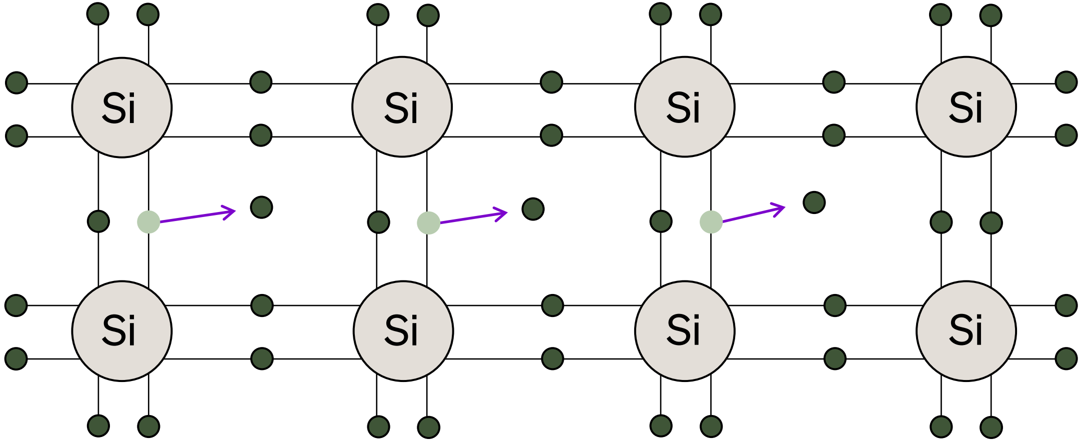
Transistors are built using silicon, an extremely common material with semiconductive properties

An atom of silicon has four electrons in its outer shell, which bond with other silicon atoms to produce a lattice structure



Source: Veritasium

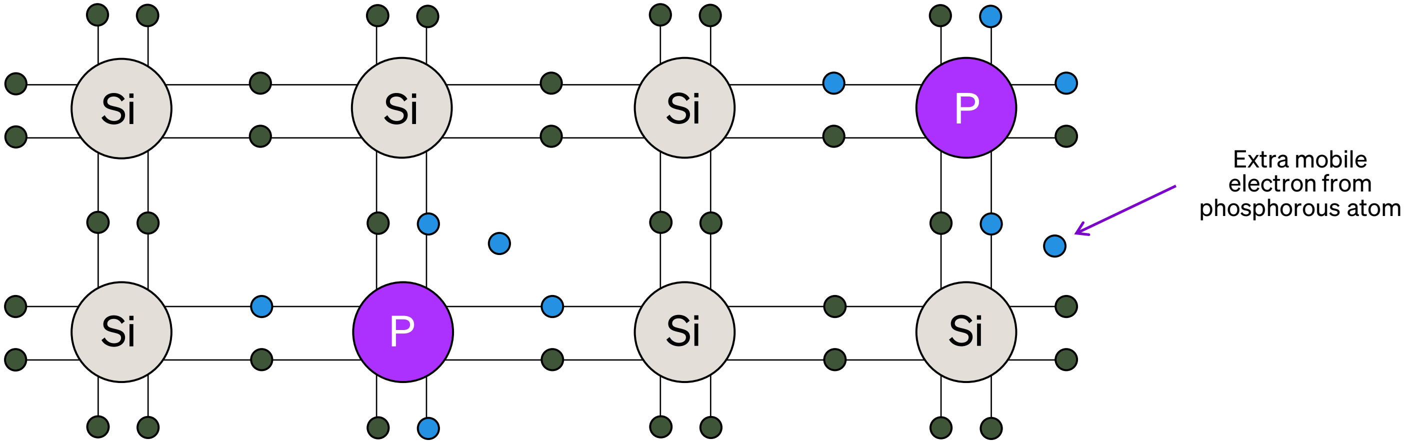
While these electrons are typically trapped in bonds, sometimes, a few electrons can come loose and flow through the lattice, which makes silicon a semiconductor



Source: Veritasium

A lattice of silicon can be further ‘doped’ with other materials to introduce more mobile electrons that can conduct electricity

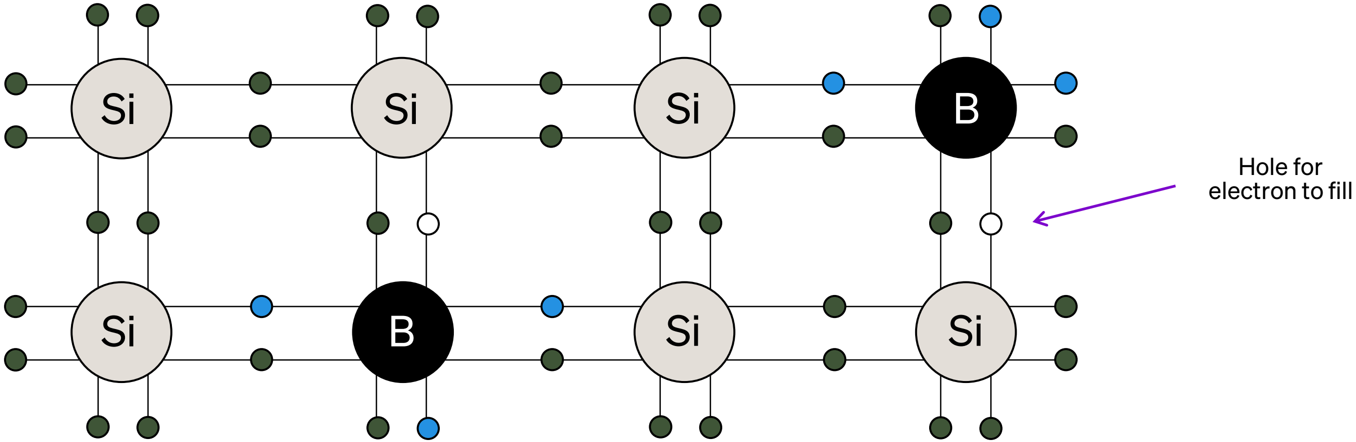
For example, an atom of phosphorous, which contains five electrons in its outer shell, can be added to the silicon lattice to introduce more mobile electrons



Source: Veritasium

This is called an ‘n-type’ semiconductor

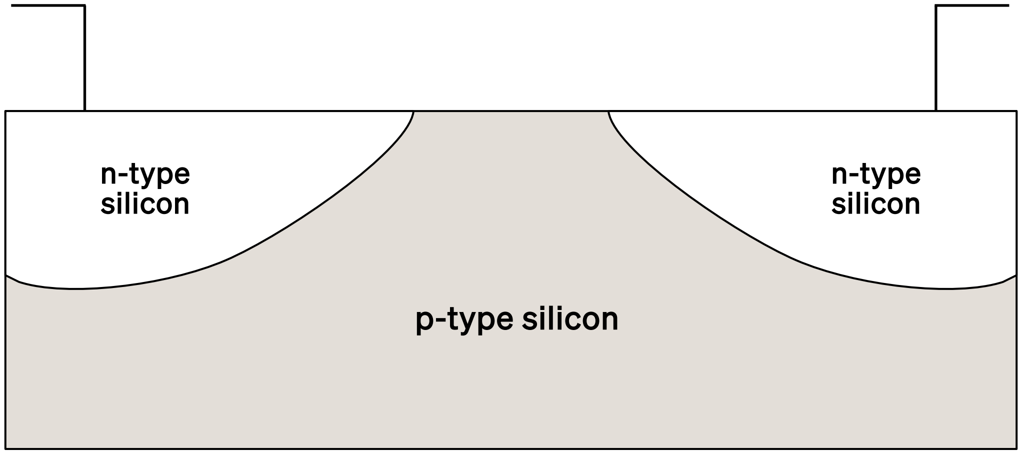
Boron, an atom with only three electrons in its outer shell, can be added to the silicon to introduce a 'hole' that an electron can fill



Source: Veritasium

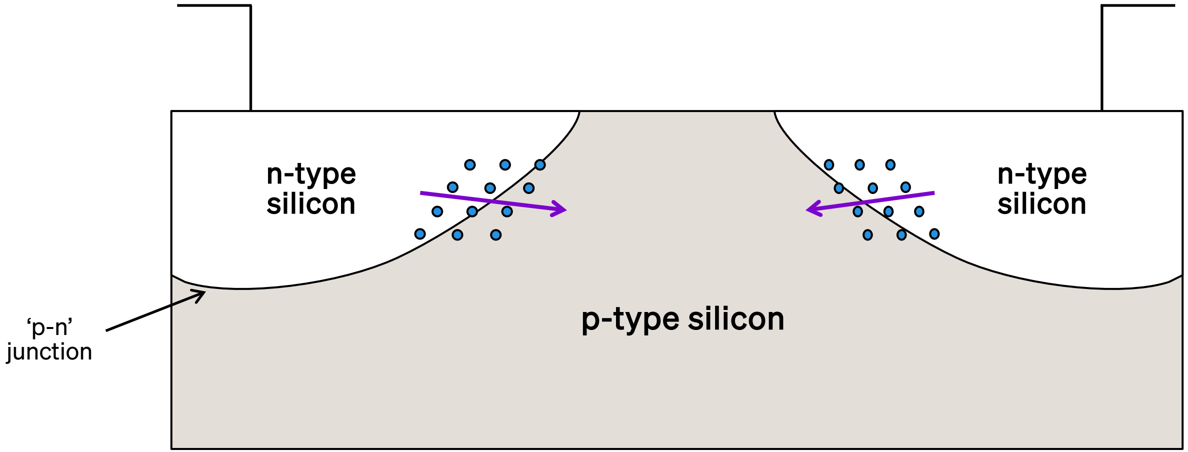
This is called a ‘p-type’ semiconductor

‘n-type’ and ‘p-type’ semiconductors can be joined together to create a tiny electric switch called a transistor



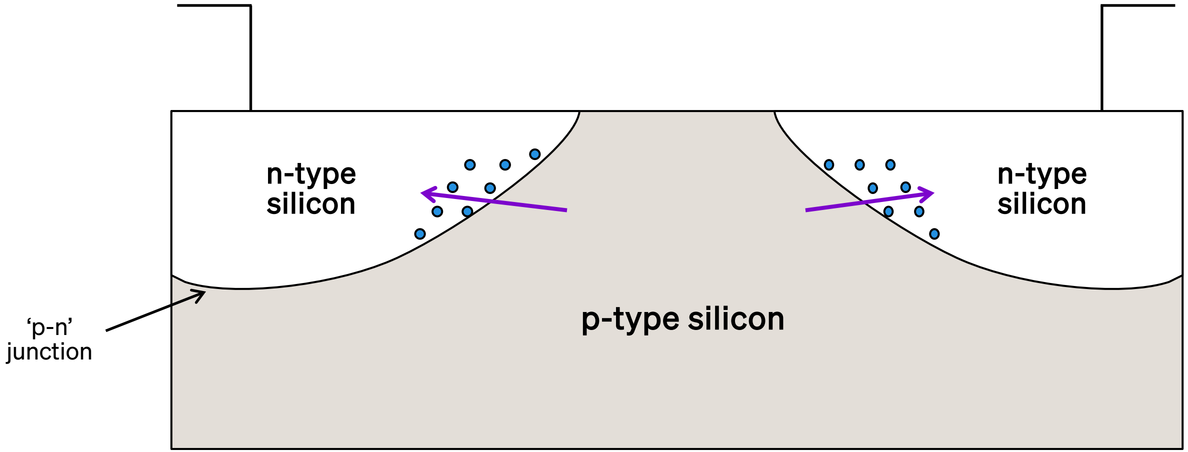
Source: Veritasium

When they are joined together, extra electrons from the ‘n-type’ silicon move across the ‘p-n junction’ to fill the holes in the ‘p-type’ silicon

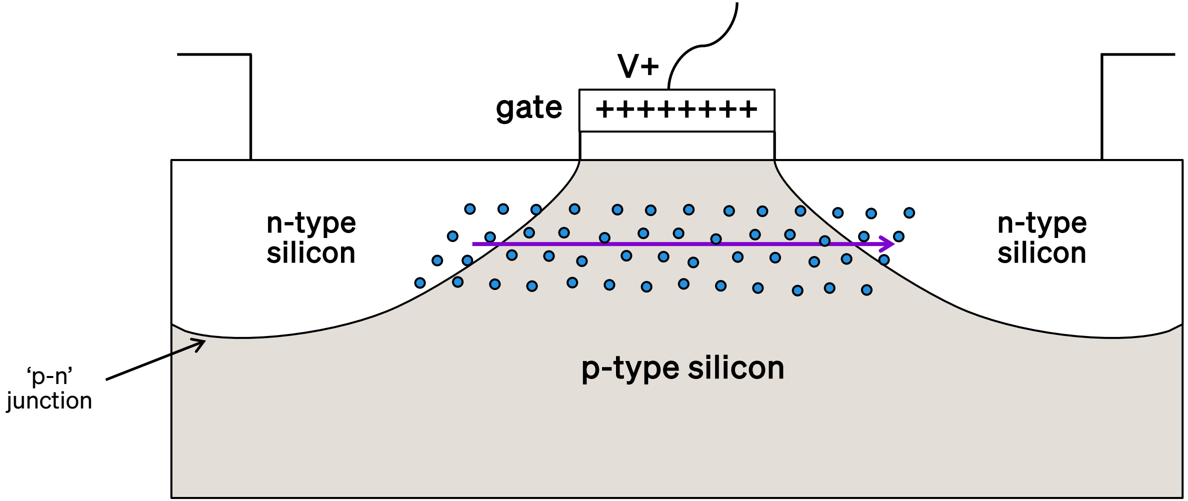


Source: Veritasium

This results in the ‘p-type’ silicon becoming negatively charged, which repels further electrons from crossing the ‘p-n’ junction

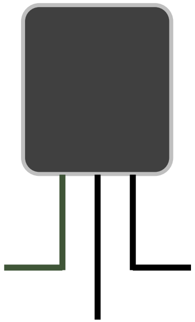


To overcome this repulsive negative force at the 'p-n' junction, a positive voltage can be applied at the 'gate' to switch the transistor on and allow electrons to flow through



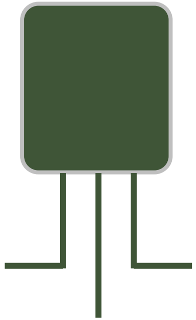
Switching the transistor on and off allows us to represent the binary digits 0 and 1, or ‘true’ and ‘false’

0



Transistor is off because no voltage is applied at gate

1

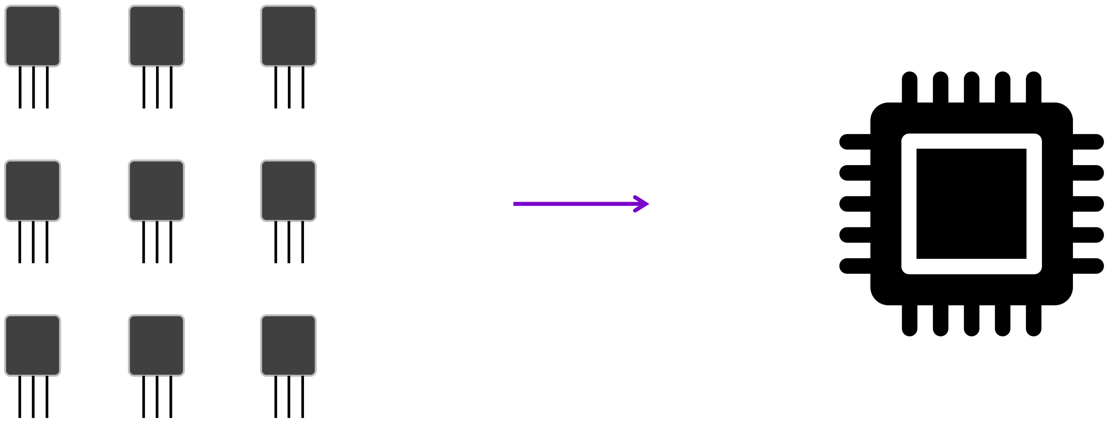


Transistor is on after a voltage is applied at gate

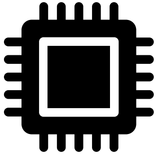
Multiple transistors and other components can be organized to build a 'microprocessor', a type of chip that can take input data and process it according to defined instructions

Multiple Components…

Organized into a microprocessor

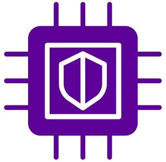


There are many different types of microprocessors, and each are best suited for different types of applications



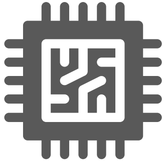
Central Processing Unit

Primary component of a computer that executes a wide set of instructions



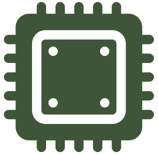
Field Programmable Gate Array

Reconfigurable processor that can be programmed by the user



Application Specific Integrated Circuit

Custom-designed chip to perform a narrower range of tasks very efficiently



Graphics Processing Unit

Specialized processor for computing graphics and other parallel tasks like AI

Source: Richard (2023), Understanding Semiconductors

To understand how different types of microprocessors work and how they are designed, we first need to understand the central processing unit (CPU)

Central Processing Unit

Primary component of a computer that executes a wide set of instructions

Field Programmable Gate Array

Reconfigurable processor that can be programmed by the user

Application Specific Integrated Circuit

Custom-designed chip to perform a narrower range of tasks very efficiently

Graphics Processing Unit

Specialized processor for computing graphics and other parallel tasks

Source: Richard (2023), Understanding Semiconductors

Dive Deeper…

Further Reading & Watching

Watching:

- Transistors Explained - How Transistors Work (The Engineering Mindset)
- How Does a Transistor Work? (Veritasium)

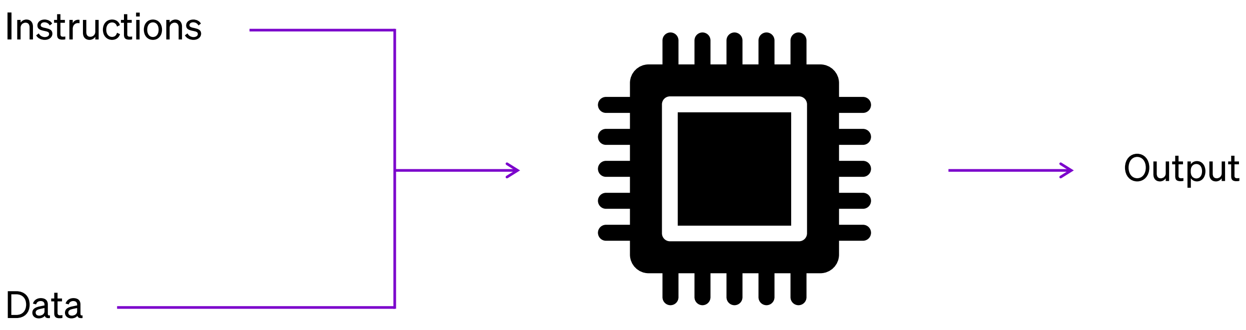
Reading:

- Transistor (TechTarget)
- Using FinFETs vs. MOSFETs for IC Design (Cadence)

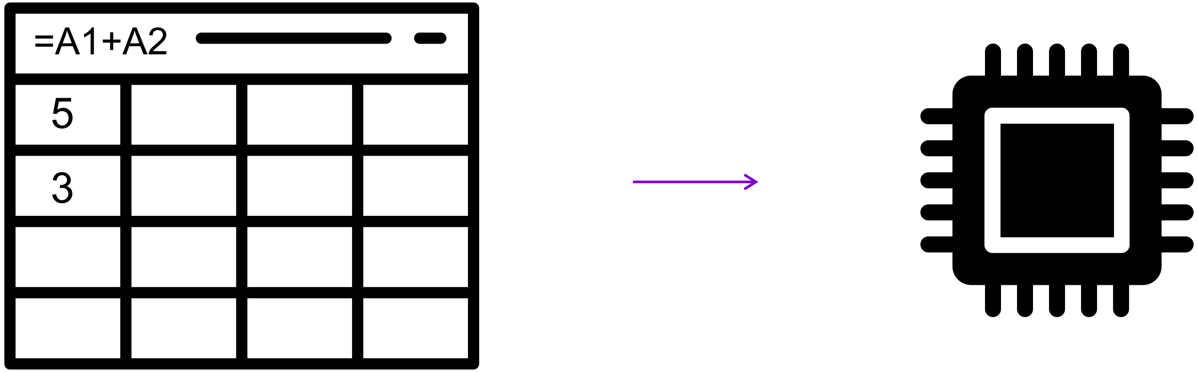
CHAPTER 03

How the CPU Works

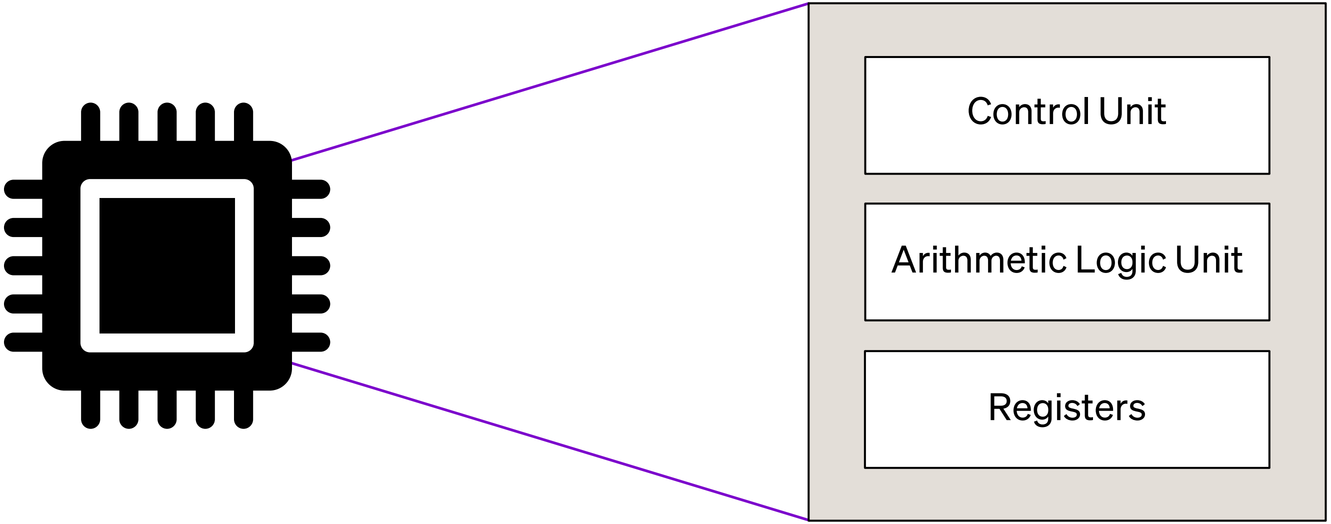
The central processing unit is like the central brain of a computer that takes instructions and data as inputs to generate an output



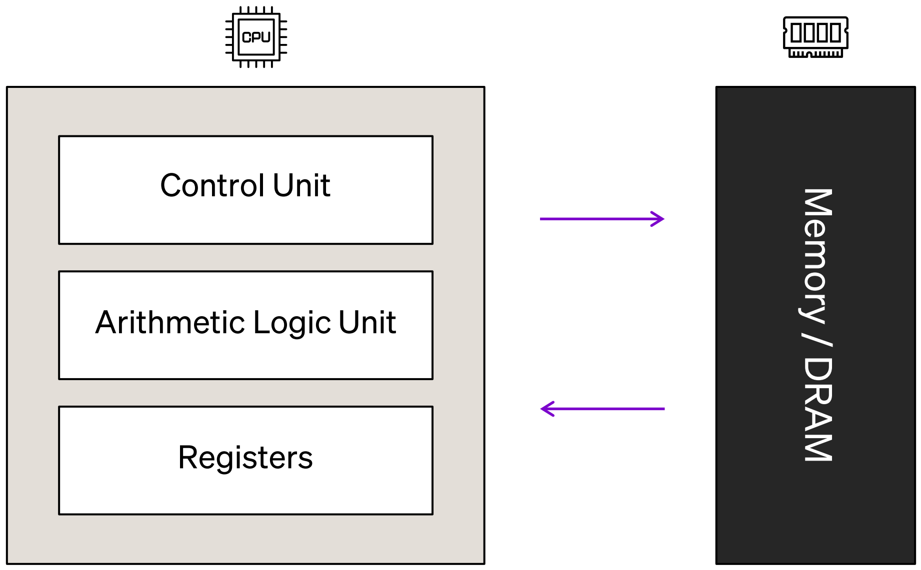
When you perform an action on a computer, like adding together two numbers in Excel, these actions are read and executed by the central processing unit



It does this using three main types of circuits, which are designed using complex arrangements of transistors



These circuits work together with outside memory units, called ‘DRAM’, to complete instructions in a process called the ‘fetch, decode, execute’ cycle



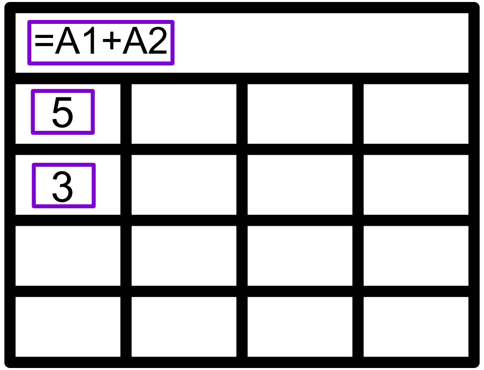
How does this work?

