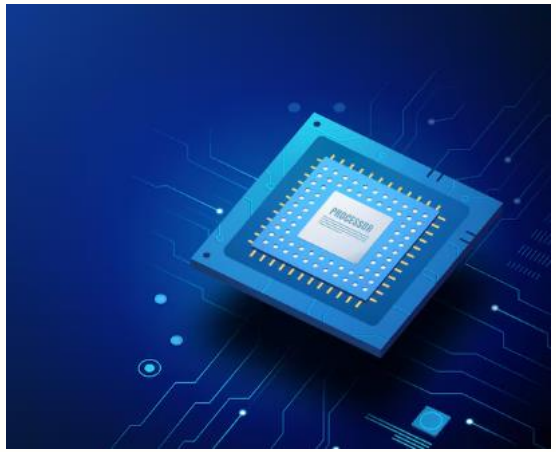
Computer in our pocket – How did we get there?

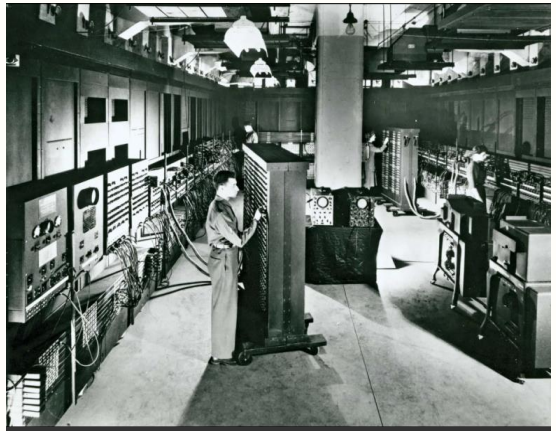
Smart phone Processor

- 15~25 billion transistors
- Tinier than palm of your hand
- Consumes a few milli watts to watts



80 years ago, a computer filled an entire room.

- Could perform up to 5,000 additions per second, much faster than its electromechanical predecessors.
- 18,000 vacuum tubes, 70,000 resistors, 10,000 capacitors, 6,000 switches, and 1,500 relays.

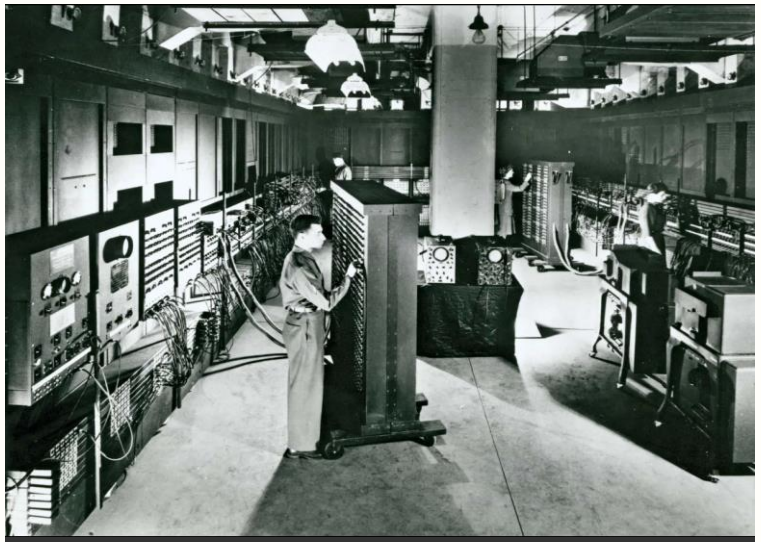


ENIAC

ENIAC (Electronic Numerical Integrator and Computer), c. 1946.

Courtesy of the Moore School of Electrical Engineering, University of Pennsylvania

How did the Computing evolve?



ENIAC

ENIAC (Electronic Numerical Integrator and Computer), c. 1946.

Courtesy of the Moore School of Electrical Engineering, University of Pennsylvania

From

← This to This →

From ENIAC (1946) to Smartphones (2026) today is one of the most fascinating engineering stories ever

“If computers became smaller and more powerful over time, what do you think made this possible?”

“The history of computing is basically the story of how we **kept innovating to pack more capability into smaller hardware while keeping programming manageable**.”

Let’s imagine, how was this made possible?

Engineers Always Ask THREE Questions

1. Why?

- Why are we trying to create this?. Find the purpose and motivation.

2. What?

- What are going to build?. Create the spec requirements.

3. How?

- How are we going to accomplish this. Detail the Process, Strategy and actions needed

What's a nice way to understand the evolution from ENIAC to Smartphone?

Let's go back in time (to 1900's) and walk up to the present day..

Who else is better equipped than an Electrical/Electronics/Computer engineer for this journey?!

Shall we?

The Evolution of the Microcontroller: From Discrete Logic to ARM

1947–1960s: The Building Blocks

1947: The Transistor Revolution

Bell Labs replaces bulky, failing vacuum tubes with the first solid-state semiconductor switch.

Discrete Logic Families

Engineers combined transistors into RTL, DTL, and the industry-standard TTL logic gates.

From Gates to Memory

Logic gates were wired into adders for arithmetic and flip-flops for sequential memory.

1970s–Modern Day: The Birth of the MCU

Microprocessor (MPU)

MPUs contain only a CPU.

Microcontroller