Applications like Microsoft Excel are written using lines of instructions called ‘code’, which contain rules that dictate how the application should function, handle data, and interact

C++

```
#include <iostream>
#include <string>

class Book {
private:#include <iostream>
#include <string>
#include <map>

double calculateFormula(const
std::string& formula, const
std::map<std::string, double>&
cells) {
   size_t plusPos = formula.find('+');
   if (plusPos != std::string::npos) {
      std::string leftCell =
```

When we write a formula into excel, the programming language analyzes the formula and interprets this to determine the operation that must be performed on the chosen cells



C++

```
#include <iostream>
#include <string>

class Book {
private:#include <iostream>
#include <string>
#include <map>

double calculateFormula(const
std::string& formula, const
std::map<std::string, double>&
cells) {
  size_t plusPos = formula.find('+');
  if (plusPos != std::string::npos) {
    std::string leftCell =
```

Then, a software tool called a 'compiler' reads the high-level code by breaking it down into tokens and converting it into a 'parse tree' to understand the structure of the program

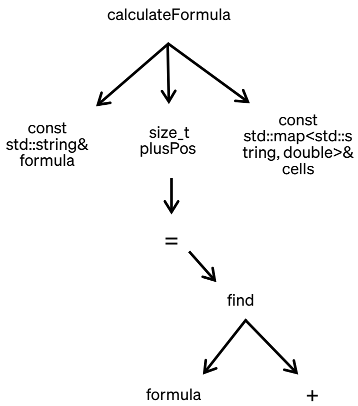
C++

```
#include <iostream>
#include <string>

class Book {
private:#include <iostream>
#include <string>
#include <map>

double calculateFormula(const
std::string& formula, const
std::map<std::string, double>&
cells) {
   size_t plusPos = formula.find('+');
   if (plusPos != std::string::npos) {
      std::string leftCell =
```

Tokens

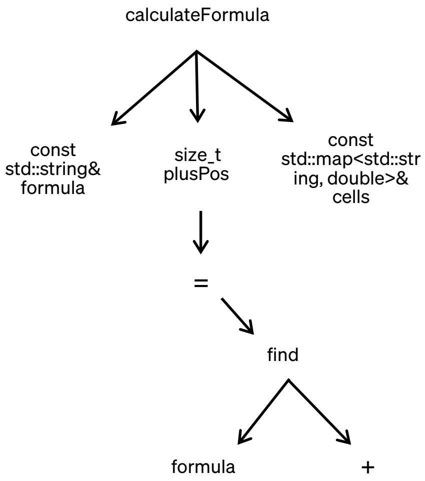
```
double
calculateFormula
(
const
std::string
&
Formula
,
```

Tree



Finally, the compiler optimizes the code to improve its performance and translates the hierarchy of the tree into binary instructions for the CPU to perform called 'machine code'

Tree



Machine Code

00010000011110001100110000100
00111100011001100001000011110
00110011000010000111100011001
10000100001

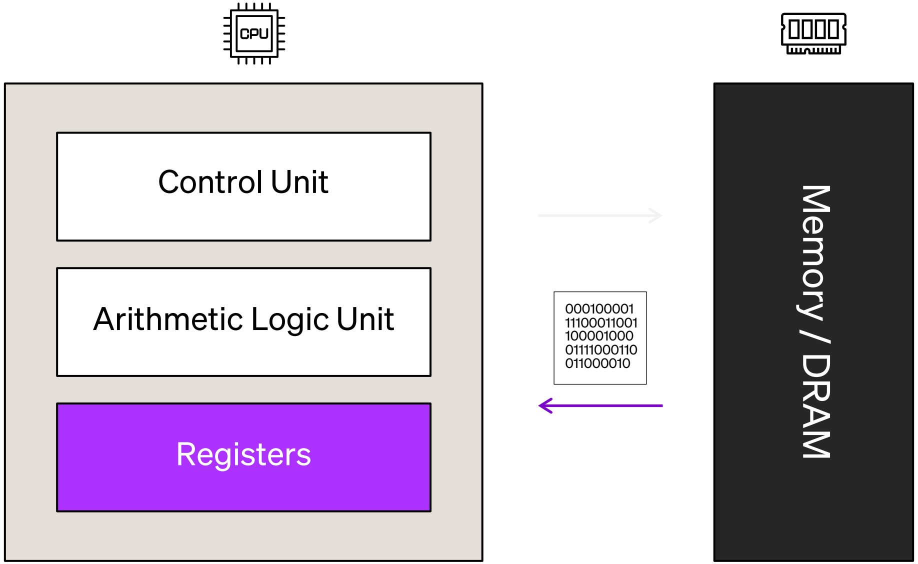
111100011001100001000011110001
10011000010000111100011001100

01000011110001100110000100001
111000110011000010000111100
01100110000100001111000110011

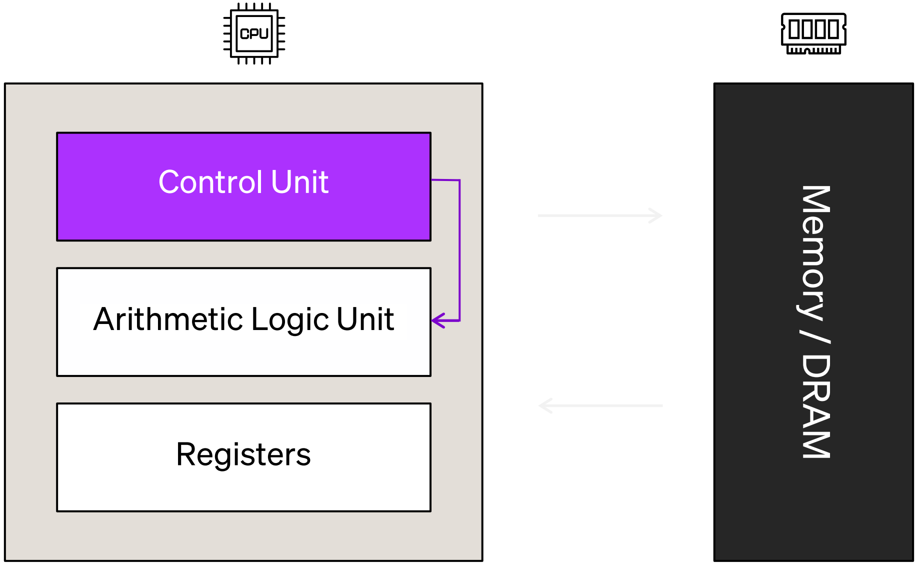
00001000001111000110011000010
000111100011

How does the CPU execute this machine code?

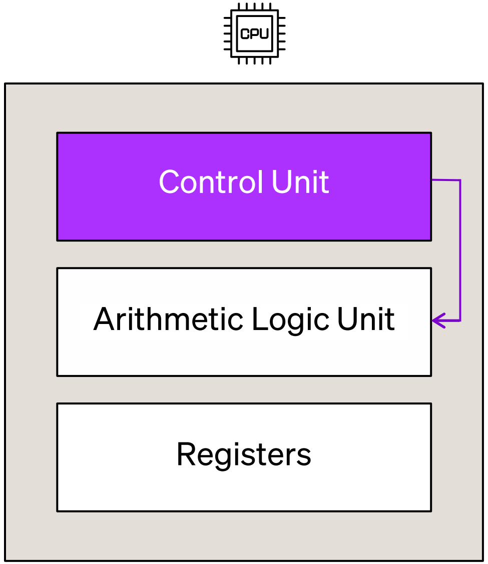
First, the CPU **fetches** the relevant machine code from memory and loads it into the instruction register, a unit of the CPU that is used to store data while processing



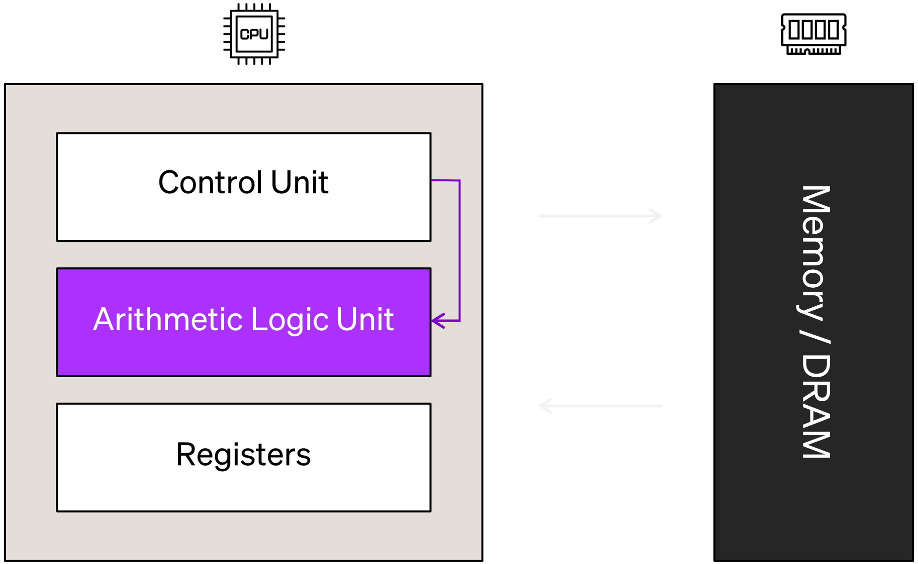
Then, the control unit of the CPU **decodes** the instruction and decides on the type of operation to perform and how this should be routed through the CPU



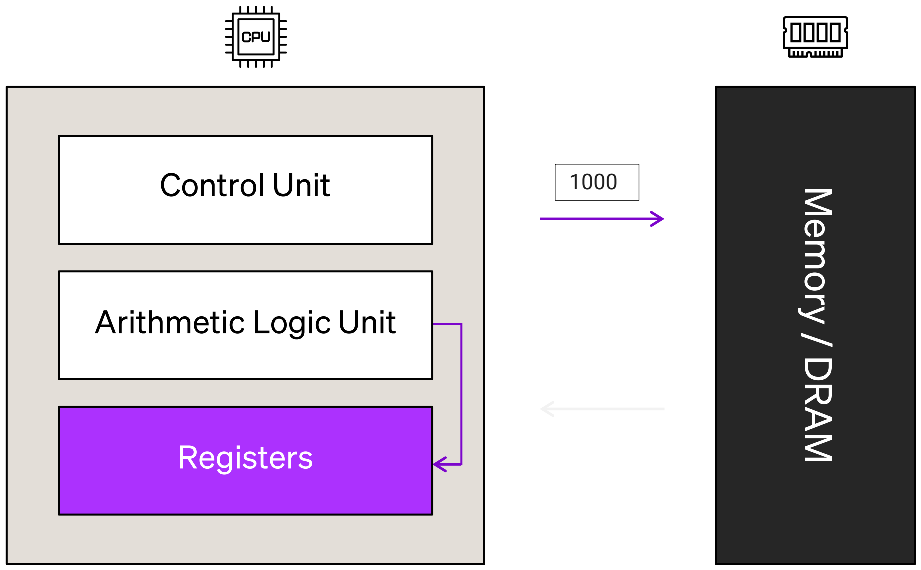
This is like the conductor of an orchestra, who directs which instruments should be playing during a performance



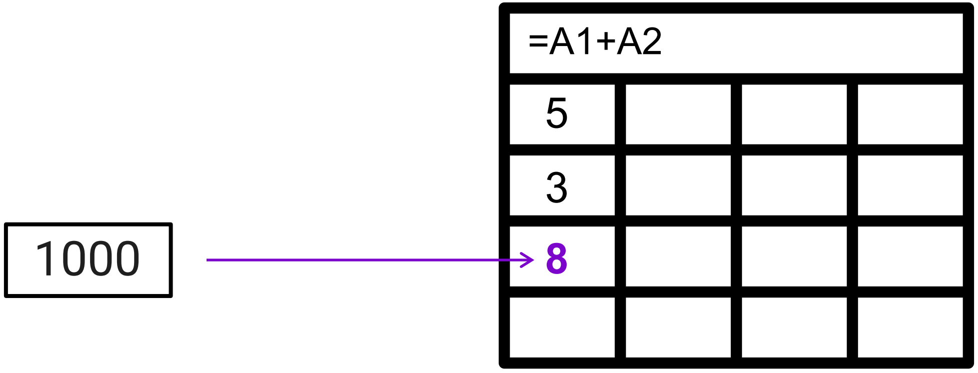
Once everything is set up, the operation is **executed** by running a series of calculations and logical operations in the arithmetic logic unit



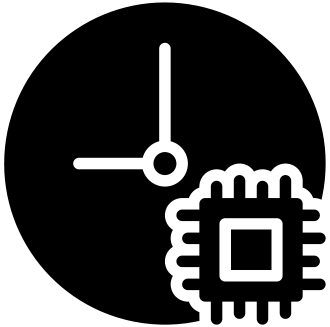
Once the operation is complete, the results of the operation are written back to the registers and/or memory

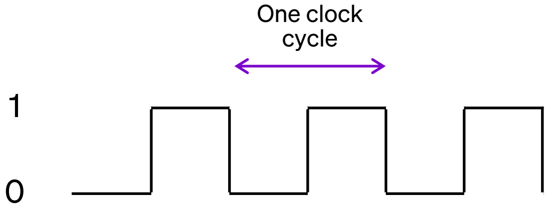


Finally, the Microsoft Excel application then retrieves this result from the memory, converts the binary data into the numerical solution and displays this in the relevant cell

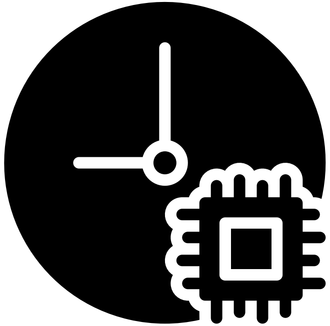


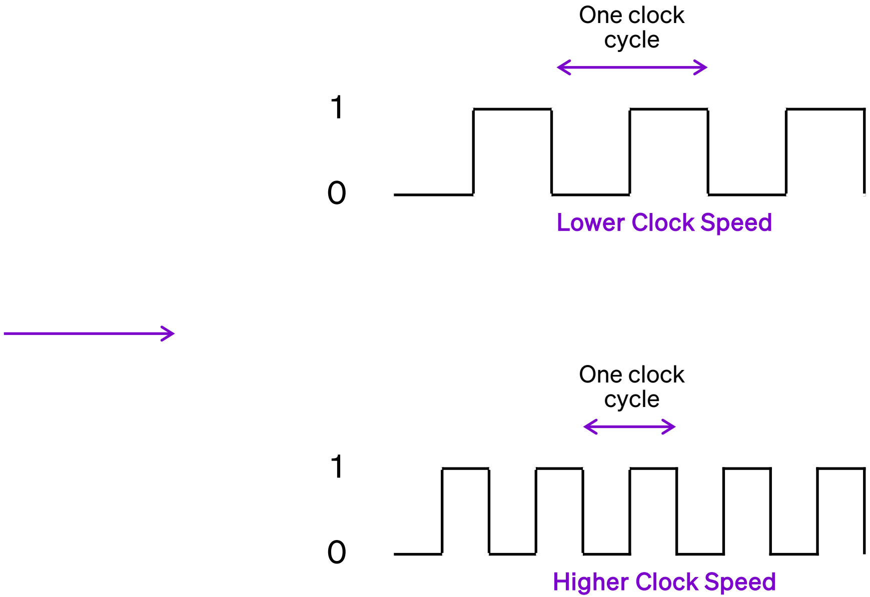
The speed at which a CPU can complete this cycle is determined by the 'clock', which works by sending a series of electric pulses that set the tempo for a computer's operations



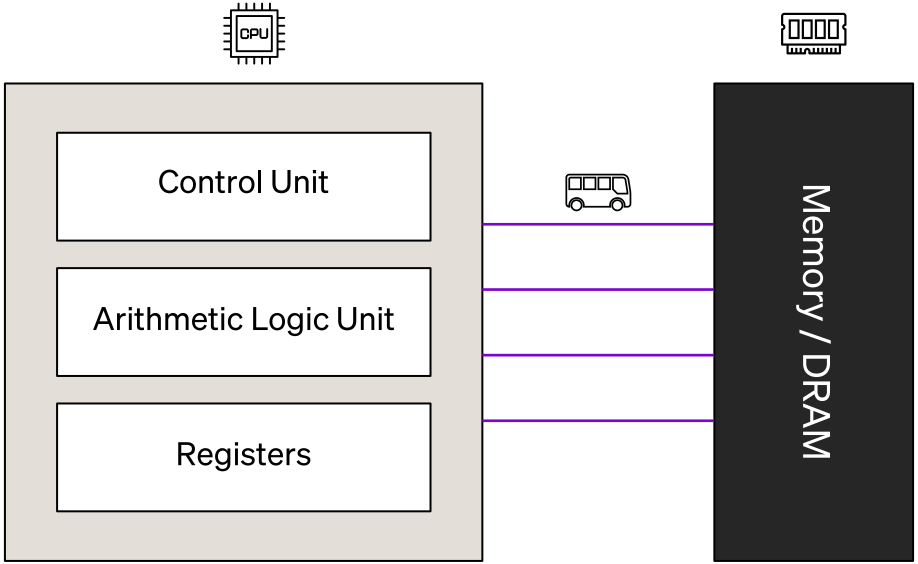


The frequency of the pulses is known as the ‘clock speed’ which is measured in hertz – the higher the clock speed, the faster the CPU can complete a set of instructions

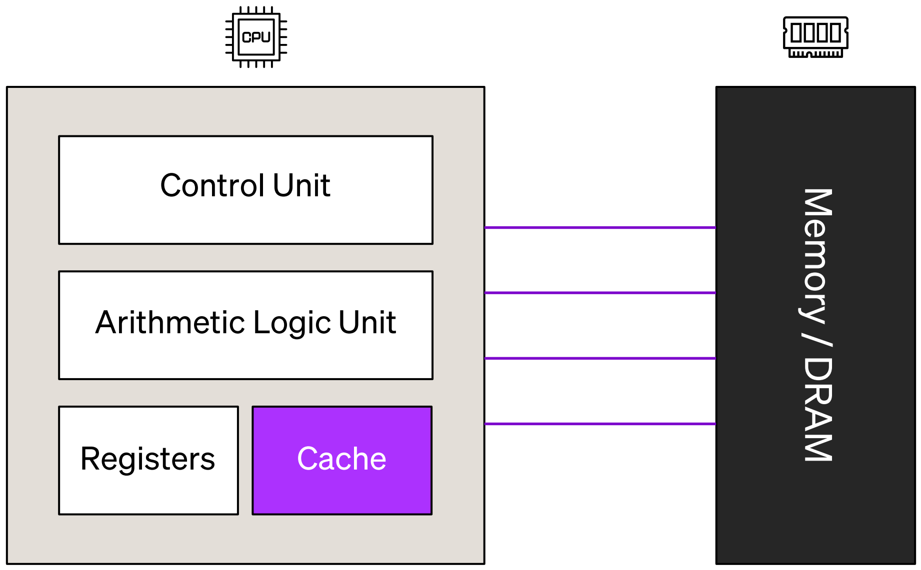




Processing time is also determined by the latency and bandwidth of the 'bus', a series of wires that transfer data and instructions between the memory and the CPU

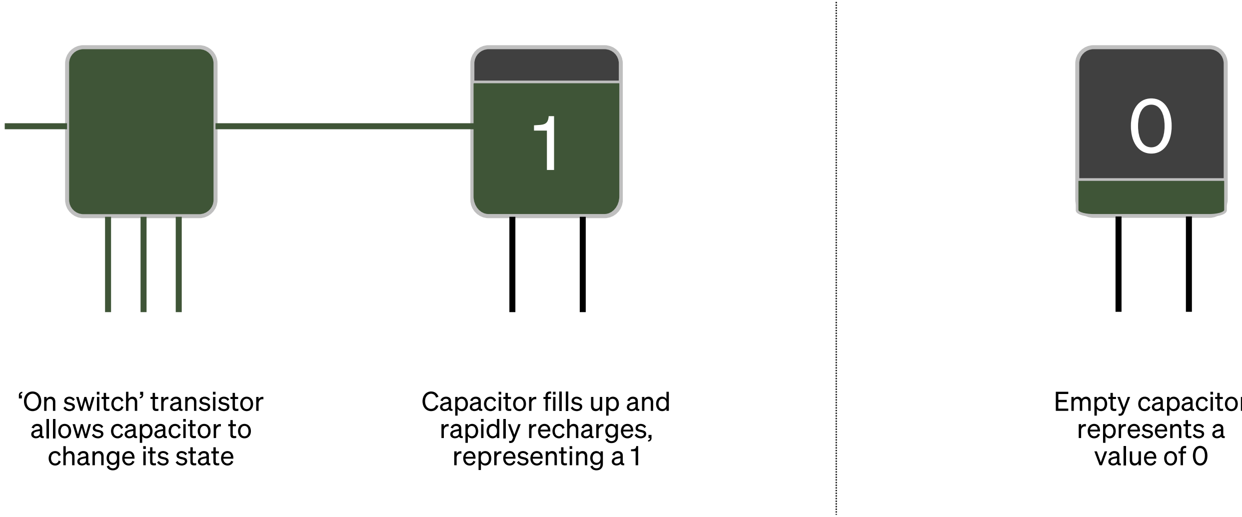


To minimize this latency, CPUs contain multiple levels of on-chip memory called 'caches' that are much faster than the main memory and store frequently used data and instructions



How are DRAM and cache different?

System memory, called ‘DRAM’, stores data using circuits of transistors and capacitors, which represent a binary digit ‘1’ when they hold charge

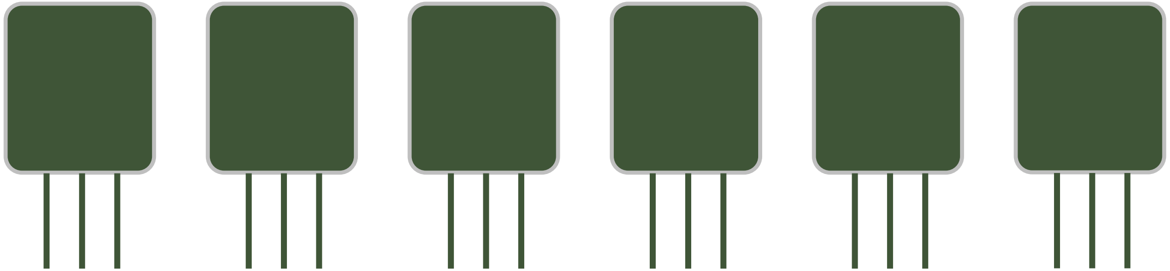


‘On switch’ transistor allows capacitor to change its state

Capacitor fills up and rapidly recharges, representing a 1

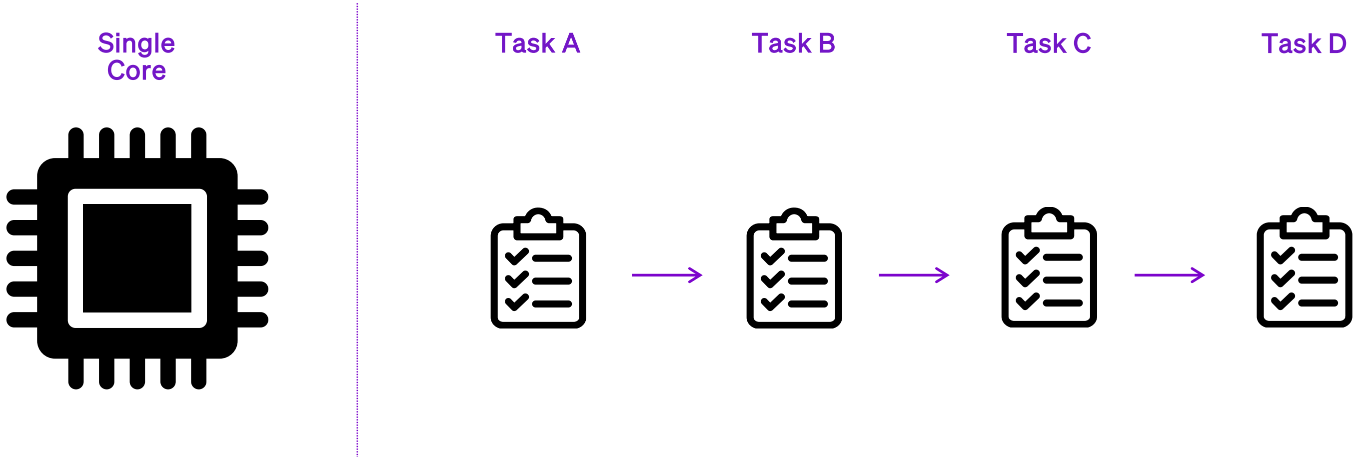
Empty capacitor represents a value of 0

Cache is a different type of memory called 'SRAM' which allows data to be fetched and decoded much faster by storing a single binary digit using a circuit of six transistors

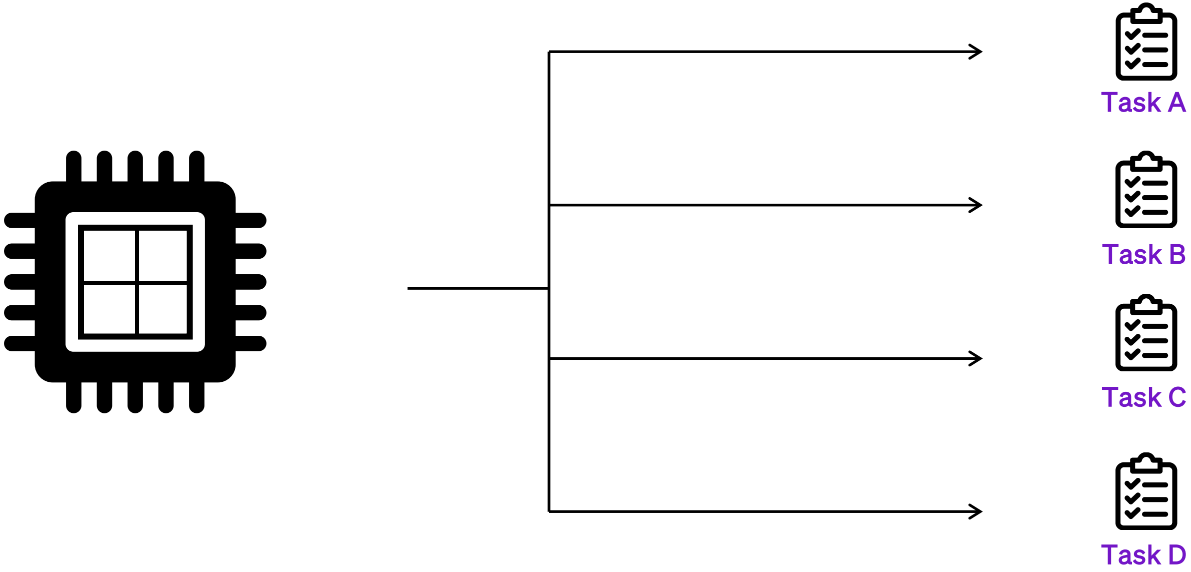


The process outlined above refers to the operations of a single microprocessor inside a CPU, called a ‘core’

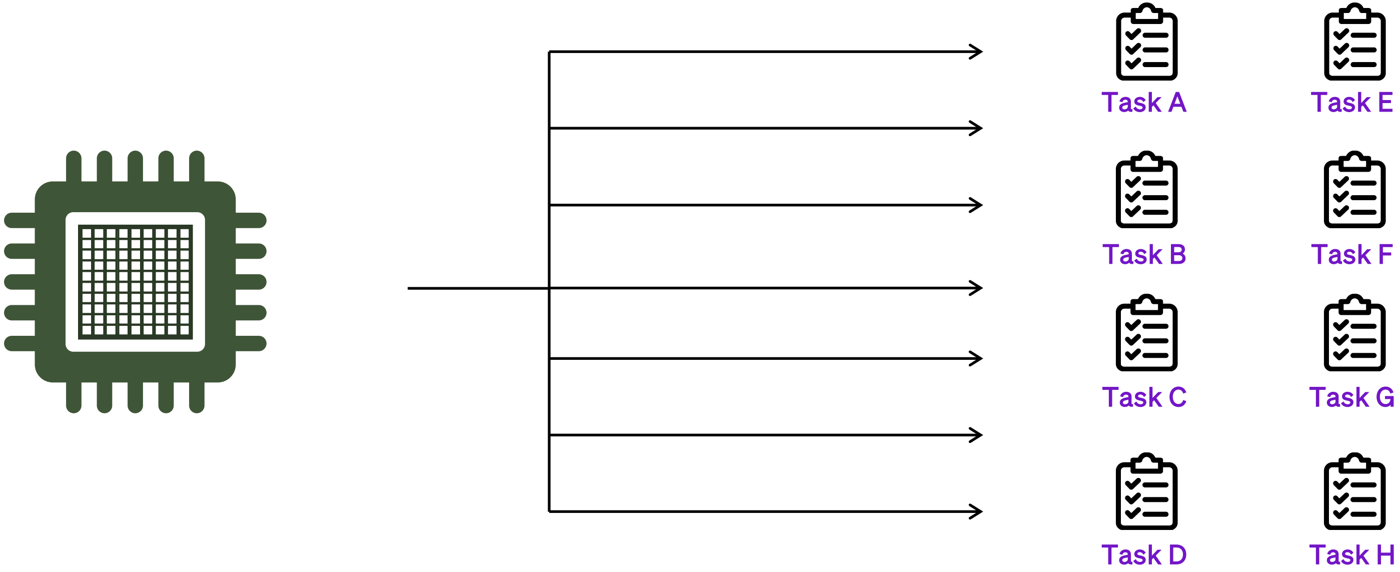
A single ‘core’ inside a CPU can process instructions serially, or one after the other



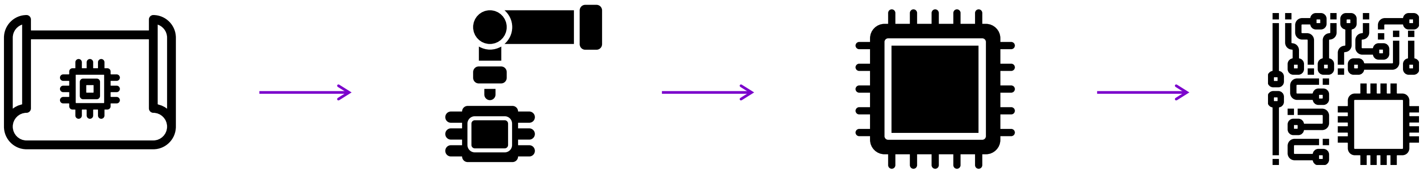
So, CPUs are often designed with multiple cores, which allows multiple tasks to be completed in parallel



Other types of processors, like GPUs, contain thousands of individual cores to process vast amounts of information in parallel



Building a microprocessor is a complex, multi-stage process



Design

Designing the circuit of a chip to fit a customer need

Manufacturing

Integrated circuits printed onto a sheet of silicon wafer

Packaging

Cutting the chips apart and packaging them in plastic or ceramic

Integration

Soldering a chip onto a circuit board or integrating into a product

This process begins with designing the underlying circuit of a chip

Design

Designing the circuit of a chip to fit a customer need

→

Manufacturing

Integrated circuits printed onto a sheet of silicon wafer

→

Packaging

Cutting the chips apart and packaging them in plastic or ceramic

→

Integration

Soldering a chip onto a circuit board or integrating into a product

Dive Deeper…

Further Reading & Watching

Watching:

- How a CPU Works (In One Lesson)
- The Central Processing Unit (Crash Course Computer Science)
- How Do Computers Read Code? (Frame of Essence)
- The Fetch-Execute Cycle: What's Your Computer Actually Doing? (Tom Scott)
- SRAM vs DRAM : How SRAM Works? (All About Electronics)

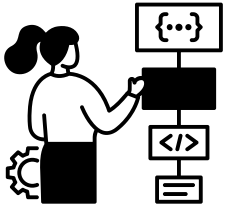
Reading:

- How Computers Work: The CPU and Memory (University of Rhode Island)
- How RAM Works (HowStuffWorks)

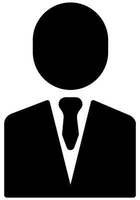
CHAPTER 04

Chip Design

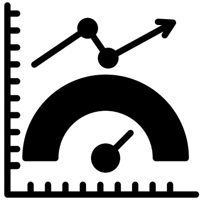
Designing a new chip begins with a ‘system architect’ working to define what the chip will do and what functions it will require to meet customer demands



System Architect…



Works with
business teams…



…to define what the chip
will require to meet
customer demands

Source: Anysilicon

This is a constant tradeoff between performance, size, power-efficiency and cost

Performance

Trying to achieve the highest processing speed while constrained by other factors

Size

Building a chip that is sufficiently powerful but small enough to fit in a device

Power-Efficiency

Building a chip that is sufficiently powerful but has a sufficiently long battery life

Cost

Optimizing for performance, size and power while minimizing design and production cost

Source: Richard (2023), Understanding Semiconductors

For some applications like desktop computers, performance and cost are the primary considerations

Performance

Trying to achieve the highest processing speed while constrained by other factors

Size

Building a chip that is sufficiently powerful but small enough to fit in a device

Power-Efficiency

Building a chip that is sufficiently powerful but has a sufficiently long battery life

Cost

Optimizing for performance, size and power while minimizing design and production cost

Source: Richard (2023), Understanding Semiconductors

But for other applications like mobile computing, size and power efficiency take precedence

Performance

Trying to achieve the highest processing speed while constrained by other factors

Size

Building a chip that is sufficiently powerful but small enough to fit in a device

Power-Efficiency

Building a chip that is sufficiently powerful but has a sufficiently long battery life

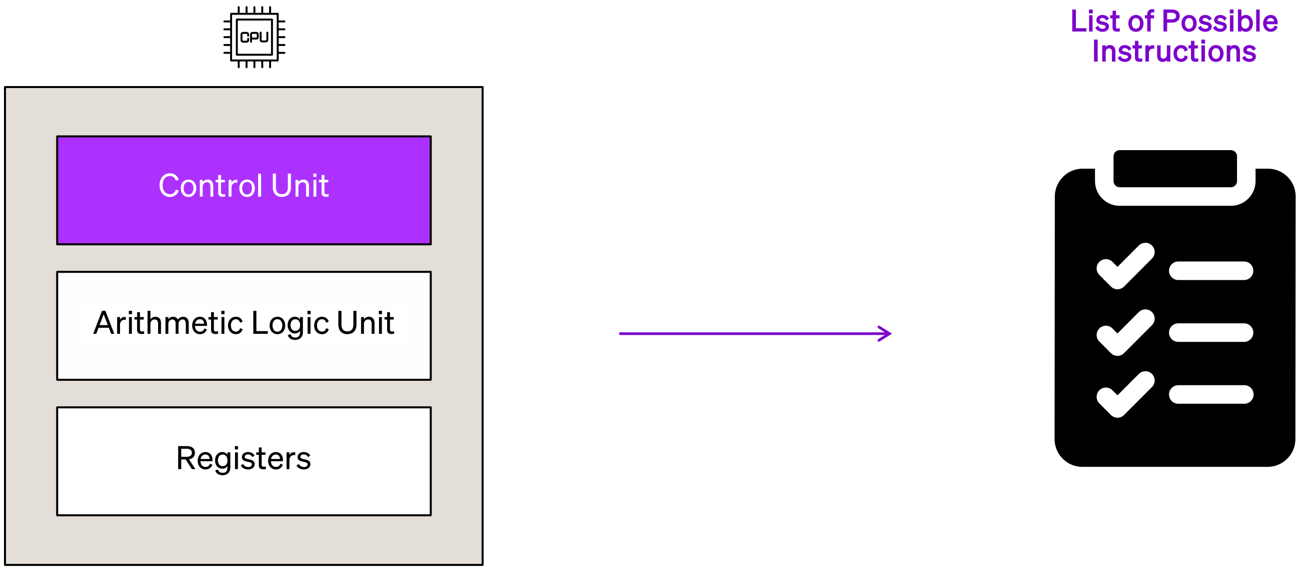
Cost

Optimizing for performance, size and power while minimizing design and production cost

Source: Richard (2023), Understanding Semiconductors

To optimize for these goals, chips are designed according to different types of ‘macro-architectures’ called instruction sets

The instruction set of a processor determines the collection of instructions that the control unit of a processor can execute



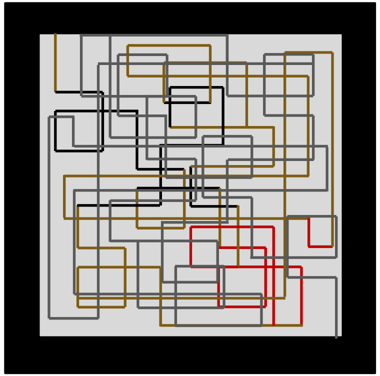
Source: Bina Bhatt, RISC vs CISC

Traditionally, processors were designed using complex instruction set computing (CISC) which consisted of longer instruction sets that could do more with each instruction

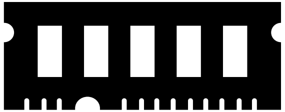
Source: Bina Bhatt, RISC vs CISC

This would require more complicated circuit designs that required more transistors and power to function, but less memory space to store instructions

Complex circuit design required to process more complex instructions



But fewer instructions were required per task, so less memory was needed to store these instructions



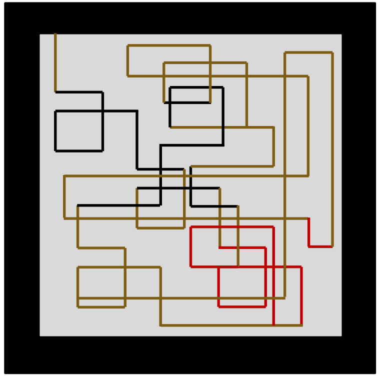
Source: Bina Bhatt, RISC vs CISC

As the cost of memory came down, reduced instruction set computing (RISC) became popular in the 1990s and consists of shorter, simpler instruction sets that do less with each instruction, but require more instructions to execute a task

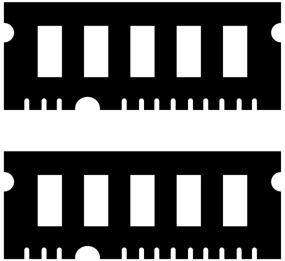
Source: Bina Bhatt, RISC vs CISC

These chips contain simpler circuit designs with fewer transistors, making them more power efficient and suitable for mobile computing applications

Simpler circuit design required to process simpler instructions

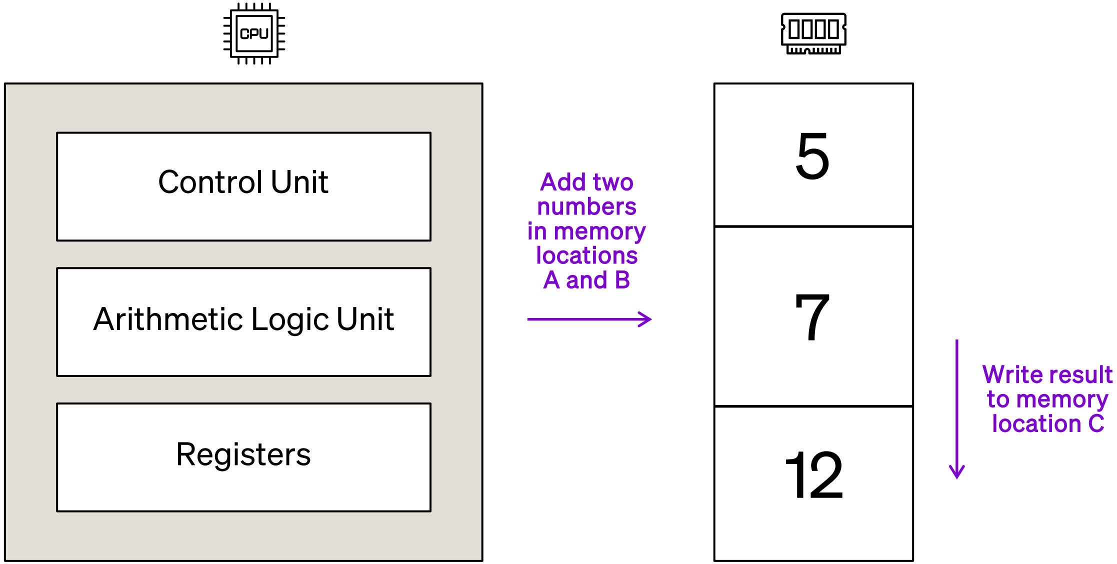


But more instructions required per task, so more memory is needed to store these instructions



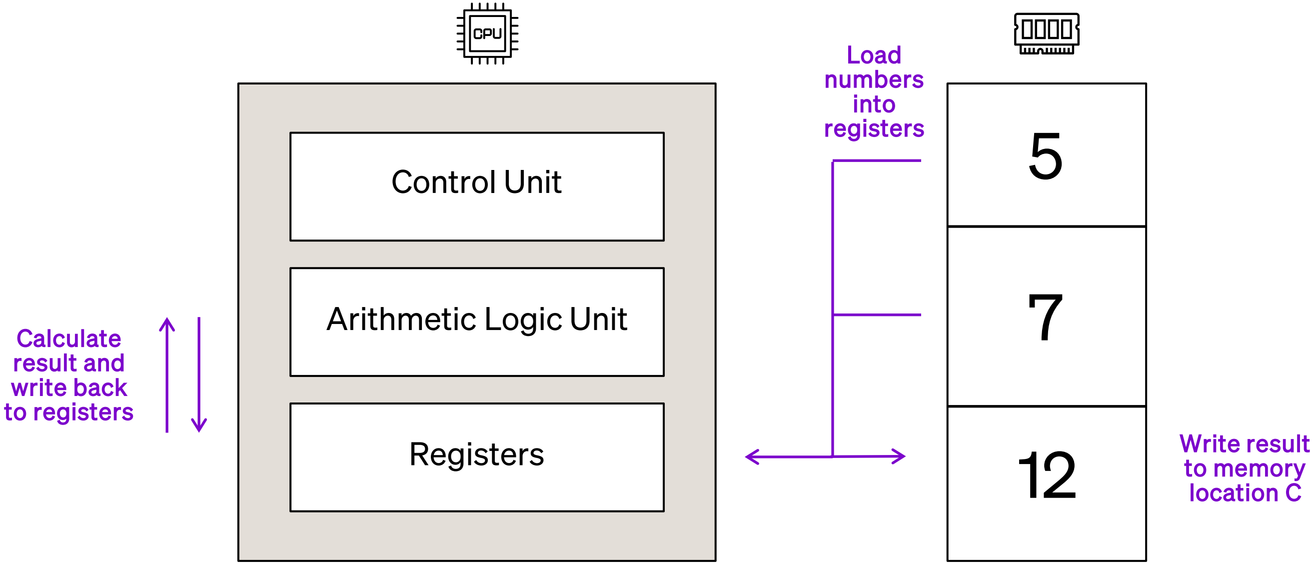
Source: Bina Bhatt, RISC vs CISC

For example, in a CISC architecture, an operation to add two numbers together can operate and write directly to data stored in memory using a single instruction

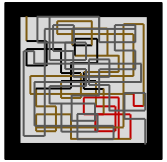


Source: Bina Bhatt, RISC vs CISC

In RISC architecture, operations require numbers to be first loaded into registers, then added together, and finally written back to memory using shorter, simpler instructions

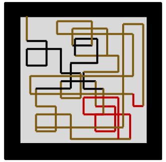


Each instruction set is optimized for different computing needs, and requires a different type of software design since software is compiled and executed differently



Complex Instruction Set Computing (CISC)

- More work done by the silicon vs code
- Less code space & less memory required
- Rich instruction set better for higher-level programming languages without complex compilers



Reduced Instruction Set Computing (RISC)

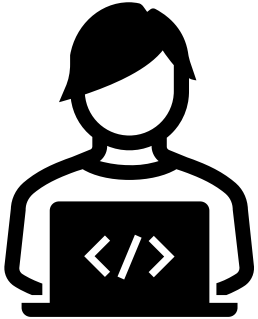
- More work done by code vs silicon
- More code space & more memory required
- Simpler hardware leads to less power consumption which is better for mobile devices and IoT

Source: Bina Bhatt, RISC vs CISC

Once an instruction set is decided upon, a ‘microarchitecture’ consisting of the individual circuitry that complies with the instruction set must be designed

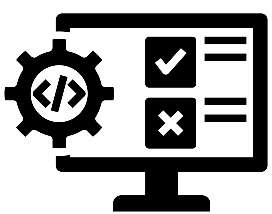
To do this, a design engineer uses a 'hardware description language' (HDL), which is similar to a programming language, to describe the desired behavior and structure of the chip

Hardware engineer describes the functionality of the circuit using a 'hardware description language' like VHDL or Verilog



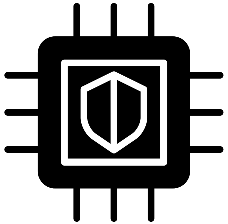
```
module AndGate(
  input wire a, // First input to the
AND gate
  input wire b, // Second input to
the AND gate
  output wire out // Output of the
AND gate
);

// Describe the behavior of the
AND gate
assign out = a & b;

endmodule
```

Once a chip is described in a hardware description language, other software and hardware tools can be used to simulate how the chip design will perform and test for errors



Universal Verification Methodology

Digital testbenches to run simulations for different parts of a chip’s design



Field Programmable Gate Arrays

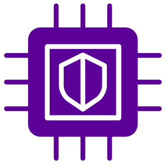
Types of chips that can be physically programmed to test the design of a chip

Source: Richard (2023), Understanding Semiconductors

Field programmable gate arrays (FPGAs) are a type of customizable chip that can be programmed to replicate and test the circuitry of the new chip design

Central Processing Unit

Primary component of a computer that executes a wide set of instructions



Field Programmable Gate Array

Reconfigurable processor that can be programmed by the user

Application Specific Integrated Circuit

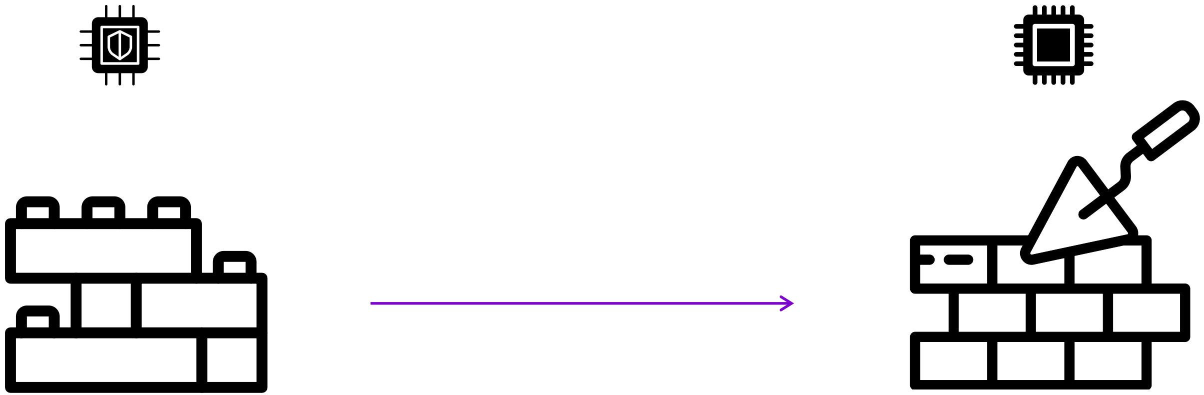
Custom-designed chip to perform a narrower range of tasks very efficiently

Graphics Processing Unit

Specialized processor for computing graphics and other parallel tasks

Source: Richard (2023), Understanding Semiconductors

This allows chip designers to test the designs of their chips on silicon using a reprogrammable circuit before sending their designs to be fabricated



FPGA allows designers to test reconfigurable designs, like lego blocks…

…which can then be sent to a foundry to be fabricated in a more permanent chip

After simulating the chip's functionality, another set of software tools are used to convert the chip's hardware description language into a physical design

Hardware Description Language

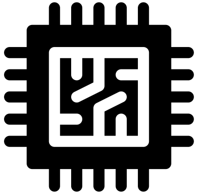
```
module AndGate(
   input wire a, // First input to the
AND gate
   input wire b, // Second input to
the AND gate
   output wire out // Output of the
AND gate
);

// Describe the behavior of the
AND gate
assign out = a & b;

endmodule
```

→

'Netlist' contains a list of all the components in the circuit and how they connect with each other

→

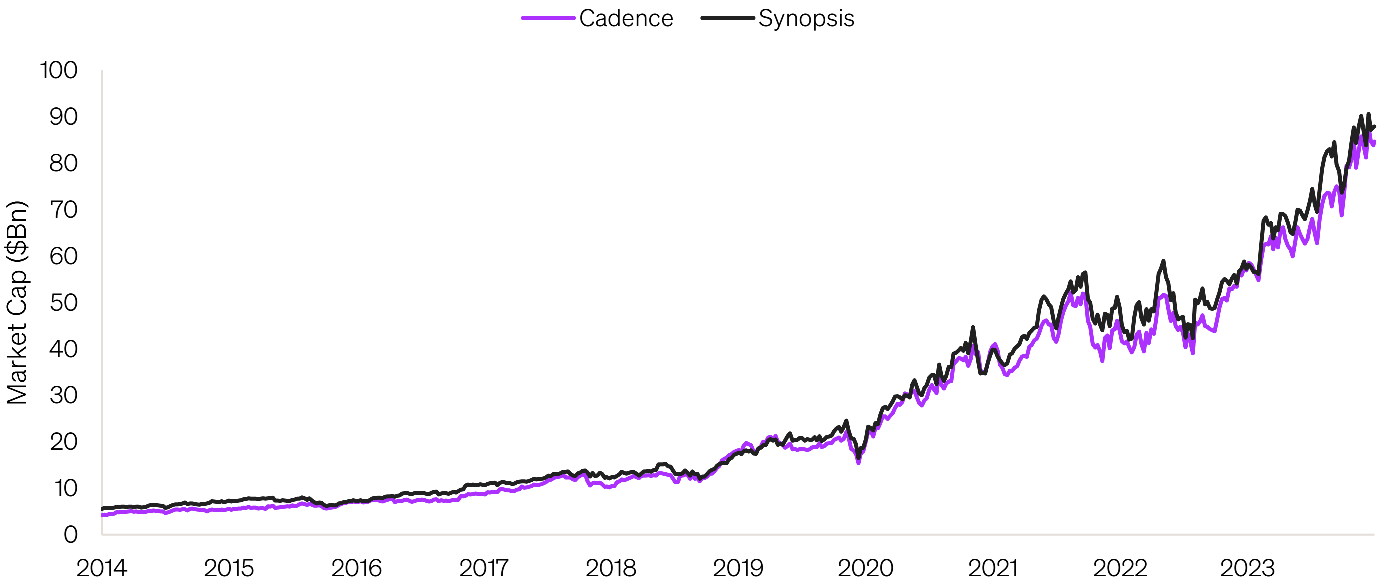
Physical Design of Chip



Source: Asianometry

Two main companies, Cadence and Synopsis, offer a wide range of electronic design and automation (EDA) software tools to design and verify chip circuits

Market Cap ($Bn)



Finally, the 'GDS' file containing the physical design of the chip is validated to ensure it can be manufactured by the chosen foundry, and sent to the manufacturer for fabrication

GDS file containing physical design of chip…

…sent to foundry to be fabricated

Source: Richard (2023), Understanding Semiconductors

How are chips manufactured?

Design

Designing the circuit of a chip to fit a customer need

Manufacturing

Integrated circuits printed onto a sheet of silicon wafer

Packaging

Cutting the chips apart and packaging them in plastic or ceramic

Integration

Soldering a chip onto a circuit board or integrating into a product

Dive Deeper…

Further Reading & Watching

Watching:

- Designing Billions of Circuits With Code (Asianometry)
- How VLSI Revolutionized Semiconductor Design (Asianometry)
- The Growing Semiconductor Design Problem (Asianometry)

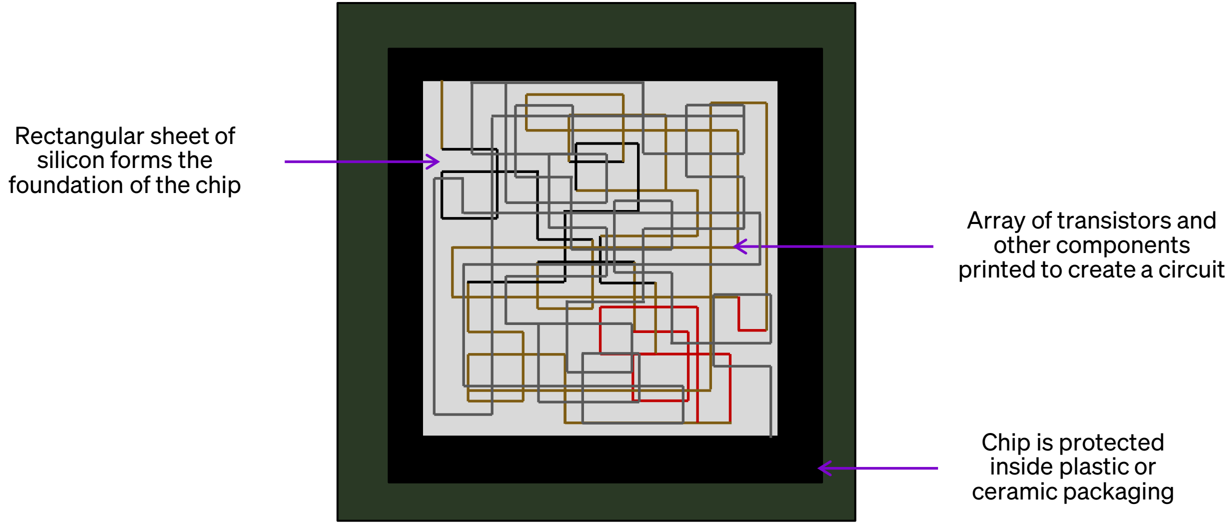
Reading:

- An Outline of the Semiconductor Chip Design Flow (Design & Reuse)
- IC Design and Manufacturing Process (Cadence)
- What is an FPGA? (Diligent)

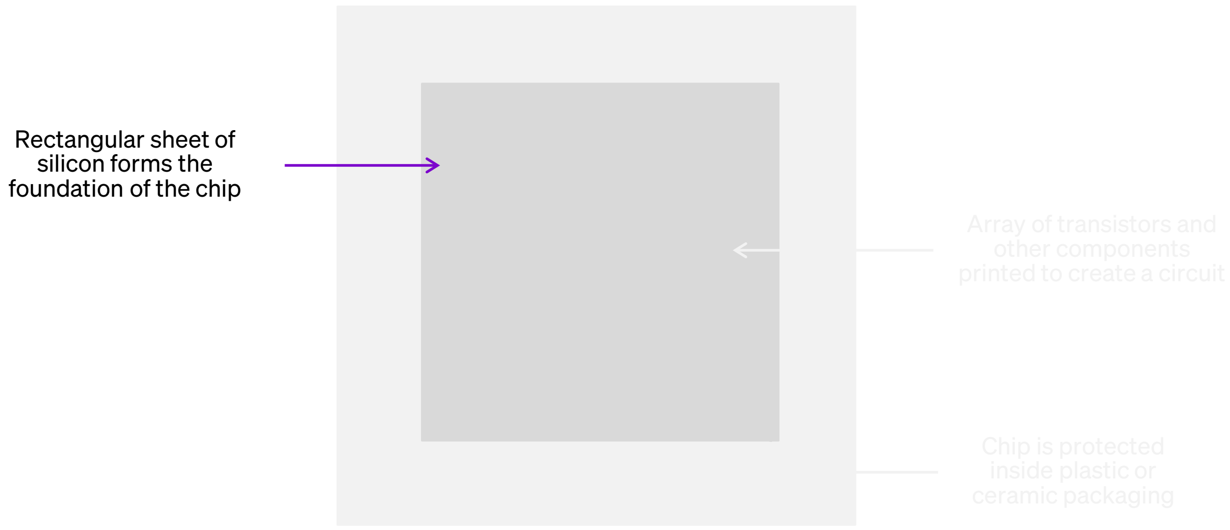
CHAPTER 05

Chip Manufacturing

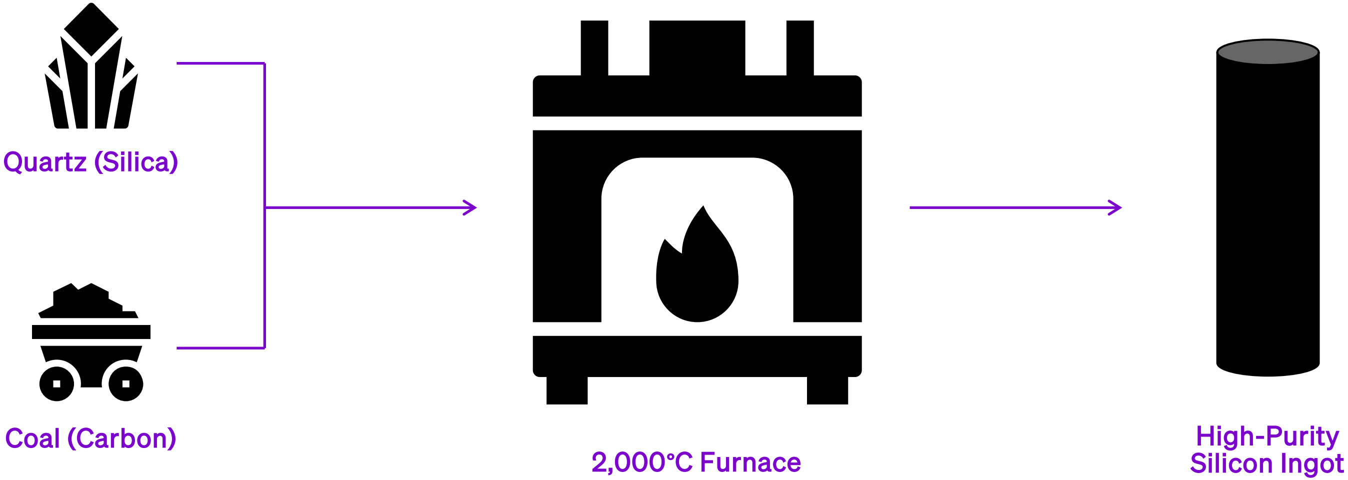
Microprocessors contain an array of transistors and other components printed onto a sheet of silicon called a die, and encased in plastic or ceramic packaging



The process of manufacturing a microprocessor begins with preparing the silicon, which becomes the foundation for the rest of the chip

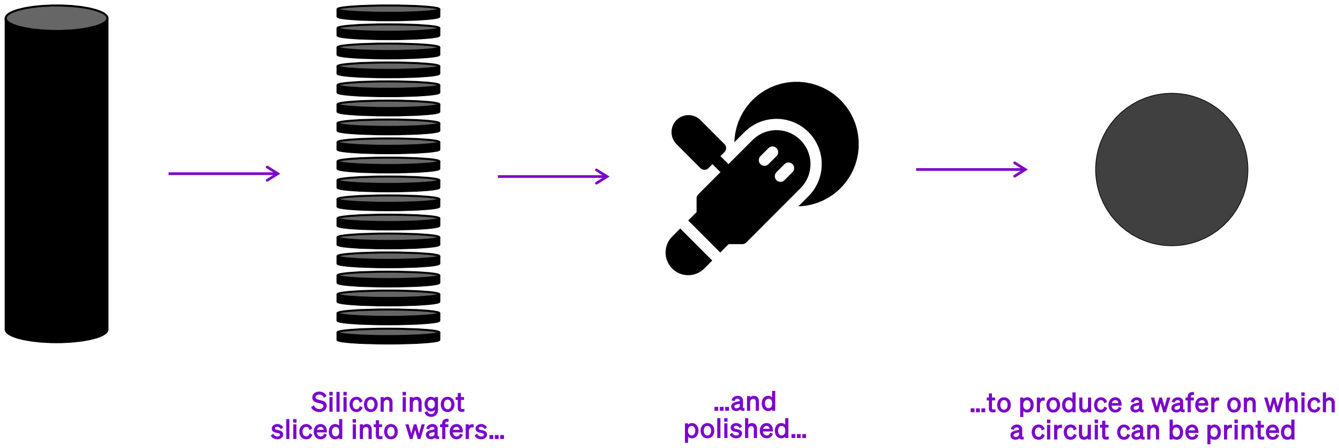


Silicon wafers are produced by first heating a mixture of quartz and carbon in a furnace to create a silicon rod called an ‘ingot’



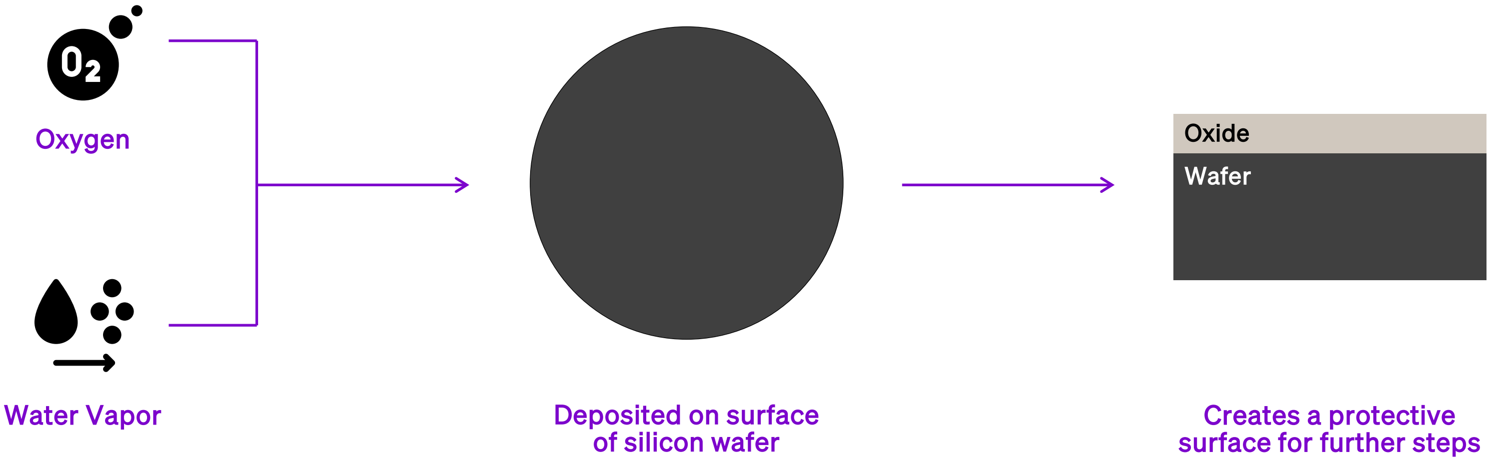
Source: Samsung

Then, the ingots are sliced into thin disks and polished to produce the silicon ‘wafers’ on which transistor arrays are printed

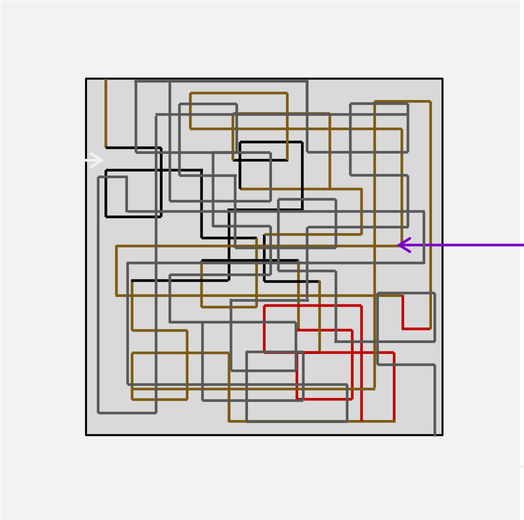


Source: Samsung, Asianometry

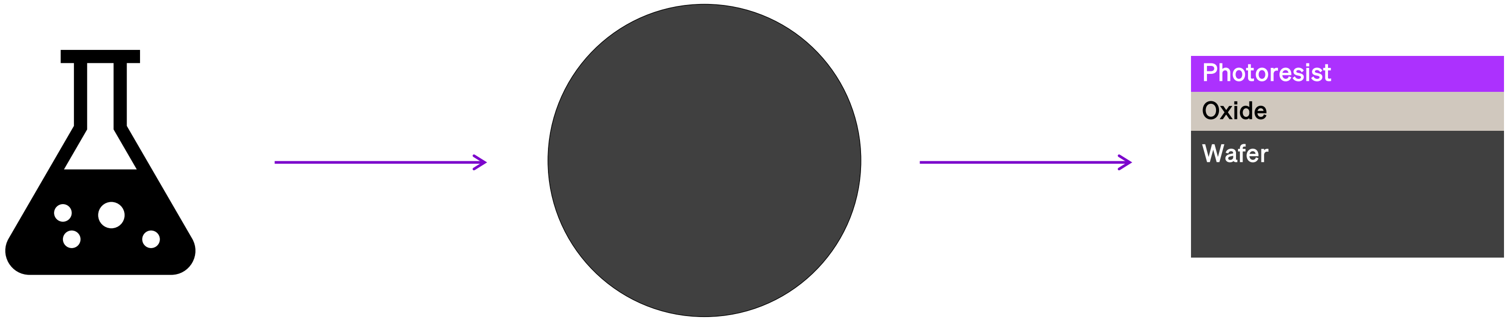
Oxygen or water vapor are deposited onto the silicon wafer to build up an oxide film that protects the surface of the wafer and prevents current leakage between circuits



Once the silicon is prepared, we then need to print the circuit of components onto the silicon wafer



To do this, the wafer is then coated with a thin layer of 'photoresist', a light-sensitive chemical used to trace a circuit design onto the wafer

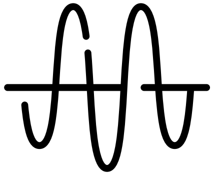


Light-sensitive chemical (photoresist)

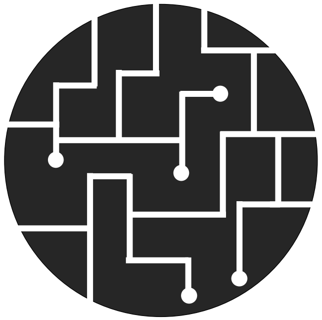
Deposited on surface of silicon wafer

Used to trace a circuit design onto the wafer

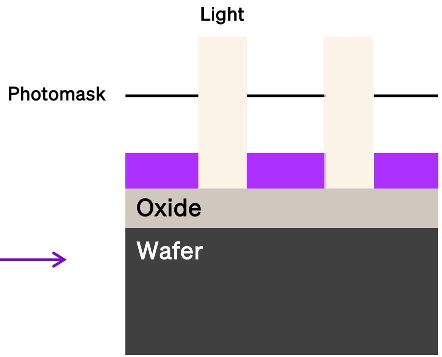
Ultraviolet rays are passed through a 'photomask' stencil containing an outline of the chip's circuit to draw the chip's design into the photoresist in a process called 'lithography'



UV light with a very small wavelength…



…is passed through a 'photomask' stencil…

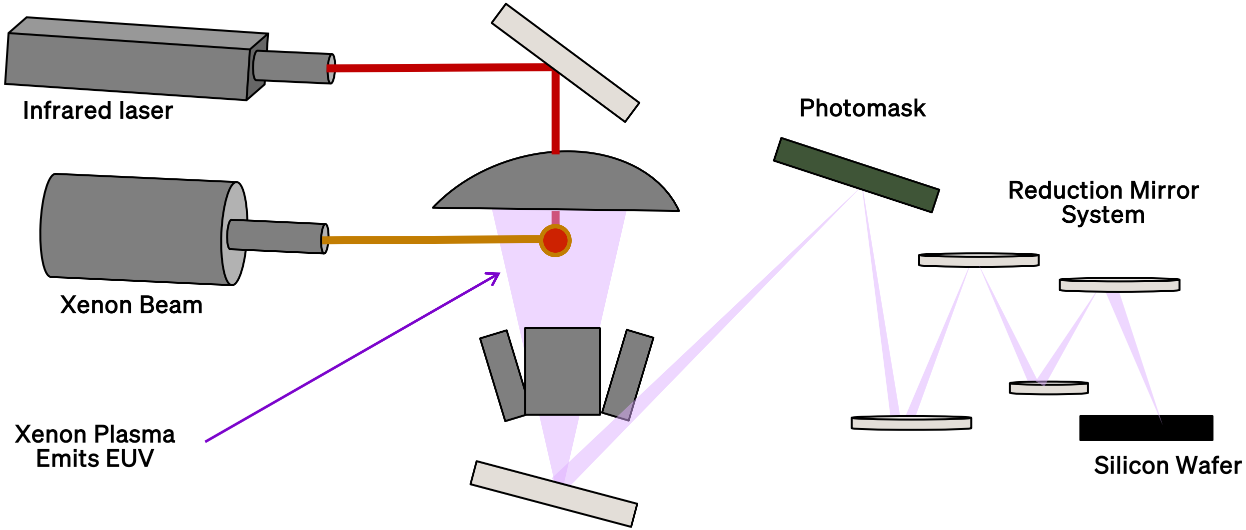


…to draw the circuit onto the surface of the wafer

Source: Samsung, Asianometry

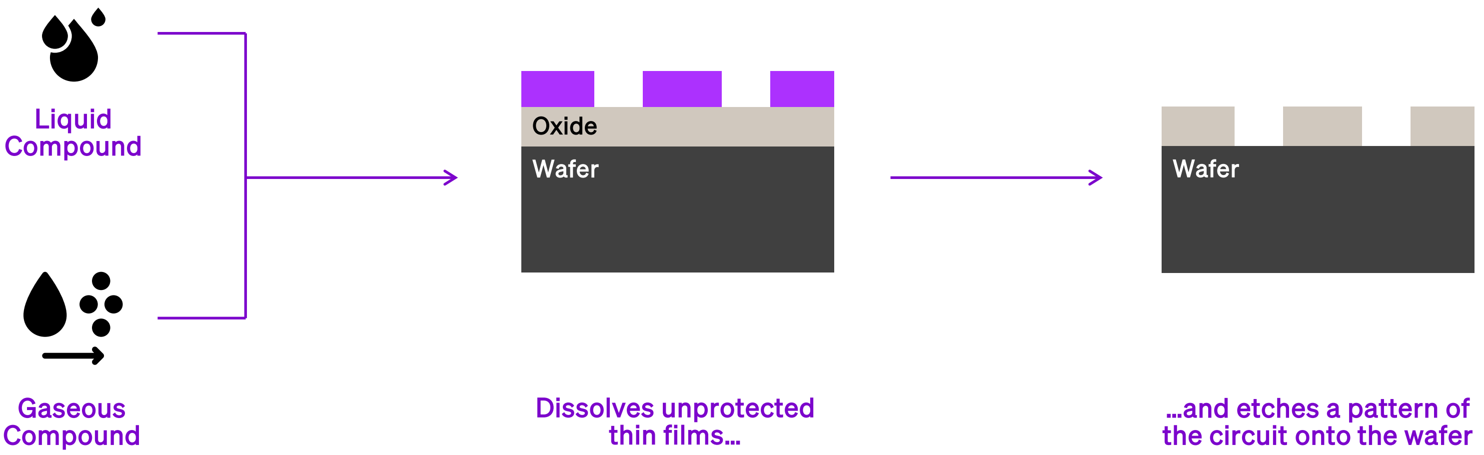
The wavelength of traditional 'deep ultraviolet light' is too wide to print cutting-edge chips, so 'extreme ultraviolet light' is generated to produce thinner wavelengths for lithography

How Extreme Ultraviolet Light (EUV) is Generated

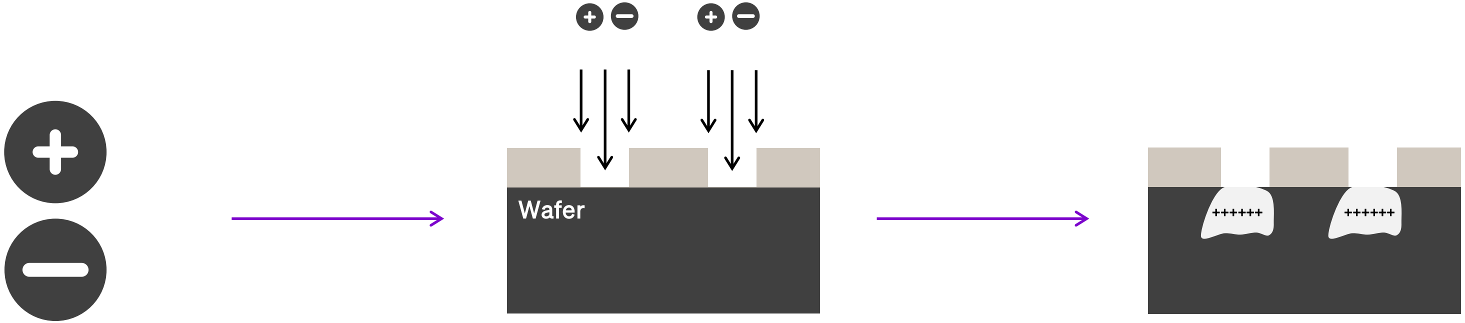


Source: BRG – A Brief Overview of Nanotechnology

Unnecessary materials are then carved out from the layers of the wafer using liquid or gaseous compounds to dissolve the photoresist and etch the circuit pattern onto the wafer



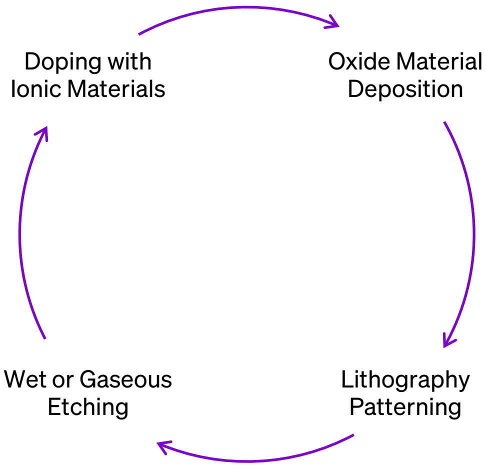
To improve conductivity by introducing positive and negative charges, impurities called ‘dopants’ are shot under the surface of the wafer in a process called ‘ion implantation’



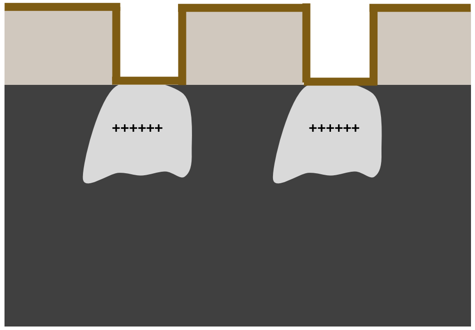
Positive and negative ions called dopants… | …are shot under the surface of the wafer… | …to improve control and conductivity of the circuitry

This process is repeated multiple times to etch many different layers of circuits into the chip, and then metallic interconnects are deposited to create wires for electricity to flow

This process is repeated many times…

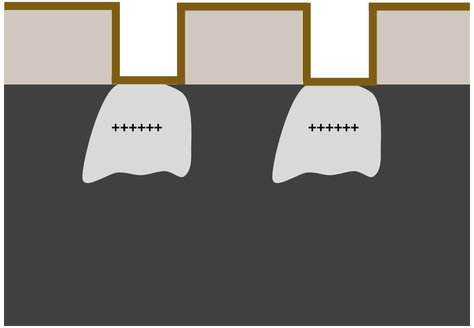


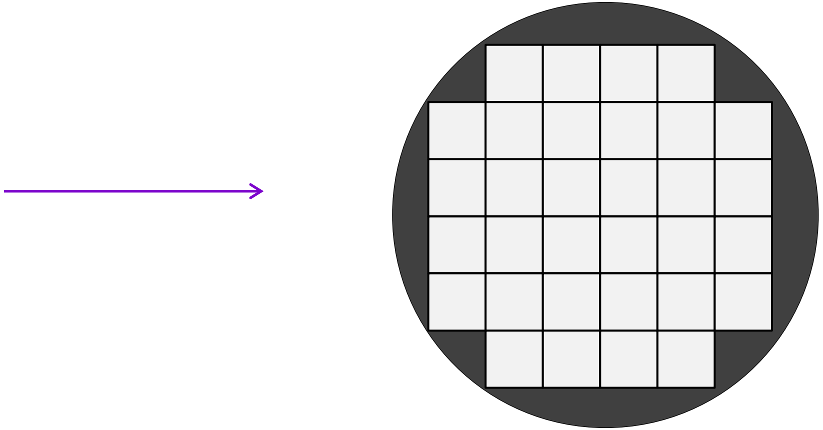
…and thin metal film is deposited to connect the circuitry together



Source: Richard (2023), Understanding Semiconductors

The final output of this process is a silicon wafer containing many hundreds of individual die





Each wafer typically contains small defects due to contaminants, so die sizes are often kept smaller with more chips on each wafer to achieve higher chip yields

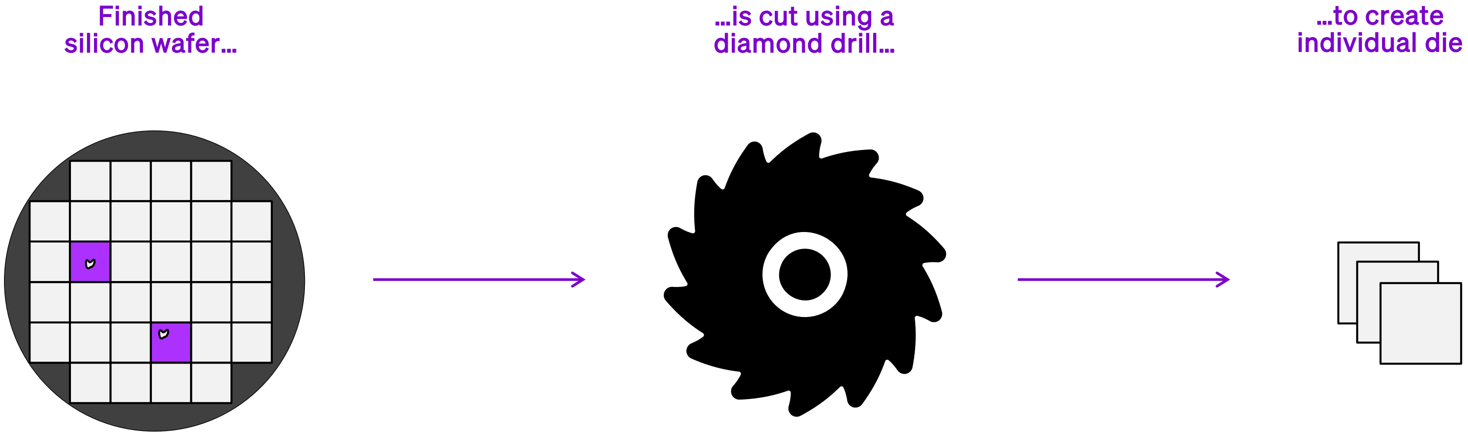
Defects result in
~78% yield (7/9)

Defects result in
~94% yield (30/32)

SOCIAL CAPITAL_

Source: Richard (2023), Understanding Semiconductors

To create chips that can be used in devices, each die must first be sliced from the wafer using a diamond drill....



…and then packaged in plastic or ceramic to protect the chip and create interconnects to integrate it into the rest of the system

Design

Designing the circuit of a chip to fit a customer need

Manufacturing

Integrated circuits printed onto a sheet of silicon wafer

Packaging

Cutting the chips apart and packaging them in plastic or ceramic

Integration

Soldering a chip onto a circuit board or integrating into a product

Dive Deeper…

Further Reading & Watching

Watching:

- Semiconductor Manufacturing Process Explained (Samsung)
- What Goes On Inside a Semiconductor Wafer Fab (Asianometry)
- Intel Fab Tour! (Linus Tech Tips)
- Why The World Relies On ASML For Machines That Print Chips (CNBC)

Reading:

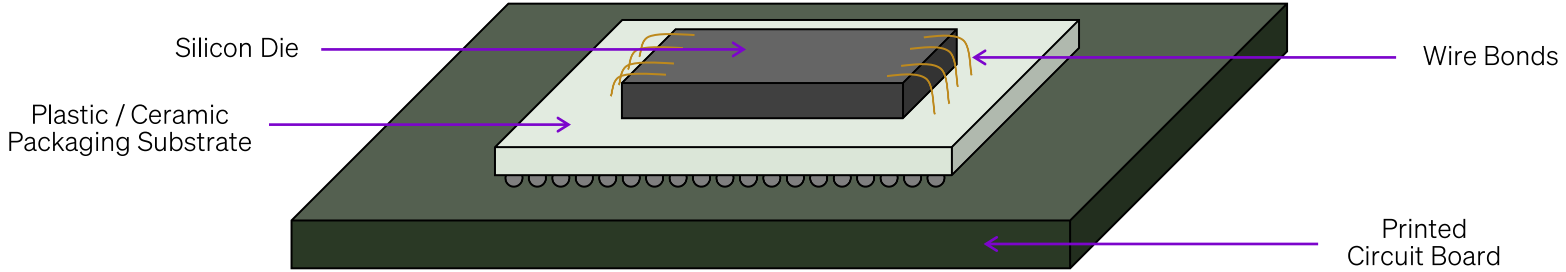
- Understanding Semiconductors: A Technical Guide for Non-Technical People (Corey Richard)
- Embracing Chaos: The Imperfect Art of Semiconductor Manufacturing And Lithography (SemiAnalysis)

CHAPTER 06

Chip Packaging

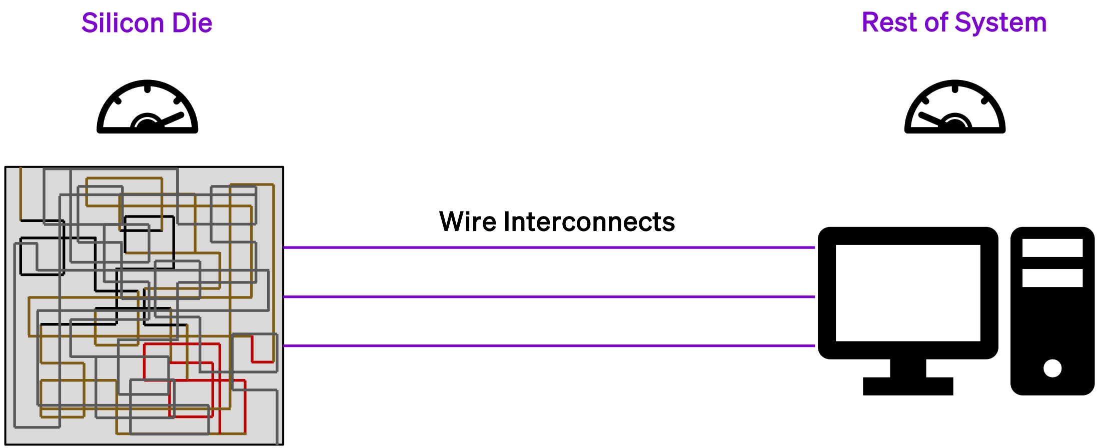
The goal of packaging is to protect a chip from the outside world, and build interconnects to integrate it into a broader system

Traditionally, packaging was developed using wire bonding, where the active area of a silicon die was connected to the rest of the system using metal wire bonds



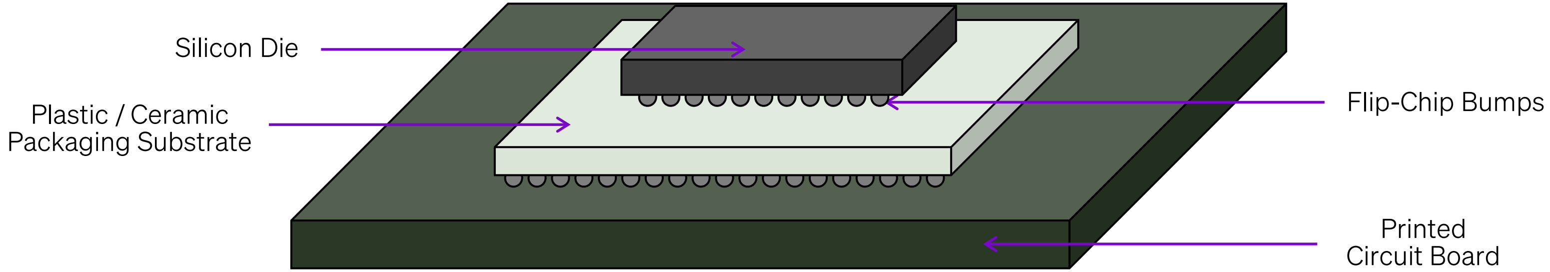
Source: Asianometry, Richard (2023)

However, wire bonding reduced the number of possible interconnects between the chip and the rest of the system, creating a bottleneck that reduced data transfer speeds



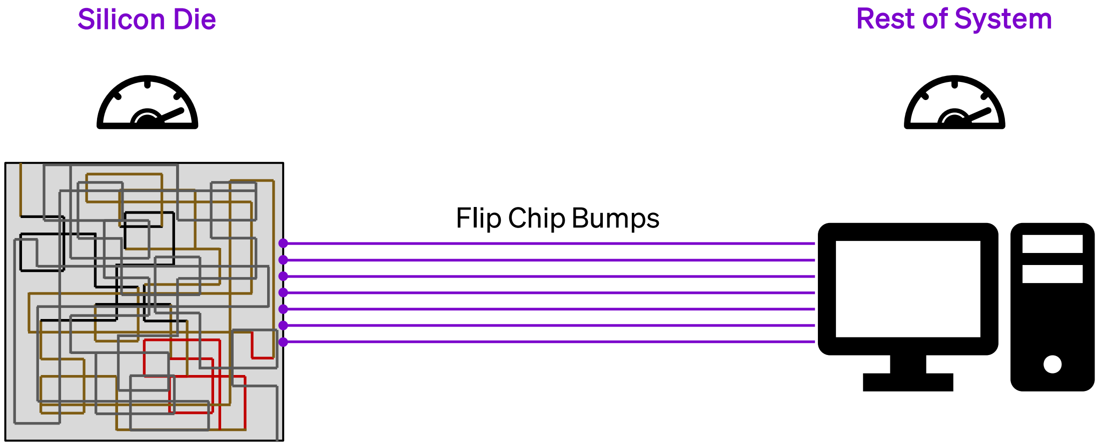
Source: Asianometry, Richard (2023)

To increase the contact area of the interconnects, 'flip-chip' packaging turned over the silicon die and added small balls of solder to serve as interconnects instead of wires



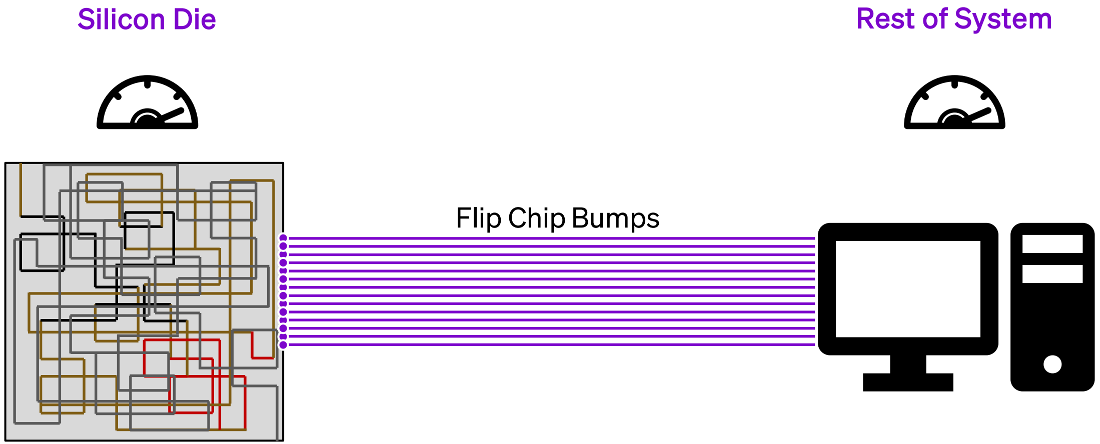
Source: Asianometry, Richard (2023)

This reduced latency and improved data transfer speeds between the silicon die and the rest of the system



Source: Asianometry, Richard (2023)

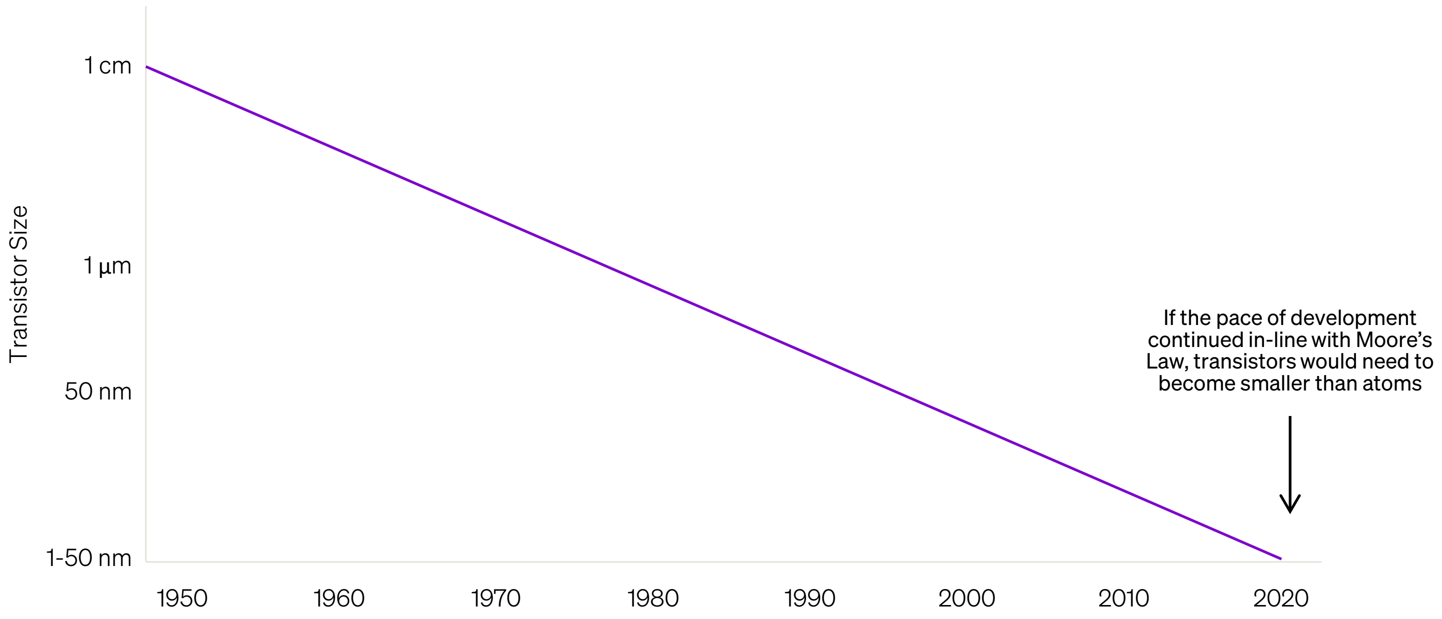
As transistor count and die performance has increased, companies have responded by increasing the number of bumps to minimize the bottleneck caused by interconnects



Source: Asianometry, Richard (2023)

But Moore's law is slowing as physical constraints restrict how small transistors can get

Transistor Size Trend Line According to Moore's Law



Source: Pokropivny et al. (2007), Introduction to Nanomaterials and Nanotechnology

Dive Deeper…

Further Reading & Watching

Watching:

- A Brief History of Semiconductor Packaging (Asianometry)
- Semiconductor Packaging Explained (Samsung Semiconductor)

Reading:

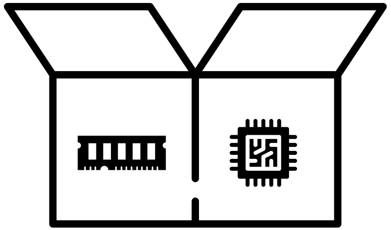
- Packaging to Protect the Chips from External Elements (Samsung Semiconductor)

CHAPTER 07

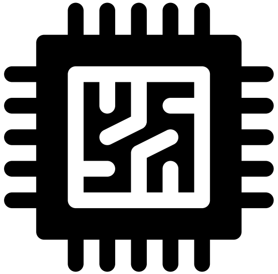
Overcoming Moore's Law

As Moore's law slows, companies are experimenting with advanced packaging methods and custom chip designs to improve performance without increasing transistor count

Chips can be packaged closely with components like memory to improve performance

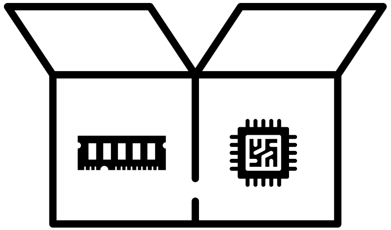


New types of custom chips can be optimized for specific tasks



The goal of advanced packaging is to integrate different components within a system more closely to reduce latency and improve performance

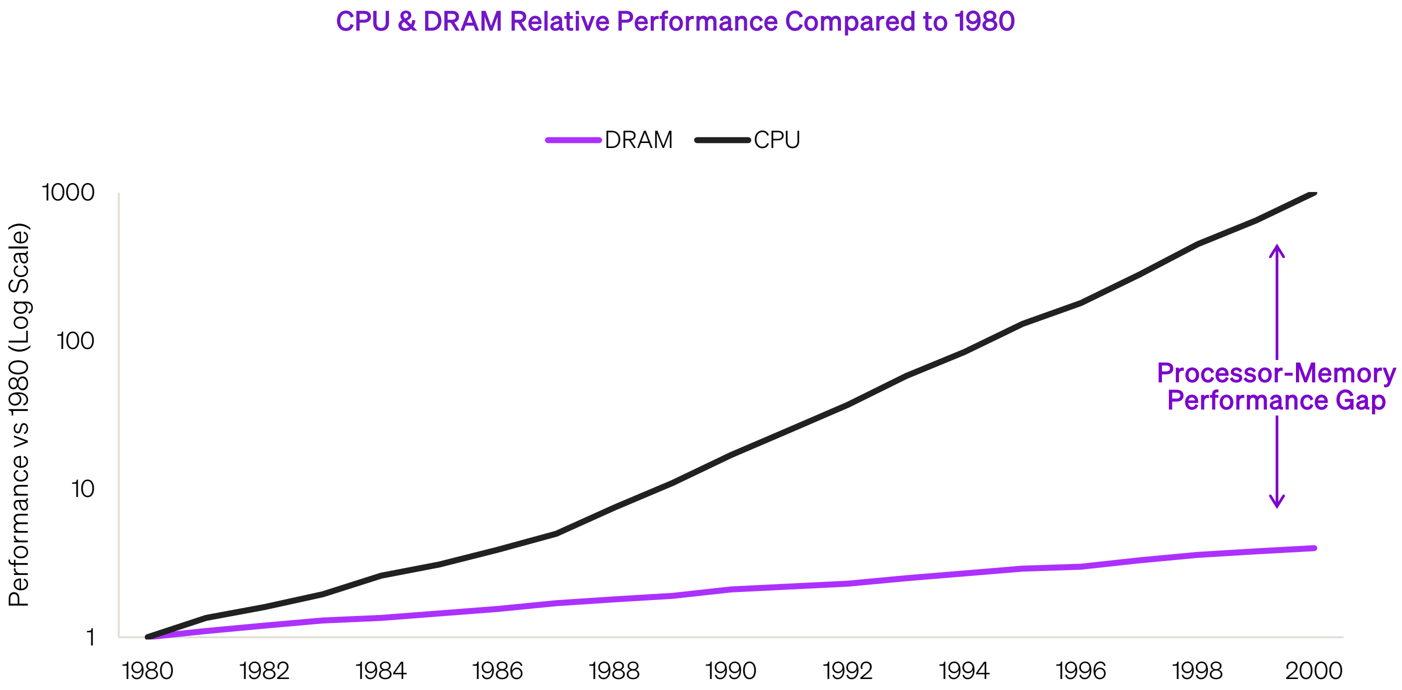
Chips can be packaged closely with components like memory to improve performance



New types of custom chips can be optimized for specific tasks

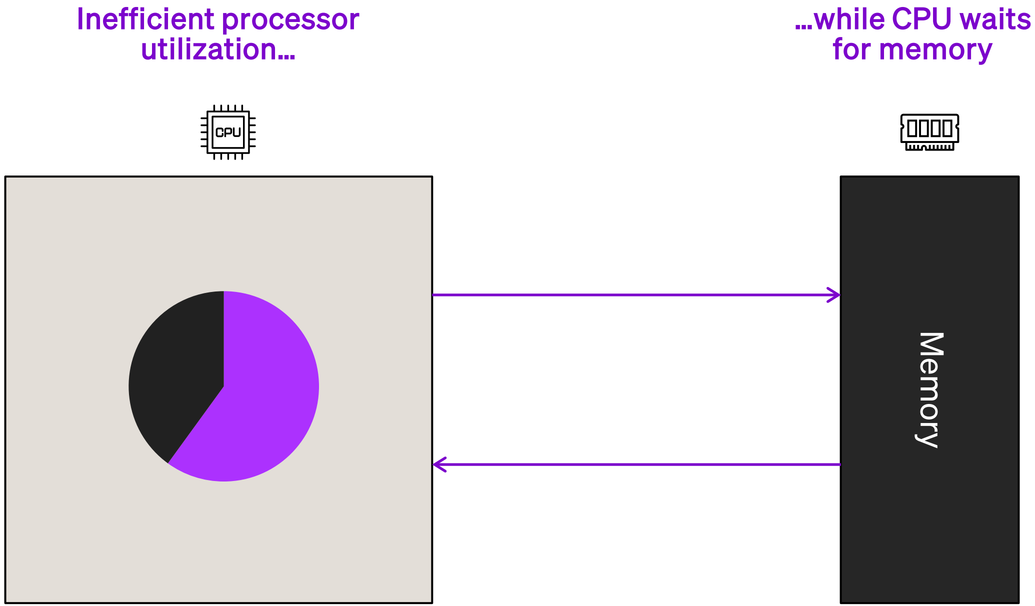
Advanced Packaging

Since the early 1980s, the relative performance of compute has far outpaced memory, which has suffered from low latency



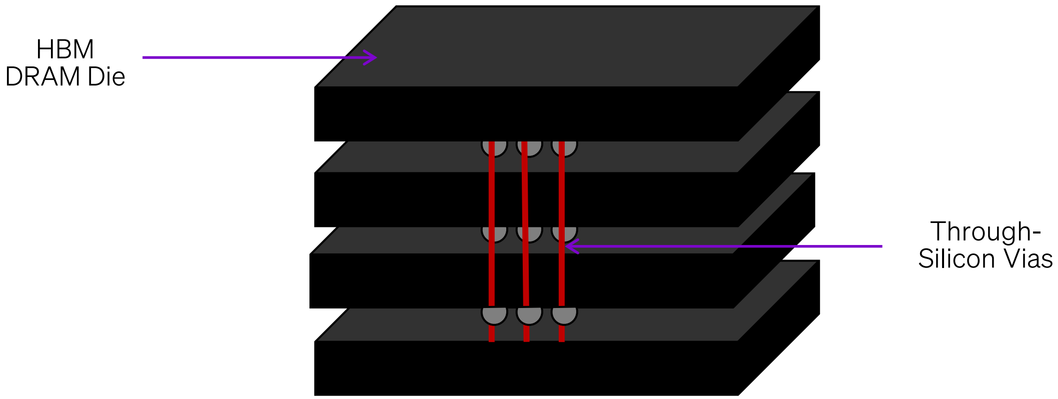
Source: Wang et al. (2023), Research Progress in Architecture and Application of RRAM

This has led to inefficient processor utilization as idle periods emerge while the CPU waits for instructions and data from the memory, called the 'Von Neumann bottleneck'



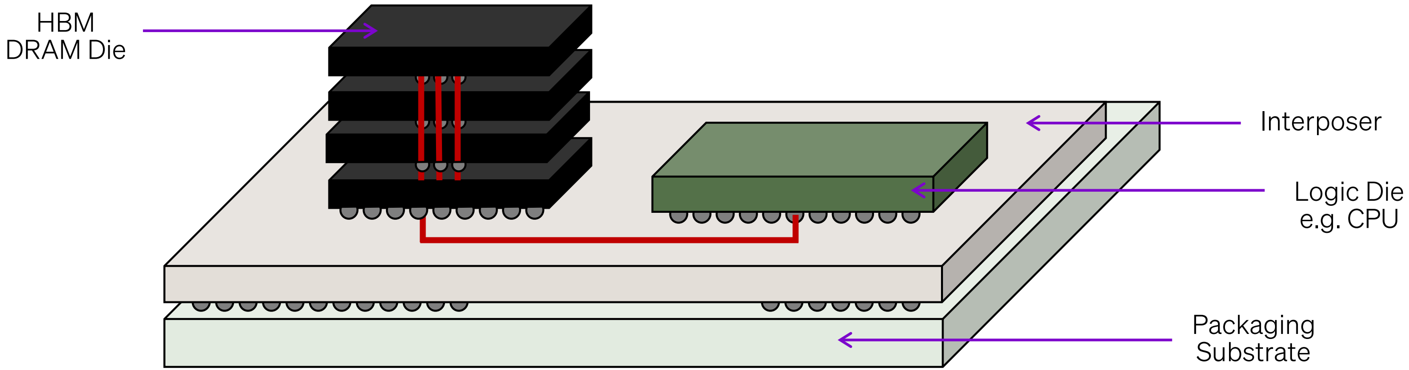
Source: TechTarget

To overcome these bottlenecks, new techniques stack memory dies on top of each other and connect them together using ‘through-silicon vias’ to create high bandwidth memory

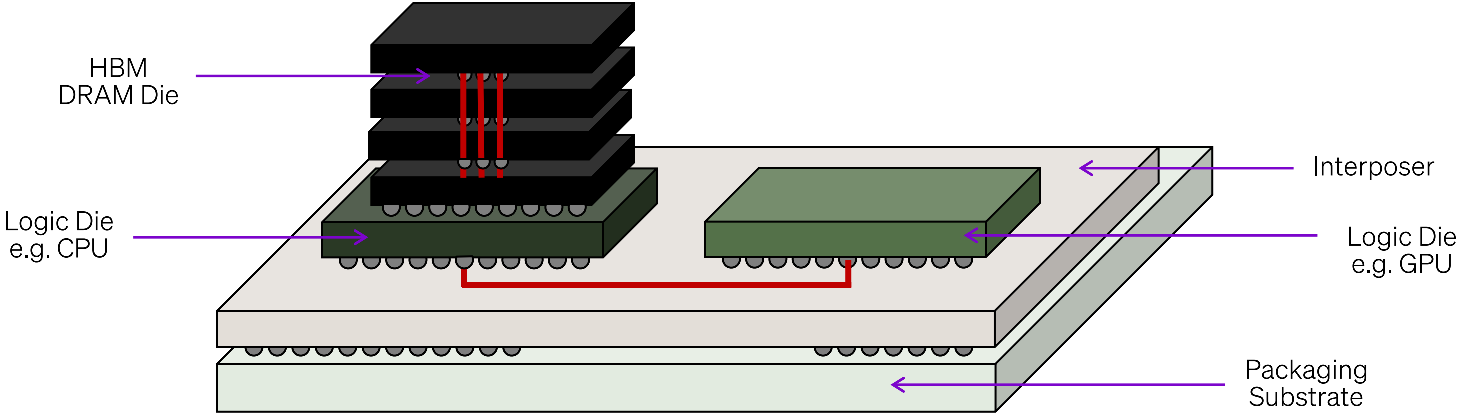


Source: Asianometry

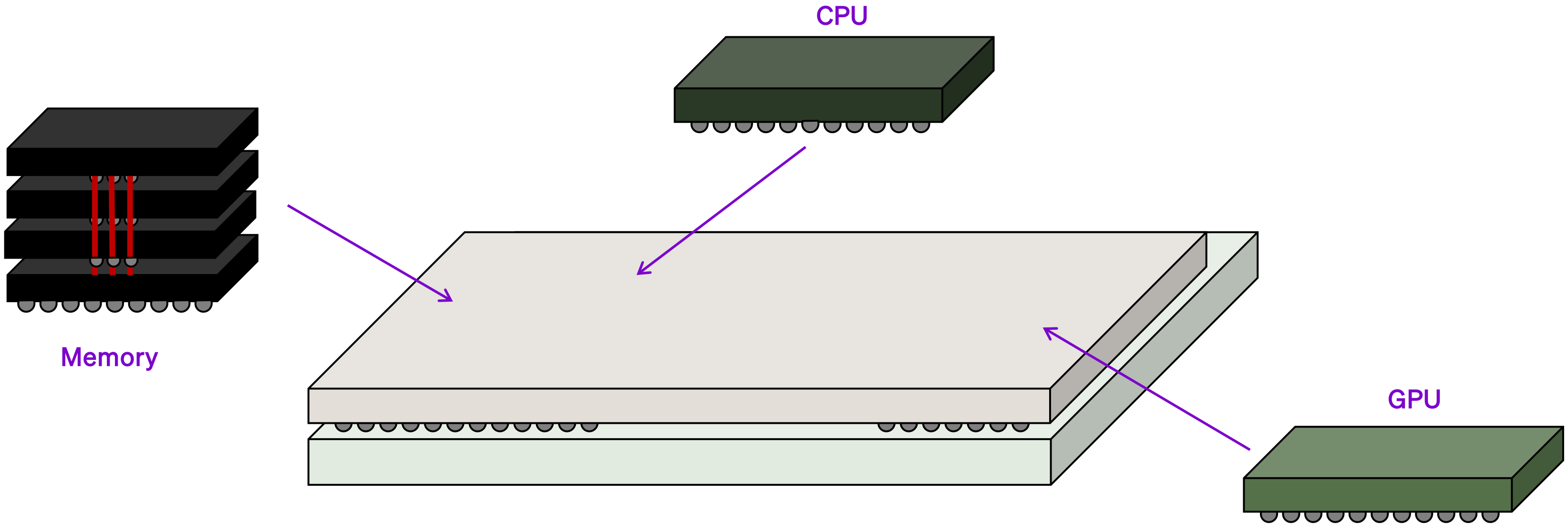
High bandwidth memory can be connected next to a processor via a silicon interposer (like a bridge) in a '2.5D' stack...



…or directly on top of the processor itself in a '3D stack' for even faster latency



This is a type of ‘system in package’, where multiple components are fabricated separately but integrated into a single chip module

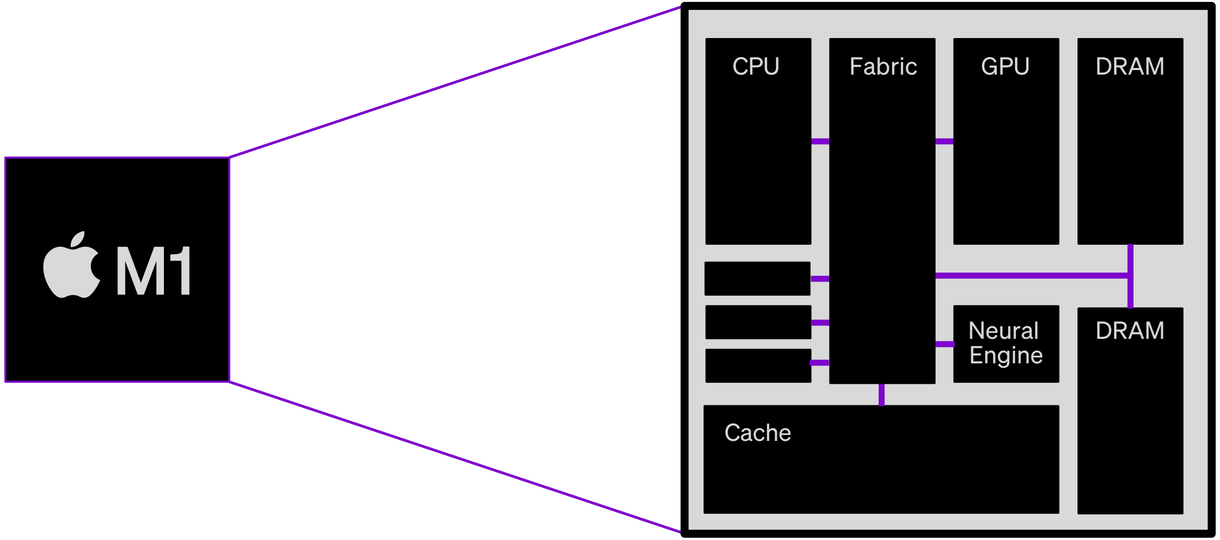


Source: Richards (2023), Understanding Semiconductors

Other methods attempt to integrate system components more closely through the design of the silicon die

Chip Design

Companies like Apple fabricate multiple components on a single die called a 'system on chip' to reduce latency, improve power efficiency and make the system more compact



Source: Apple

This is a type of application specific integrated circuit (ASIC), a type of custom chip that is designed to perform better than more general chips for specific tasks

Central Processing Unit

Primary component of a computer that executes a wide set of instructions

Field Programmable Gate Array

Reconfigurable processor that can be programmed by the user

Application Specific Integrated Circuit

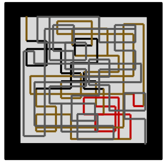
Custom-designed chip to perform a narrower range of tasks very efficiently

Graphics Processing Unit

Specialized processor for computing graphics and other parallel tasks

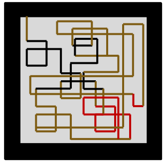
Source: Richard (2023), Understanding Semiconductors

Apple's M-series chips use the 'Arm' architecture, which is based on reduced instruction set computing instead of Intel's x86, which uses complex instruction set computing



Complex Instruction Set Computing (x86)

- More work done by the silicon vs code
- Less code space & less memory required
- Extra logic (power & heat) required to decode and execute complex instructions



Reduced Instruction Set Computing (Arm)

- More work done by code vs silicon
- More code space & more memory required
- Simpler logic leads to less power consumption and heat which is better for battery-powered devices

Source: CNBC

Companies like Apple, Amazon, Qualcomm and Nvidia license basic core designs from Arm and further customize these designs to best serve their specific needs

iPhones
and Macbooks

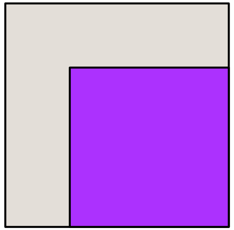
Power-efficient
server chips

CPUs for
Android devices

Deep-learning
& AI processors

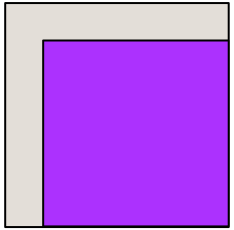
Source: CNBC

Application-specific chips can achieve better performance with a fixed number of transistors because they are optimized to make use of more of the silicon at any one time



Off-the-Shelf Chips

Lower performance and silicon utilization since many circuits are not used to run a given application



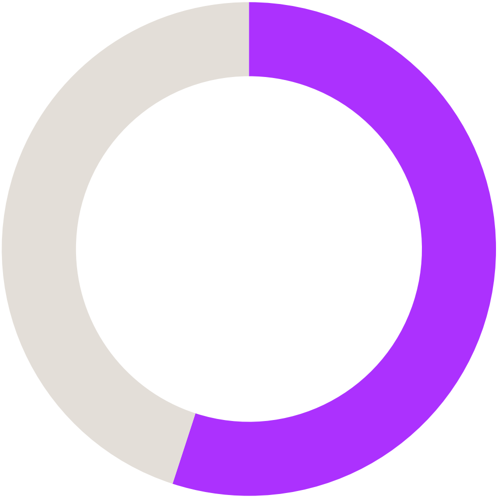
Custom Chips

Higher performance since circuits are custom designed to process a given application, leading to higher silicon utilization

Source: Richards (2023), Understanding Semiconductors

Today, more than half of Amazon's AWS server CPUs, called 'Graviton', are powered by the Arm architecture due to its superior power efficiency

% of Amazon Server CPUs by Architecture



Source: CNBC, as of November 2023

Another class of application specific chips are designed to help train and run AI models

Dive Deeper…

Further Reading & Watching

Watching:

- The World of Advanced Packaging (Applied Materials)
- Systems on a Chip (SOCs) as Fast As Possible (Techquickie)
- Why Apple's M1 Chip is So Fast (The Dev Doctor)
- Arm vs x86 - Key Differences Explained (Gary Explains)

Reading:

- Advanced Packaging (SemiAnalysis)
- Why has ARM become more popular for HPC? (Exxact)

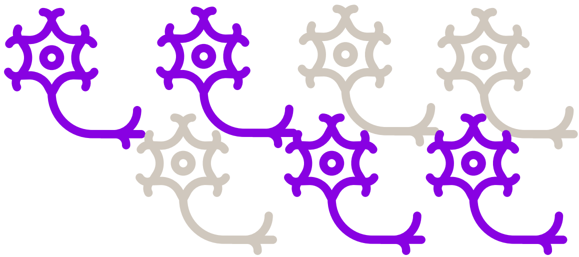
CHAPTER 08

Chips for AI

Artificial intelligence systems develop complex algorithms called neural networks which aim to replicate the human brain

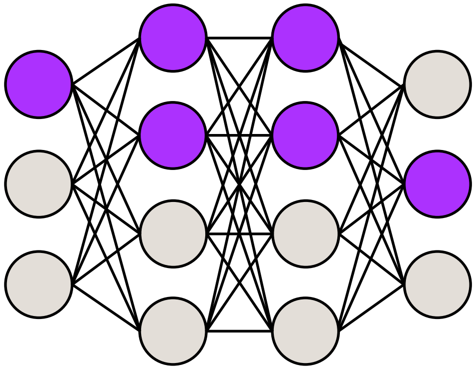
These are organized into layers of 'nodes', like neurons, which learn to process information by recognizing specific patterns in data

Neurons fire together…



=

Nodes fire together…

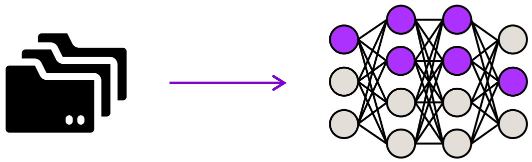


Source: Neuroscience News

Neural networks must first be trained on large amounts of data before using these learnings to ingest new data and make predictions

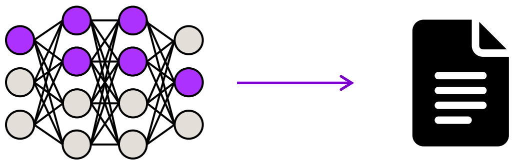
Training

Feeding large amounts of data into a neural network and adjusting how the network interprets information until it can predict the training data

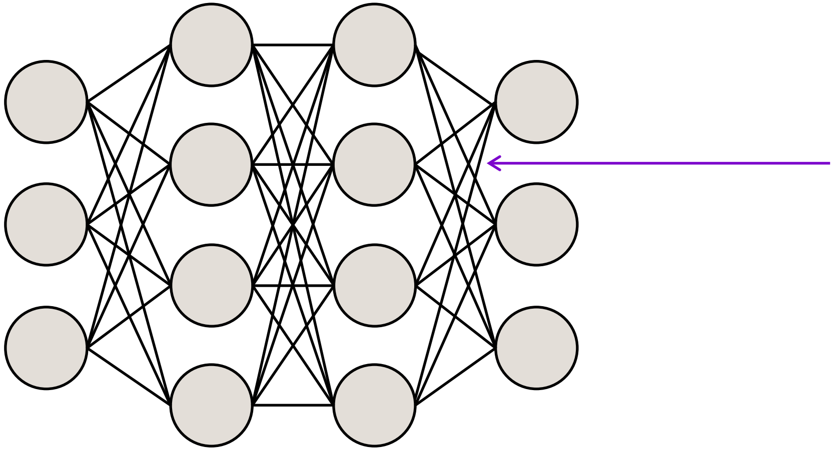


Inference

Using what the model has learned during the training process to calculate and make predictions based on new input data



Training a neural network requires large amounts of mathematical computation to adjust the ‘weights’ and ‘biases’ that the model uses to process information



$$\begin{bmatrix} 0.55 & 0.32 \\ 0.37 & 0.01 \\ 0.29 & 0.97 \\ 0.11 & 0.44 \end{bmatrix} \times \begin{bmatrix} 0.27 \\ 0.36 \\ 0.18 \\ 0.98 \end{bmatrix}$$

These computations are performed using specialized processors called GPUs, which contain thousands of cores that can perform these calculations simultaneously

Central Processing Unit

Primary component of a computer that executes a wide set of instructions

Field Programmable Gate Array

Reconfigurable processor that can be programmed by the user

Application Specific Integrated Circuit

Custom-designed chip to perform a narrower range of tasks very efficiently

Graphics Processing Unit

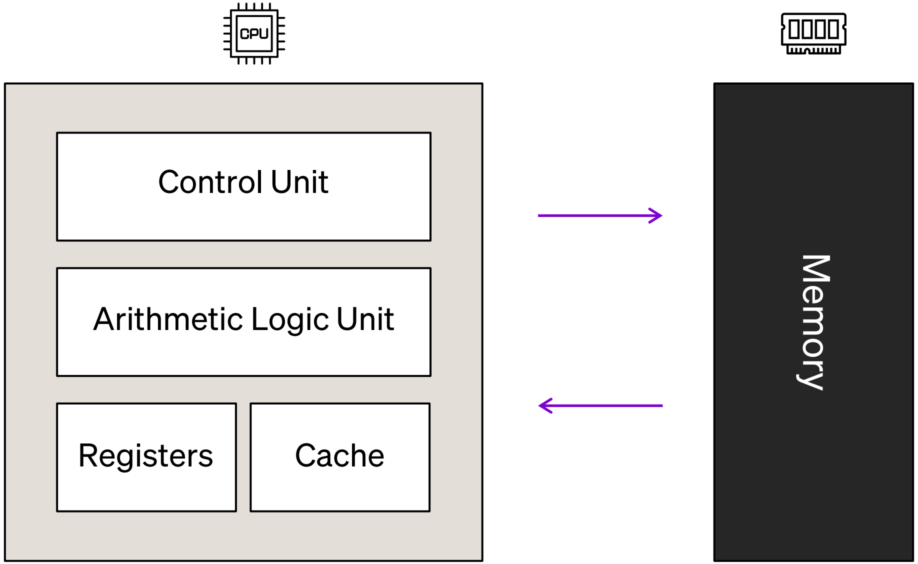
Specialized processor for computing graphics and other parallel tasks

Source: Richard (2023), Understanding Semiconductors

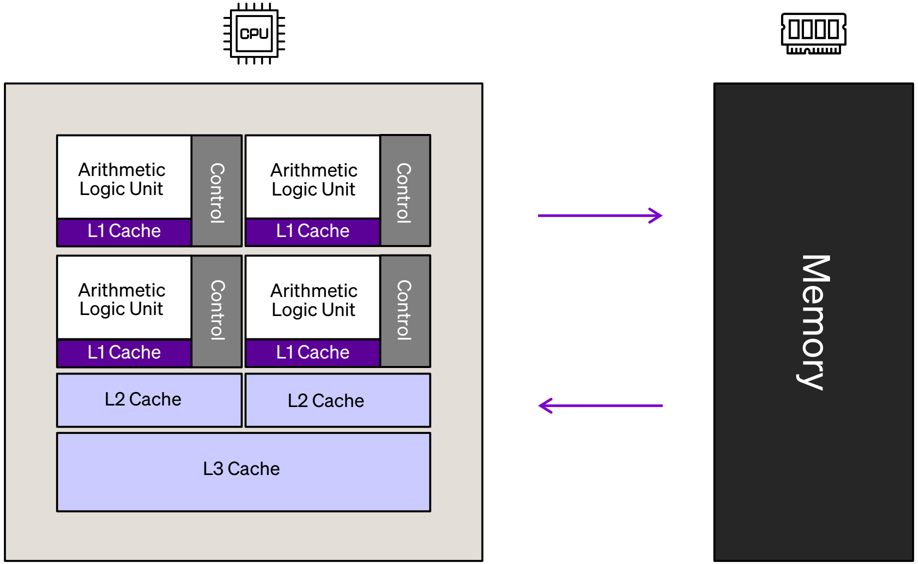
How do GPUs work?

To understand how GPUs work, we need to go back to the basics of how a CPU works

CPUs work with system memory and contain faster on-chip memory called ‘cache’ to fetch, decode and execute instructions sequentially

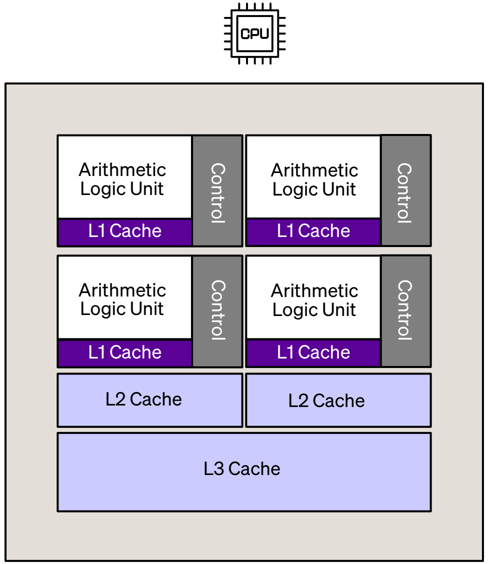


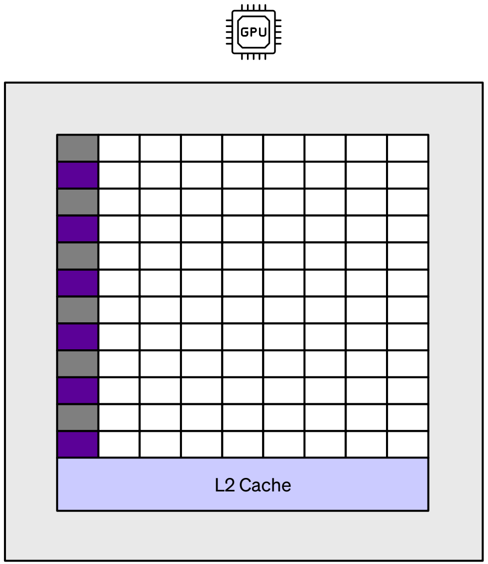
They are typically organized into several cores and dedicate much of the silicon to multiple control unit and cache circuits to reduce the latency when processing instructions



Source: Abhinav Upadhyay

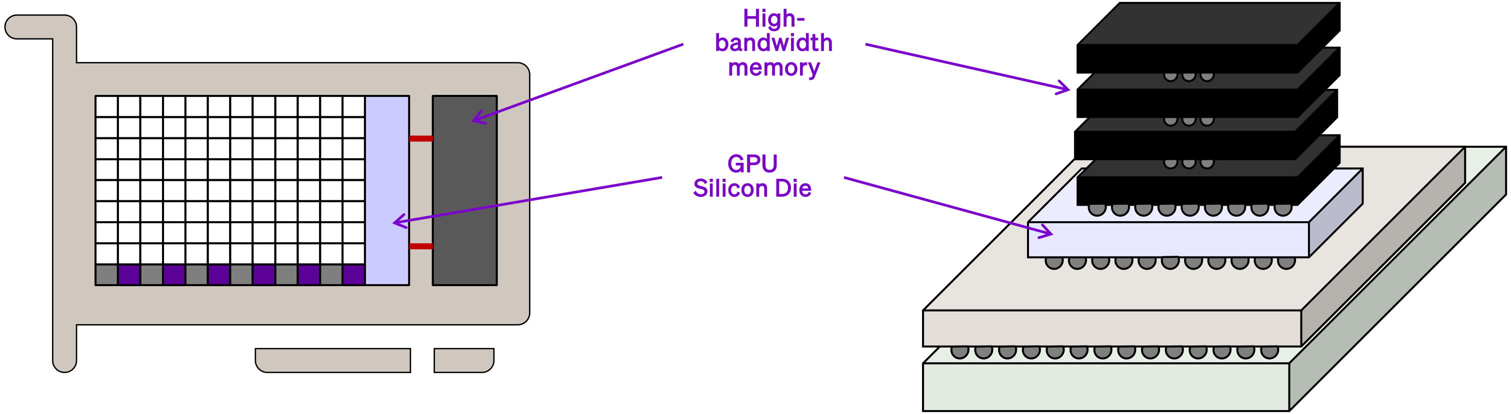
In contrast, GPUs dedicate less of the silicon towards caches and control units in favor of many ALUs which increases latency but enables massive parallel computation





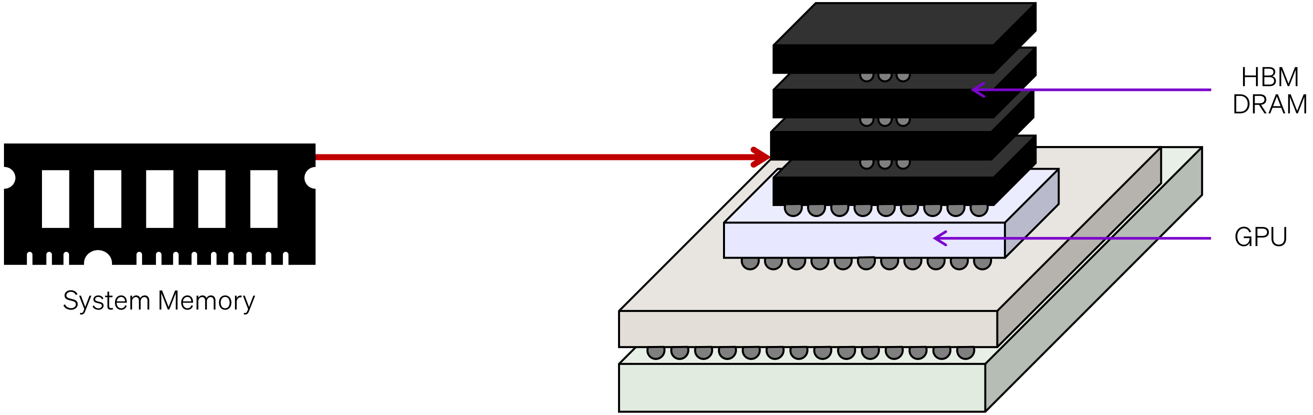
Source: Abhinav Upadhyay

A GPU card like Nvidia's H100 will typically contain both the GPU silicon die and high bandwidth memory (HBM) stacked on top of the die for low-latency AI computing



How do we train an AI model using a GPU?

When training an AI model, the training data and the model's weights and biases are loaded into the high bandwidth memory in the GPU package from the system's main memory



Then, a programmer writes a 'CUDA Kernel', a function that describes the different math functions each part of the GPU will run on the data and weights and biases in the model

```
#include <stdio.h>

// CUDA kernel to add elements of
two arrays
__global__ void vectorAdd(const
float *A, const float *B, float *C, int
numElements)
{
   int i = blockDim.x * blockIdx.x +
threadIdx.x;
   if (i < numElements) {
      C[i] = A[i] + B[i];
   }
}
```

$$\begin{bmatrix} 0.55 & 0.32 \\ 0.37 & 0.01 \\ 0.29 & 0.97 \\ 0.11 & 0.44 \end{bmatrix} \times \begin{bmatrix} 0.27 \\ 0.36 \\ 0.18 \\ 0.98 \end{bmatrix}$$

This CUDA kernel is translated by a compiler into the type of binary machine code that a GPU can execute

```
#include <stdio.h>

// CUDA kernel to add elements of
two arrays
__global__ void vectorAdd(const
float *A, const float *B, float *C, int
numElements)
{
   int i = blockDim.x * blockIdx.x +
threadIdx.x;
   if (i < numElements) {
      C[i] = A[i] + B[i];
   }
}
```

Compiler →

Machine Code

```
00010000011110001100110000100
00111100011001100001000011110
00110011000010000111100011001
10000100001

111100011001100001000011110001
10011000010000111100011001100

01000011110001100110000100001
111000110011000010000111100
01100110000100001111000110011

00001000001111000110011000010
000111100011
```

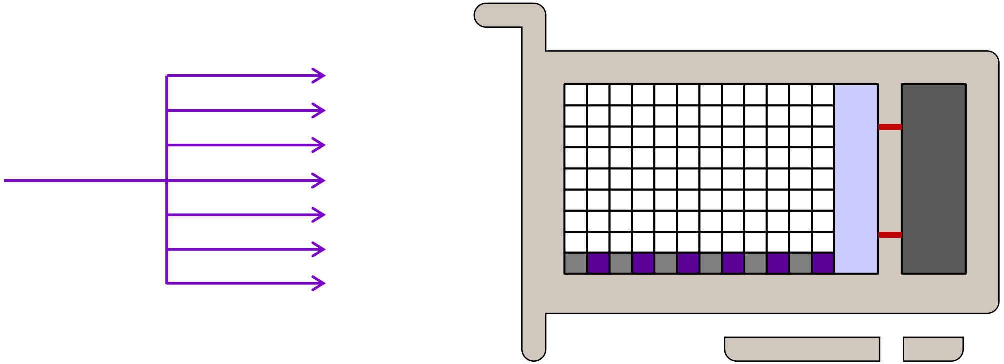
Once the kernel is executed, each of the GPU's cores is used to calculate and update the weights and biases across an AI model

000100000111100011001100000100
001111000110011000001000011110
0011001100000100000111100011001
10000100001

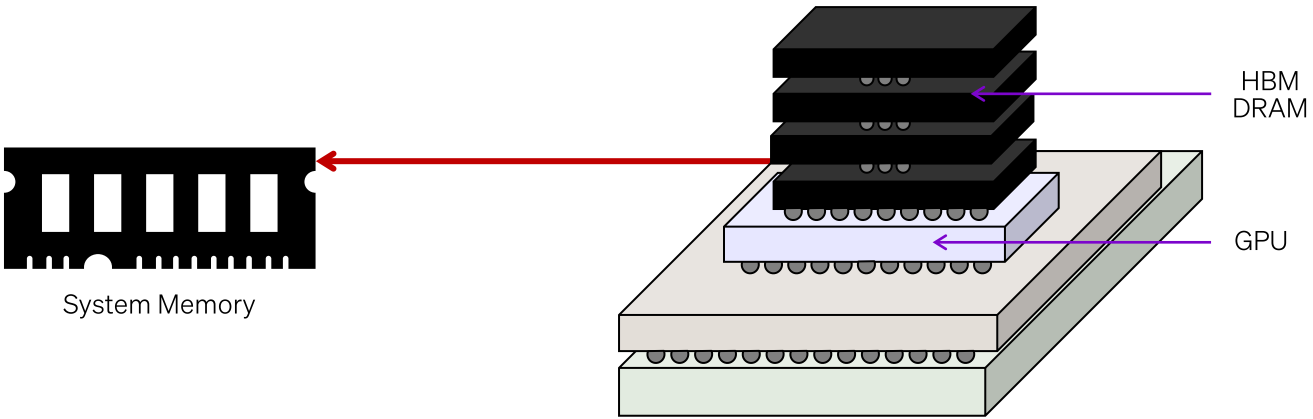
11110001100110000010000011110001
1001100000100000111100011001100

0100000111100011001100000100001
11100011001100000100000111100
0110011000001000001111000110011

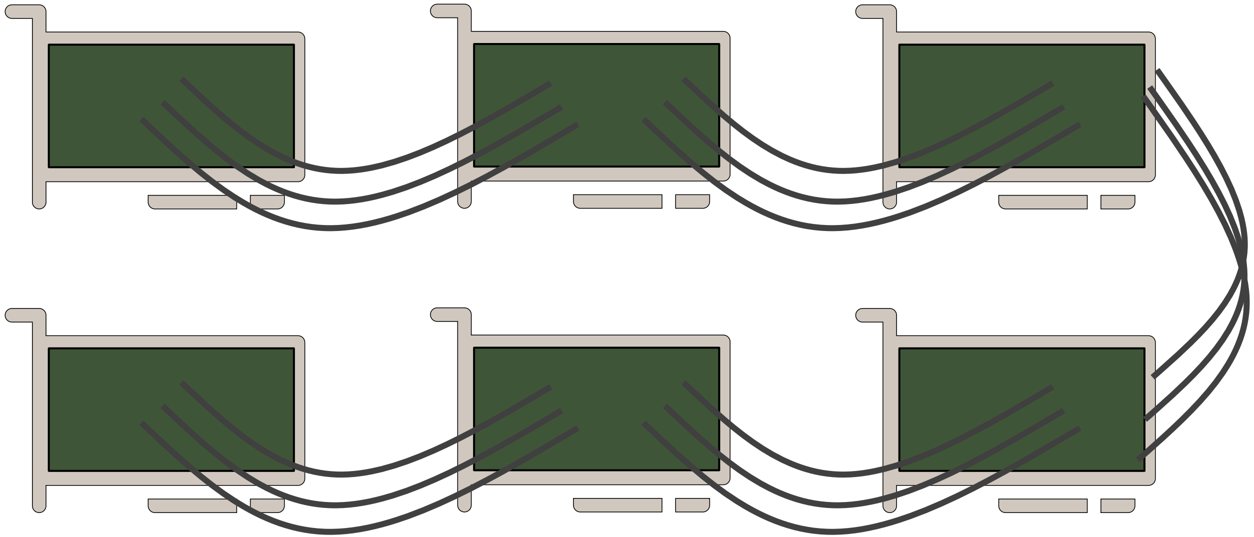
000010000011110001100110000010
000111100011



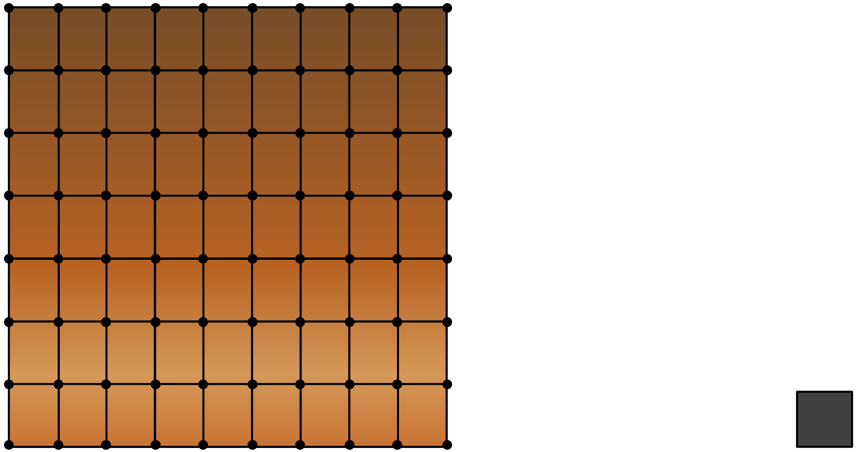
Once these calculations are complete, the final weights and biases are written back to the system's main memory for further use



Training an AI model requires a cluster of multiple GPUs connected together with wires, which introduces latency as data and instructions are transmitted between GPUs...



…so companies like Cerebras are building processors the size of wafers which contain orders of magnitude more memory and transistors to train AI models on a single chip



Cerebras Wafer Scale Engine-3

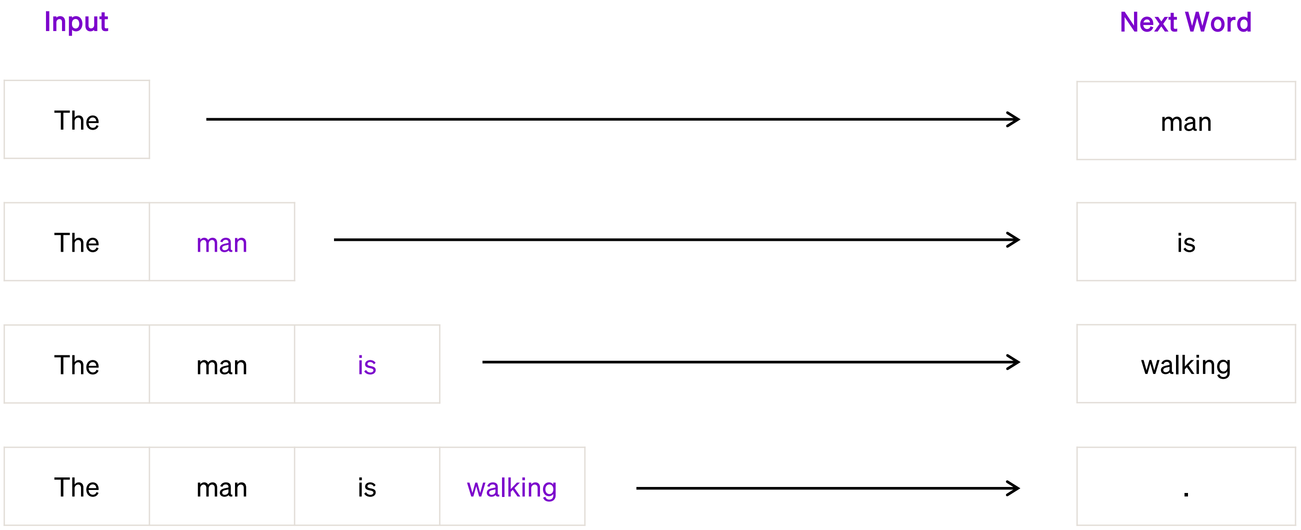
Size: 46,225 mm^2
Cores: 900,000

Nvidia H100

Size: 814 mm^2
Cores: 17,424

While GPUs excel at training due to their parallel compute architecture, they are slower at tasks like inference which require lots of sequential computations

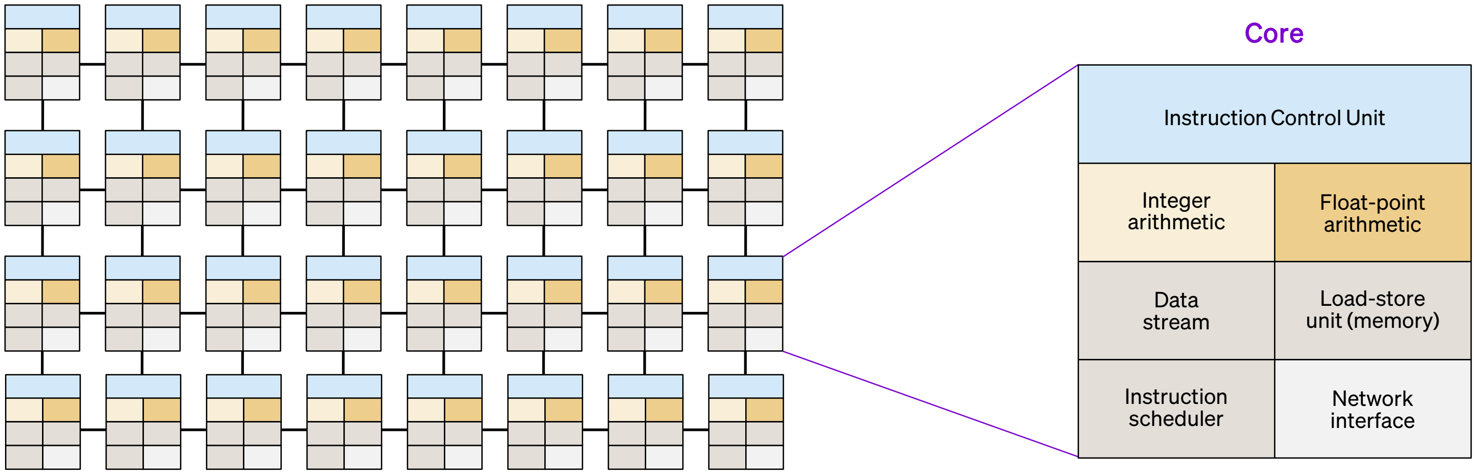
This is particularly true of language models, which work in an autoregressive manner by using the previous words in a sequence to predict the next word



Companies like Groq have built specialized processors for inference like the ‘language processing unit’ (LPU) which processes information using a radically different chip architecture

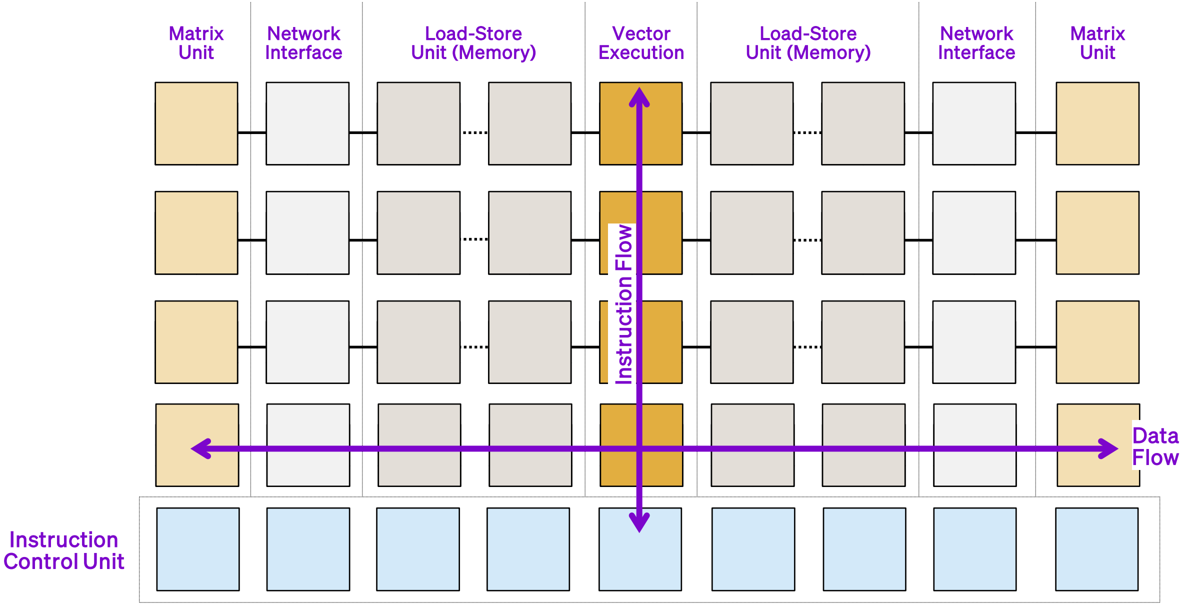
How does the LPU work?

Traditional processors consist of multiple cores, and each core consists of multiple circuits that perform different kinds of computations



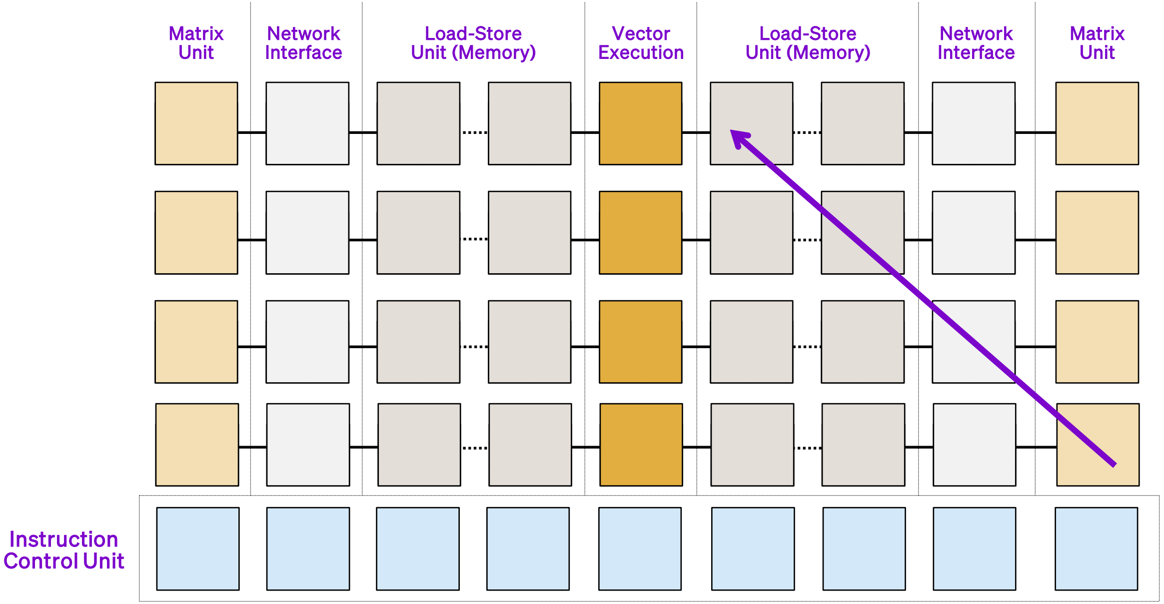
Source: Abhinav Upadhyay, Groq

The language processing unit reversed this architecture by arranging circuits for each type of computation into their own columns called ‘slices’



Source: Abhinav Upadhyay, Groq

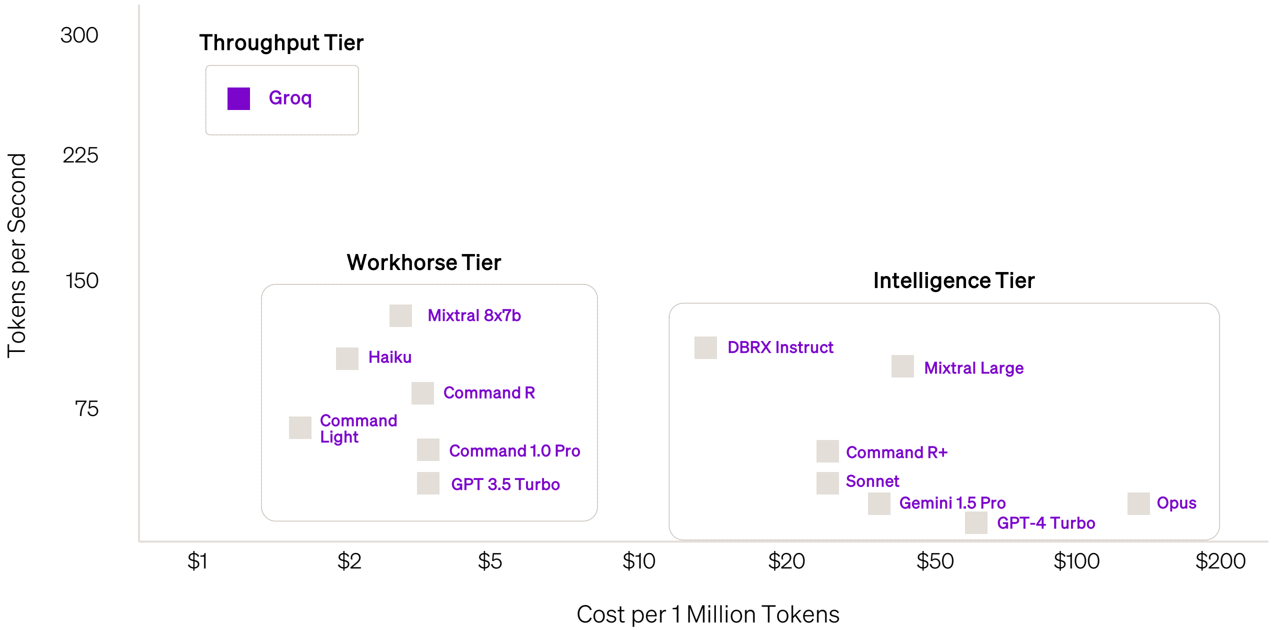
Information streams through the LPU slice to slice in a deterministic fashion scheduled by the compiler, enabling a parallel computing architecture that is suitable for inference



Source: Abhinav Upadhyay, Groq

This results in faster inference at a lower cost per token than models powered by other chip architectures

LLM Cost vs. Performance



Source: Databricks Mosaic Research

Dive Deeper…

Further Reading & Watching

Watching:

- GPUs: Explained (IBM)
- Nvidia CUDA in 100 Seconds (Fireship)
- Conversation with Groq CEO Jonathan Ross (Social Capital)
- Is it the Fastest AI Chip in the World? (Anastasi in Tech)

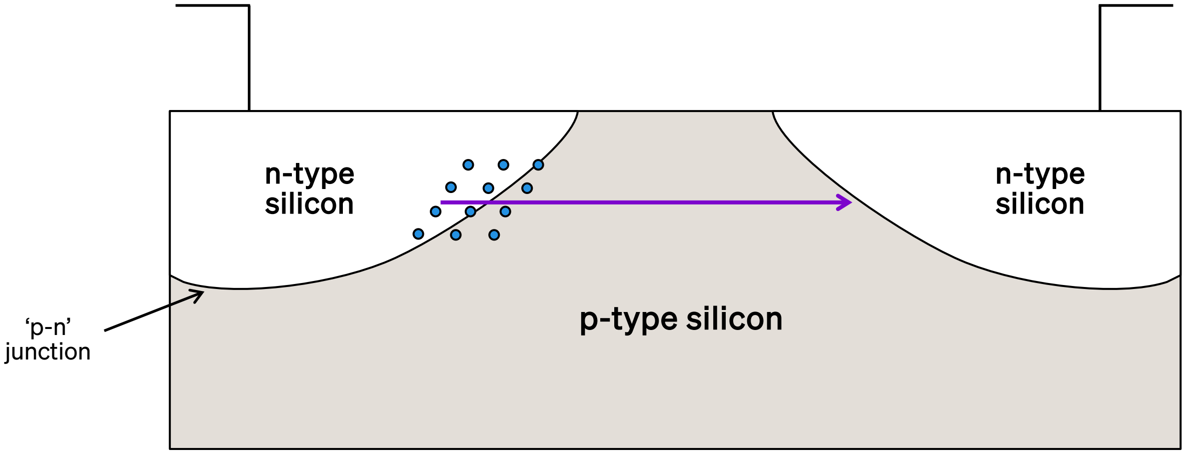
Reading:

- What Every Developer Should Know About GPU Computing (Abhinav Upadhyay)
- Cerebras CS-3: The World's Fastest and Most Scalable AI Accelerator (Cerebras)
- A Deep Dive into the Underlying Architecture of Groq's LPU (Abhinav Upadhyay)

CHAPTER 09

Quantum Computing

As transistors approach the size of atoms, we begin to see effects like 'quantum tunnelling' where electrons overcome the p-n junction without having enough energy to do so



Source: Veritasium

Quantum computers aim to harness these quantum phenomena to store and process information in a different way from traditional transistors

How do quantum computers work?

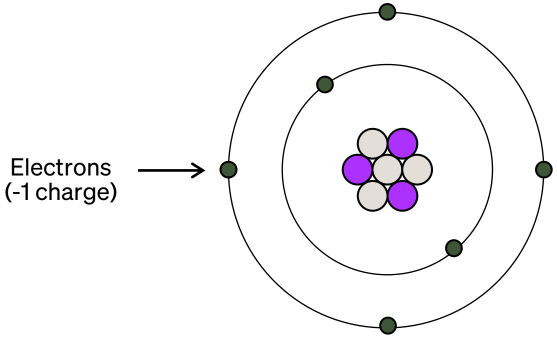
Traditional computers process information using binary ‘bits’ of 0 and 1, which reflect the off and on states of transistors

Transistor is off because no voltage is applied

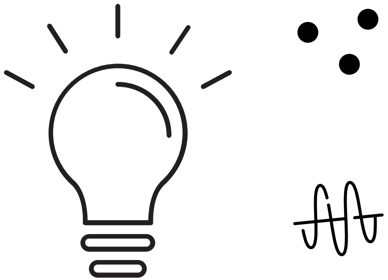
Transistor is on after a voltage is applied

Quantum computers store information using 'qubits', which are based on the manipulation and measurement of quantum particles like electrons and photons

Electrons

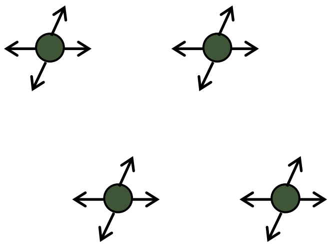


Photons

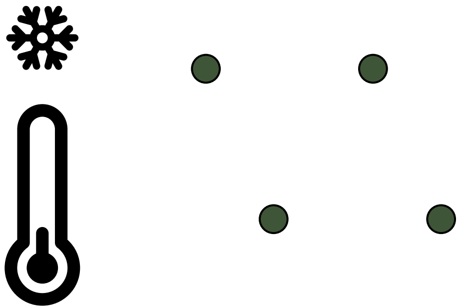


Quantum particles move around a lot at higher temperatures, so to manipulate and measure them effectively, they are cooled to near absolute-zero to restrict movement

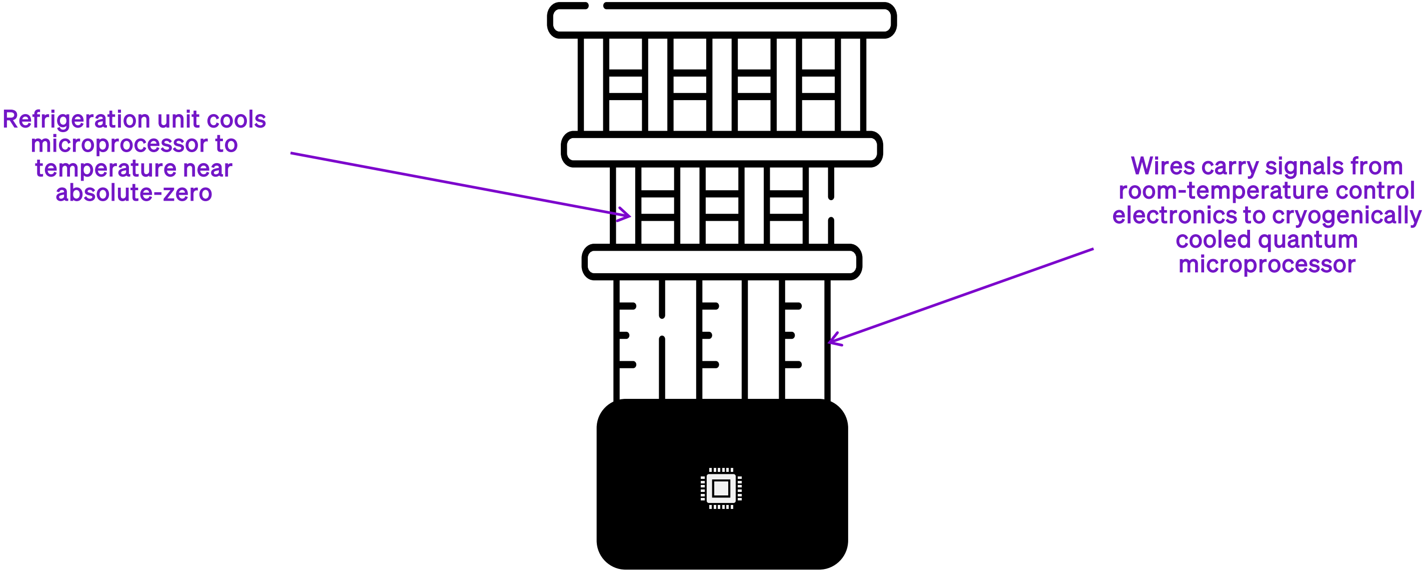
Electrons move around as thermal energy is converted to kinetic energy



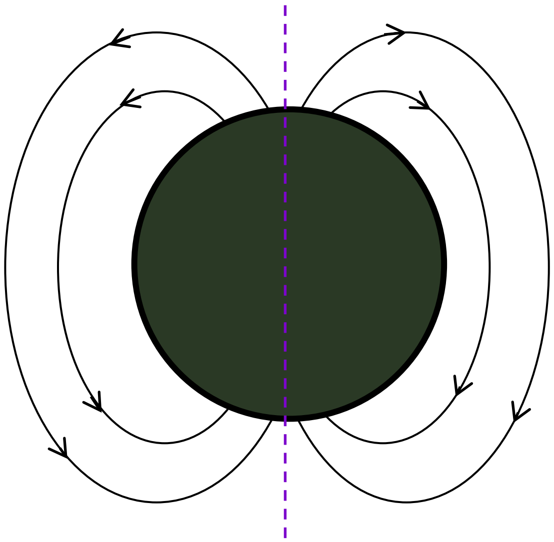
Movement is restricted when temperatures are reduced to near absolute-zero



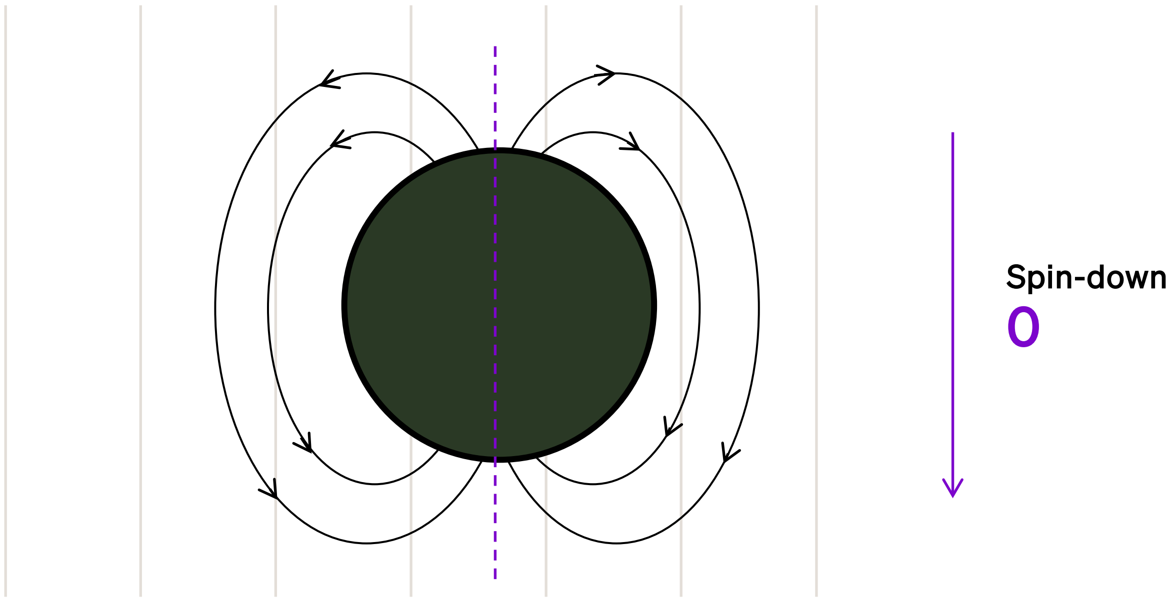
This is why quantum computers contain a multi-layered refrigeration unit to cool the processor to this temperature



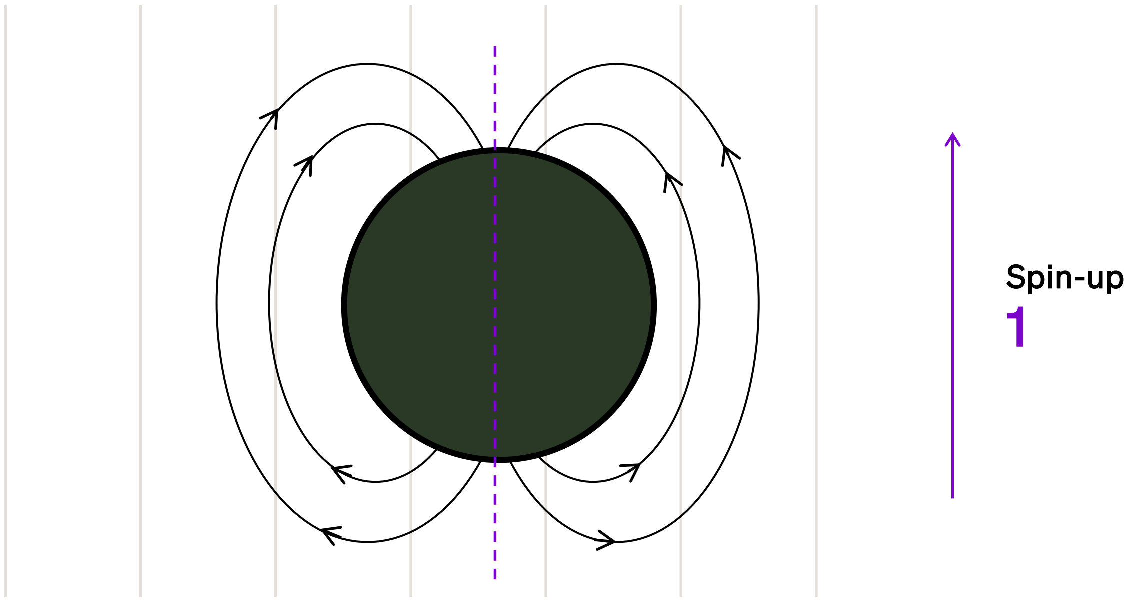
Quantum particles, like electrons, spin on their axis and have their own magnetic field like the earth



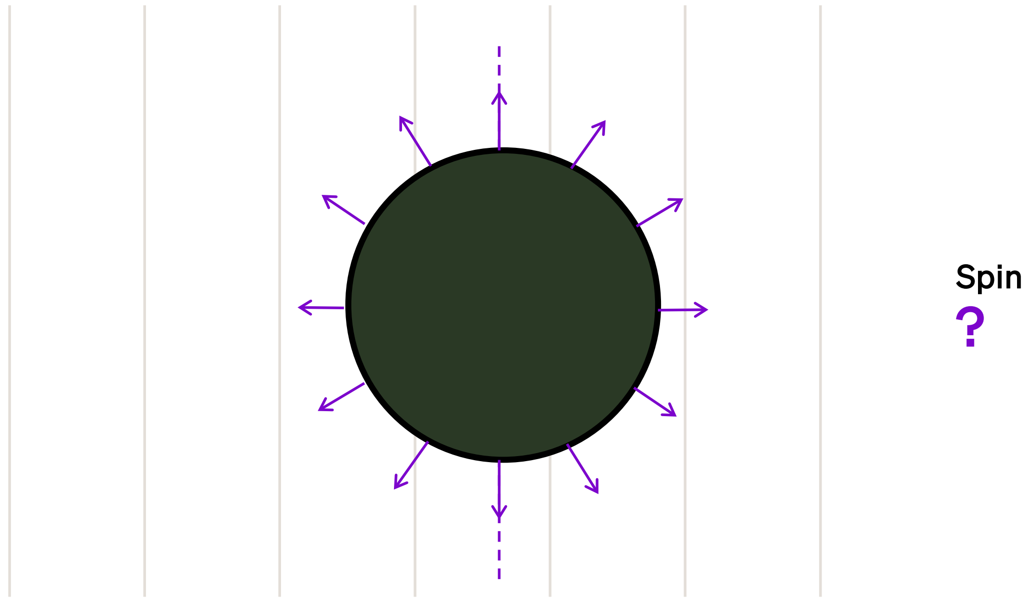
When an electron is suspended in a magnetic field, its poles will align with the direction of the magnetic field, representing a 0 or low-energy state called 'spin down'



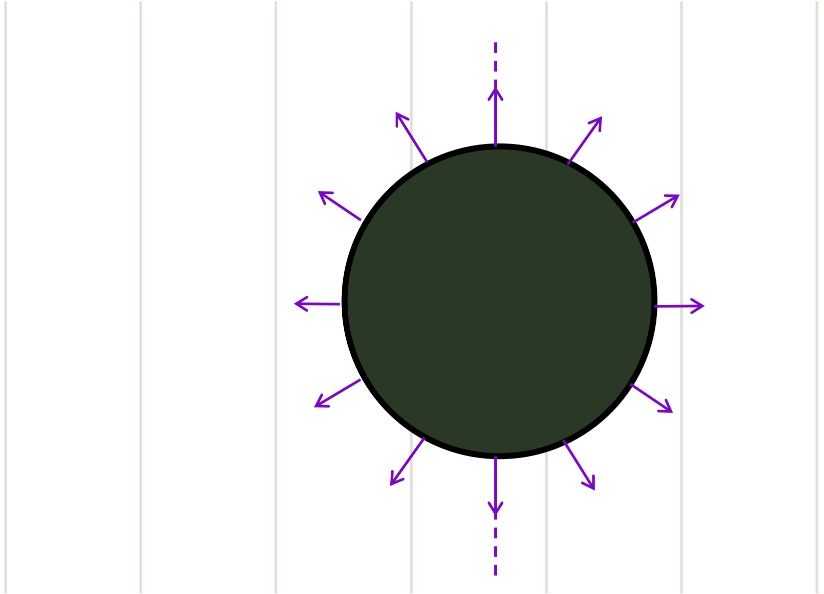
The electron can also be charged with energy to point in the opposite direction to the magnetic field which represents a ‘1’, called ‘spin-up’



Due to a quantum phenomenon called ‘superposition’, this electron is actually spinning in all directions at once, and we only know whether it is up or down at the point of measurement



This means that before being measured, each electron has an associated probability of being spin up or spin down



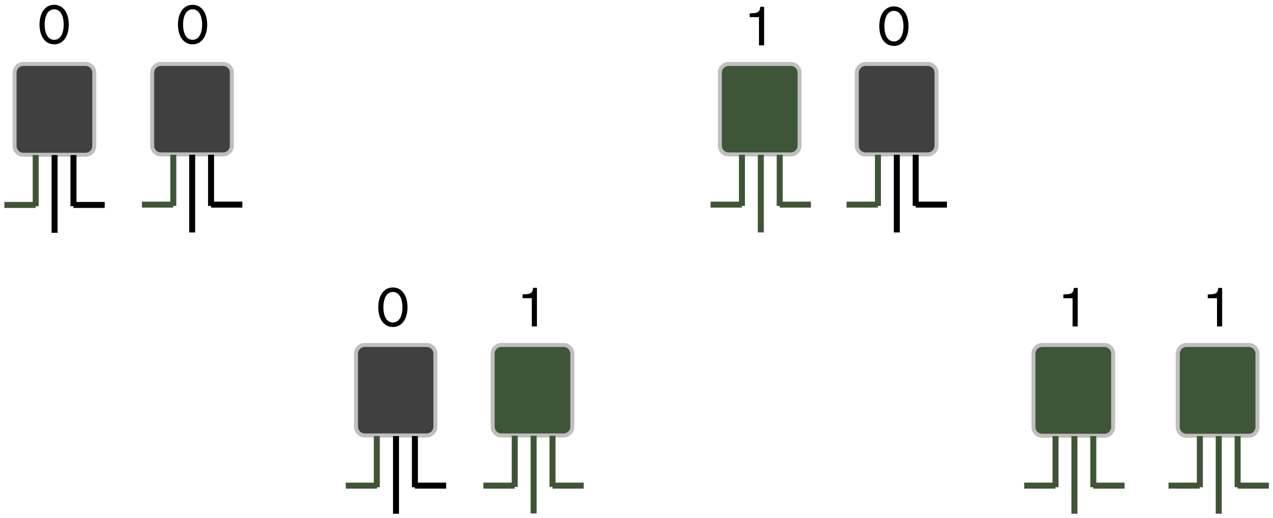
Spin

0.6 = 60% chance of being ‘spin-up

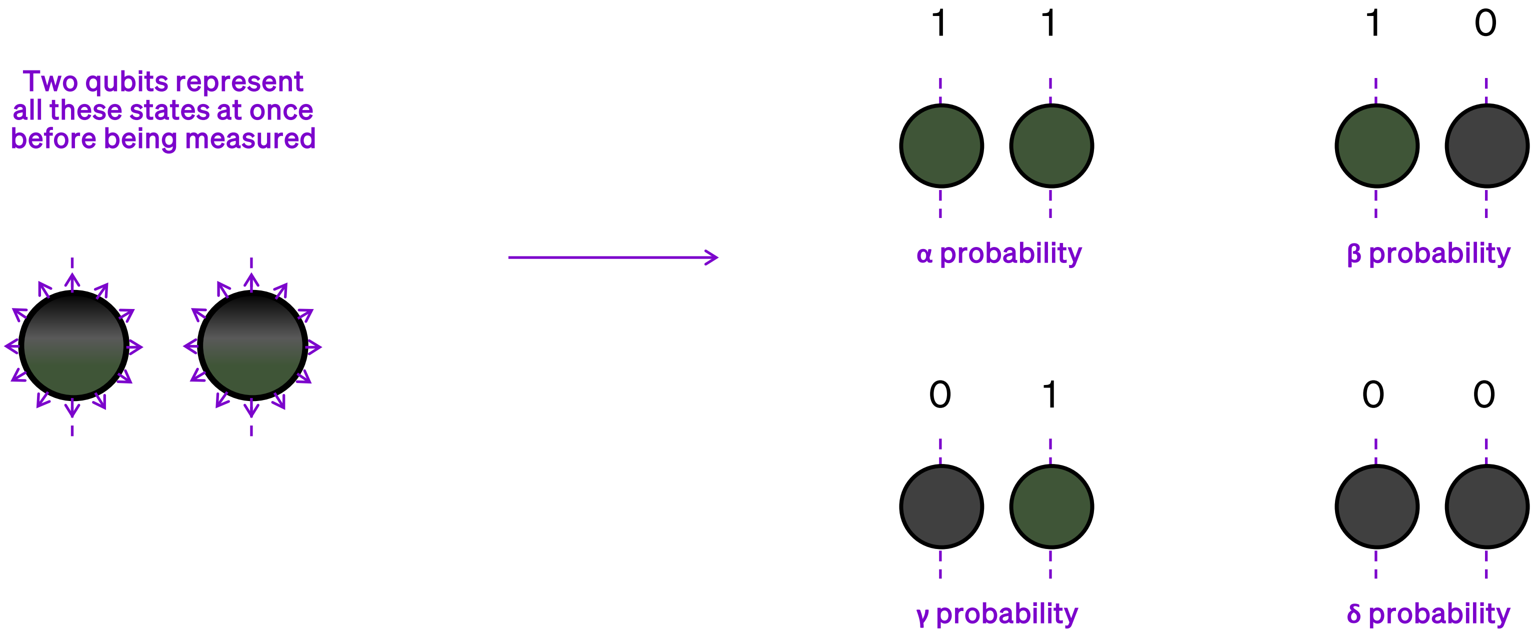
SOCIAL CAPITAL_

How do quantum computers use this to represent and store information?

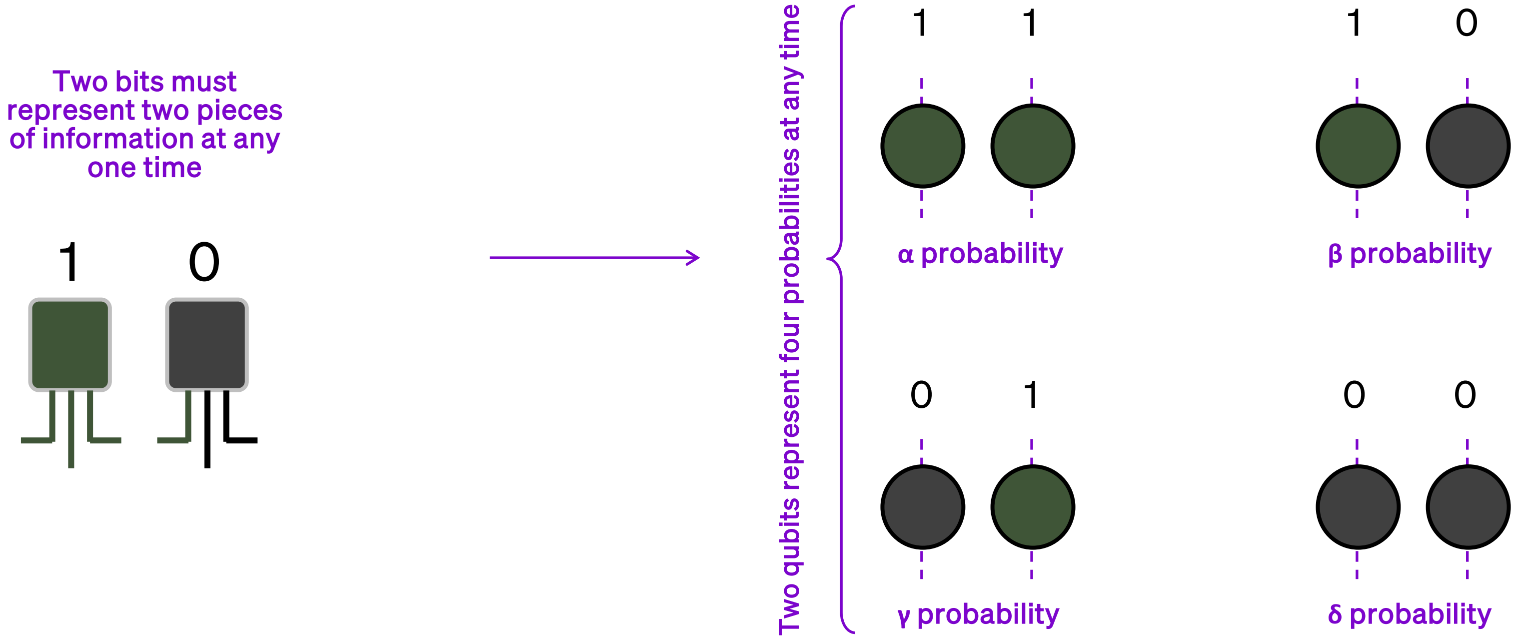
With traditional computing, using two bits allow us to create a system with four possible states that can represent and process information



In quantum computing, 'qubits' represent all these possible states at once, so at any one time, there is a given probability that two qubits represent each of these four states

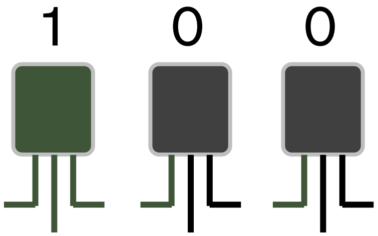


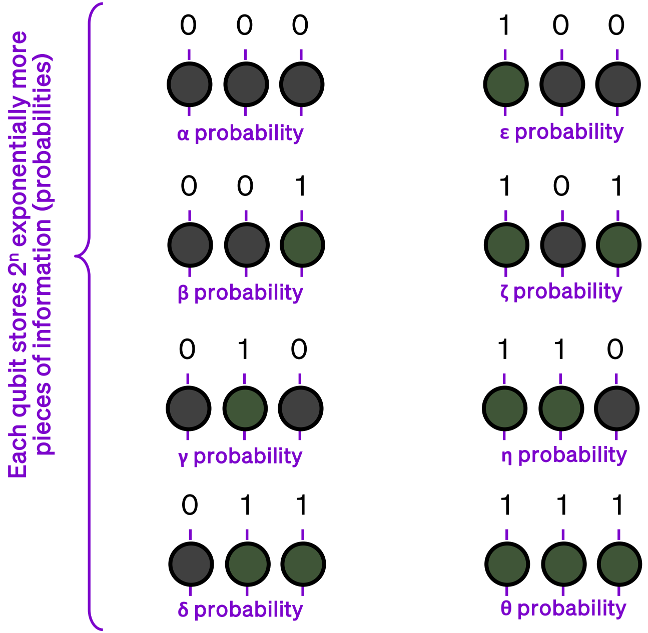
This allows two qubits to represent four pieces of information (probabilities) at any time, unlike two traditional bits, which must represent two pieces of information at a time



Each new traditional bit results in a linear increase in compute, while each qubit results in an exponential increase in compute

Each additional bit stores 1 additional piece of information





Quantum computers use these probabilities
to solve large, multi-variate problems
much faster than traditional computers

This could pose a threat to existing encryption methods, which encrypt data using large numbers with prime factors that cannot be factored quickly using traditional computers

A traditional computer will take many years to factor a large number into its two primes

10,967,535,067

104,729 104,723

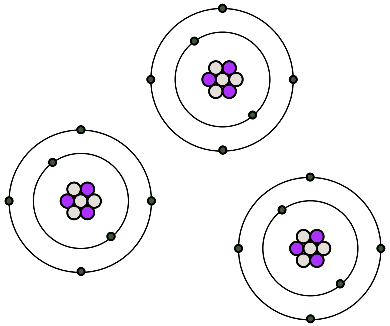
But a quantum computer can do this orders of magnitude faster, threatening current encryption methods

10,967,535,067

104,729 104,723

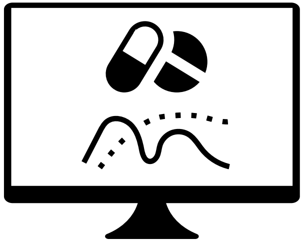
But quantum computers are very useful for modeling reactions at a particle level in nature, which could unlock new insights into material science and drug development

Quantum computers will be better at modeling chemical interactions in nature…



→

…which can unlock new insights into material science and drug development



Dive Deeper…

Further Reading & Watching

Watching:

- How Does a Quantum Computer Work? (Veritasium)
- How Quantum Computers Break The Internet… Starting Now (Veritasium)
- Quantum Computers Explained in a Way Anyone Can Understand (TheUnlockr)
- Quantum Computers, Explained with MKBHD (Cleo Abram)

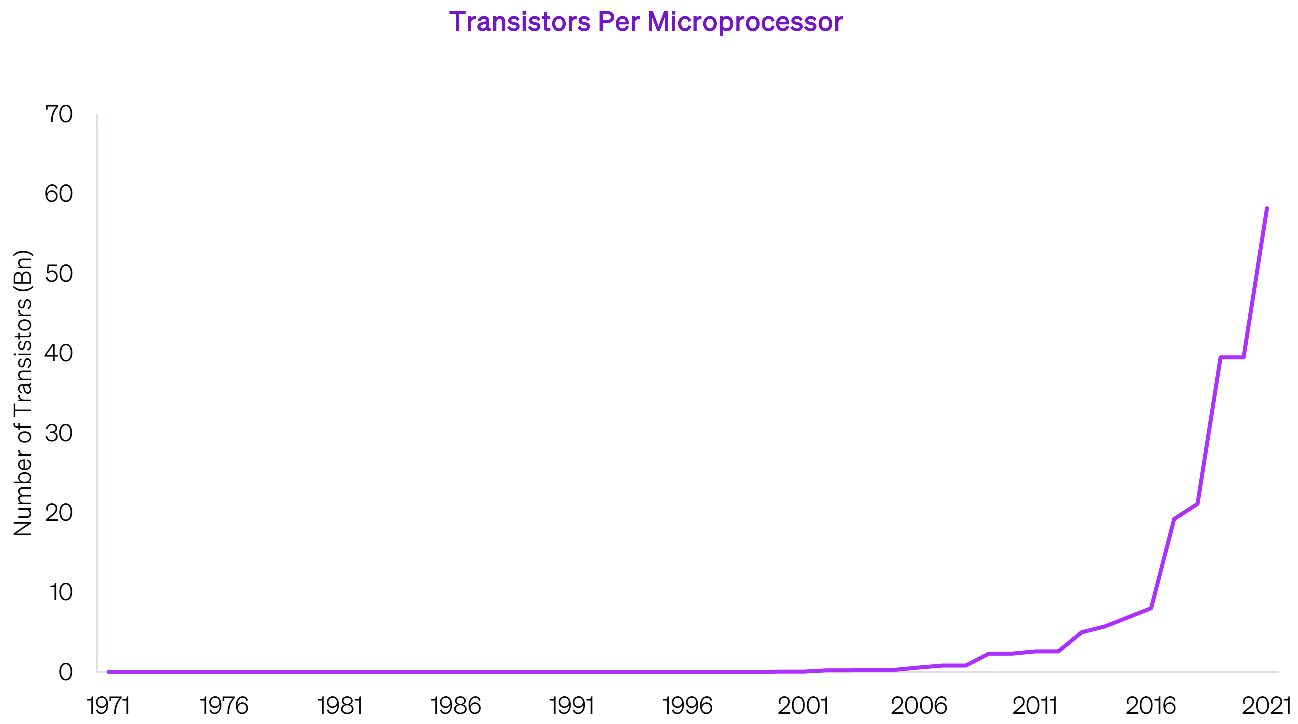
Reading:

- What is Quantum Computing? (IBM)
- Quantum Computing: What Leaders Need to Know Now (MIT)

CHAPTER 10

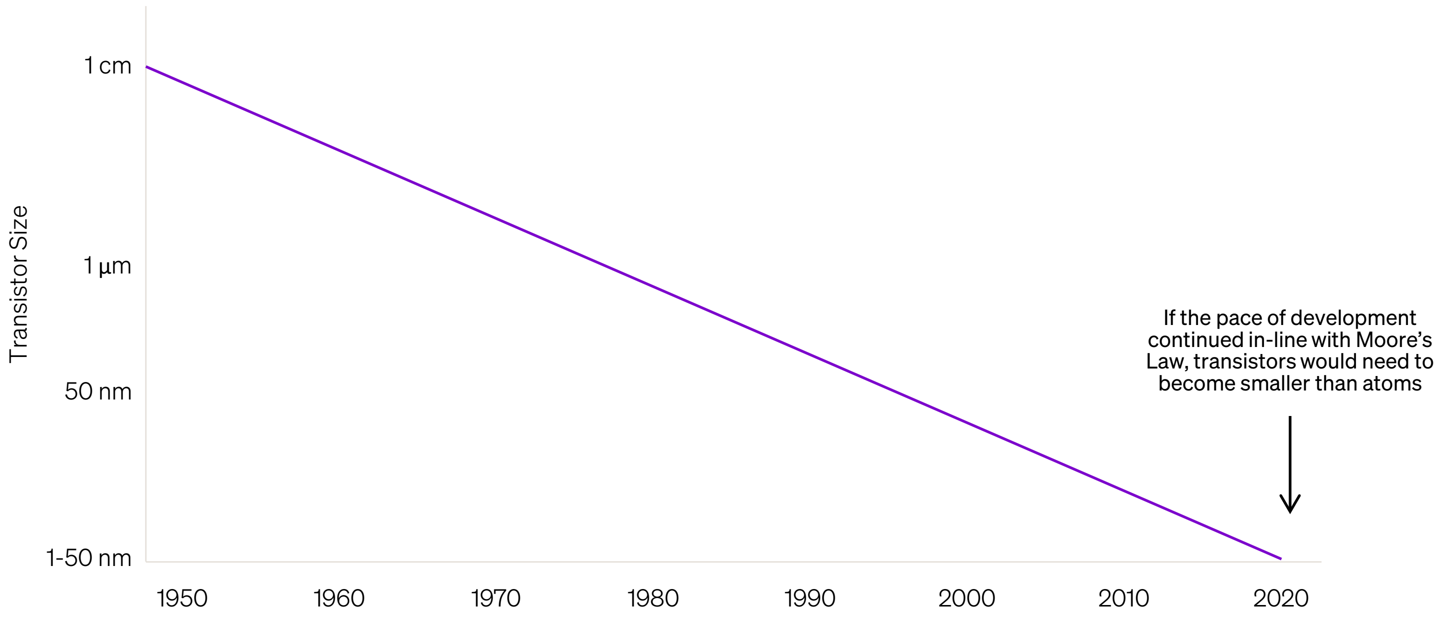
Wrapping Up

Since the early 1960s, the number of transistors that can fit on a single chip has roughly doubled every two years, in-line with a relationship called ‘Moore’s Law’



But as transistors approach the size of atoms, physical constraints are reducing the rate at which transistor count continues to scale

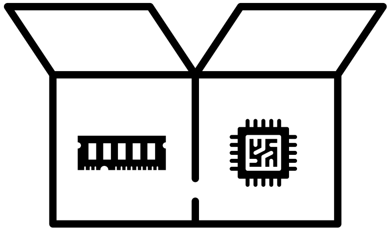
Transistor Size Trend Line According to Moore's Law



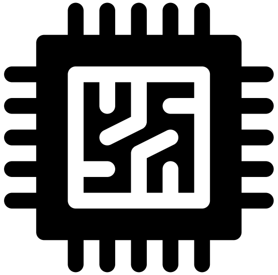
Source: Pokropivny et al. (2007), Introduction to Nanomaterials and Nanotechnology

To continue to increase performance, companies have therefore turned to packaging chips more closely with other components, and designing new types of custom chips

Chips can be packaged closely with components like memory to improve performance

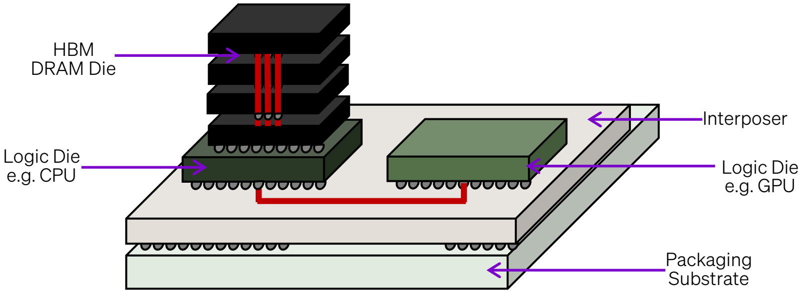


New types of custom chips can be optimized for specific tasks

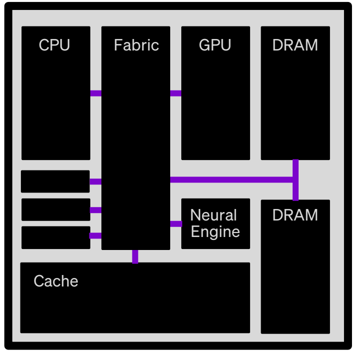


Techniques like 3D packaging and 'system on chips' can improve performance by reducing the latency of signals between components and increasing processor utilization…

3D Integration places two sets of silicon die on top of each other

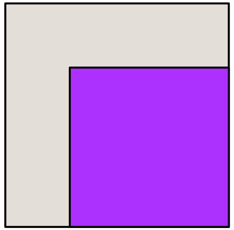


'System-on-chips' fabricate multiple components on a single die



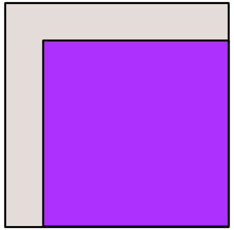
Source: Navid Asadi

…and custom silicon designs can achieve better performance with a fixed number of transistors because they are optimized to make use of more of the silicon at any one time



Off-the-Shelf Chips

Lower performance and silicon utilization since many circuits are not used to run a given application



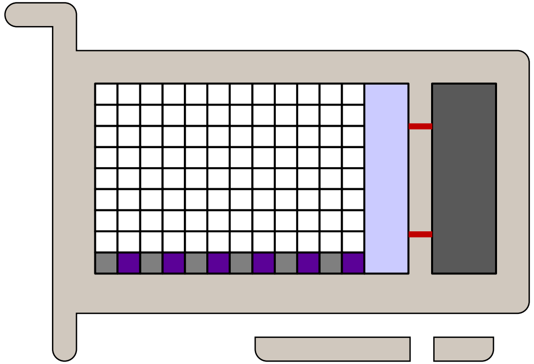
Custom Chips

Higher performance since circuits are custom designed to process a given application, leading to higher silicon utilization

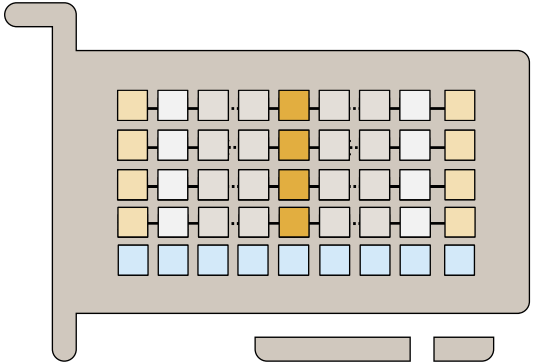
Source: Richards (2023), Understanding Semiconductors

As AI workloads increase, new types of silicon like GPUs and LPUs are being designed to train and run models

Graphics Processing Unit (Training)



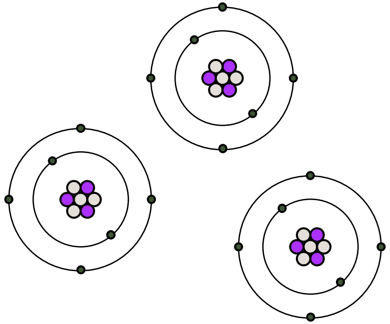
Language Processing Unit (Inference)



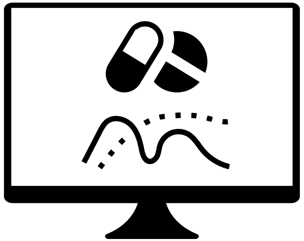
Source: Groq

And new types of computers, like quantum computers, attempt to solve complex problems using a radically different method of computing

Quantum computers leverage the properties of quantum particles like electrons…



…which can unlock new insights into material science and drug development



Source: Groq

Future Deep Dives…

Month	Theme	Deep-Dive	Summary
Dec	Energy Transition	The Global Energy Transition	What is climate change and why is it happening? Where are global carbon emissions coming from? What are the key pieces of legislation we have implemented to solve this?
Jan	Deep Tech	A Primer on Artificial Intelligence	What is Artificial Intelligence and what are the different types? How do the various models work? How is value created? What are the risks?
Feb	Life Sciences	The Business Model of Healthcare	What are the incentives that drive the behavior and outcomes of drug companies, insurers and hospitals? What new disruptions are at hand?
Mar	Deep Tech	The Future of Space	What are the legacy and emerging business models built around space? How do we access space today? What will space look like tomorrow?
Apr	Deep Tech	Moore's Law and Next Steps for Silicon	What is Moore's Law and has it broken down? What are the different types of semiconductors? Why are companies moving towards custom-designed silicon?
May	Economic Analysis	Creator Economy: The Next Phase of Media	How do consumers make decisions today? How are influencers becoming tastemakers? What legacy businesses are being disrupted?
Jun	Deep Tech	Defense 2.0: Protecting America	How much does the U.S. spend on defense? What weapons and systems do defense companies produce today? What will the future of defense look like tomorrow?
Jul	Life Sciences	A Primer on Biotech	What are the different types of drugs and therapies? How do the economics of drug companies work? Why have biotech sector returns been so poor over the past decade?
Aug	Socio-Political Trends	Is India the Next Economic Giant?	Where is India's economy today and where might it be tomorrow? What are the key demographic and social factors that are driving the country's development?
Sep	Economic Analysis	'Go Woke, Go Broke?'	Which companies have 'gone woke' and why? Where has this business strategy succeeded and failed? Do companies that 'go woke' underperform their peers?
Oct	Economic Analysis	When Companies Go 'Ex-Growth'	What does it mean for a company to go 'ex-growth'? Why does it happen? What are the implications for valuation? How can companies respond?
Nov	Socio-Political Trends	A Demographic and Social Breakdown of America	Where is America today? A visual representation of our democracy, demography, economy, quality of life, progress and more.

Disclaimer

This document is provided for educational purposes only. Nothing contained in this document is investment advice, a recommendation or an offer to sell, or a solicitation of an offer to buy, any securities or investment products. References herein to specific sectors are not to be considered a recommendation or solicitation for any such sector. Additionally, the contents herein are not to be construed as legal, business, or tax advice.

Statements in this document are made as of the date of this document unless stated otherwise, and there is no implication that the information contained herein is correct as of any other time. Certain information contained or linked to in this document has been obtained from sources believed to be reliable and current, but accuracy cannot be guaranteed.

This document contains statements that are not purely historical in nature but are “forward-looking statements” or statements of opinion or intention. Any projections included herein are also forward-looking statements. Forward-looking statements involve known and unknown risks, uncertainties (including those related to general economic conditions), assumptions and other factors, which may cause actual results, performance or achievements to be materially different from those expressed or implied by such forward-looking statements. Accordingly, all forward-looking statements should be evaluated with an understanding of their inherent uncertainty and recipients should not rely on such forward-looking statements. There is no obligation to update or revise these forward-looking statements for any reason.

This document also contains references to trademarks, service marks, trade names and copyrights of other companies, which are the property of their respective owners. Solely for convenience, trademarks and trade names referred to in this document may appear without the ® or ™ symbols, but such references are not intended to indicate, in any way, that such owner will not assert, to the fullest extent under applicable law, its rights or the right of the applicable licensor to these trademarks and trade names.

Affiliates of Social Capital are investors in Groq.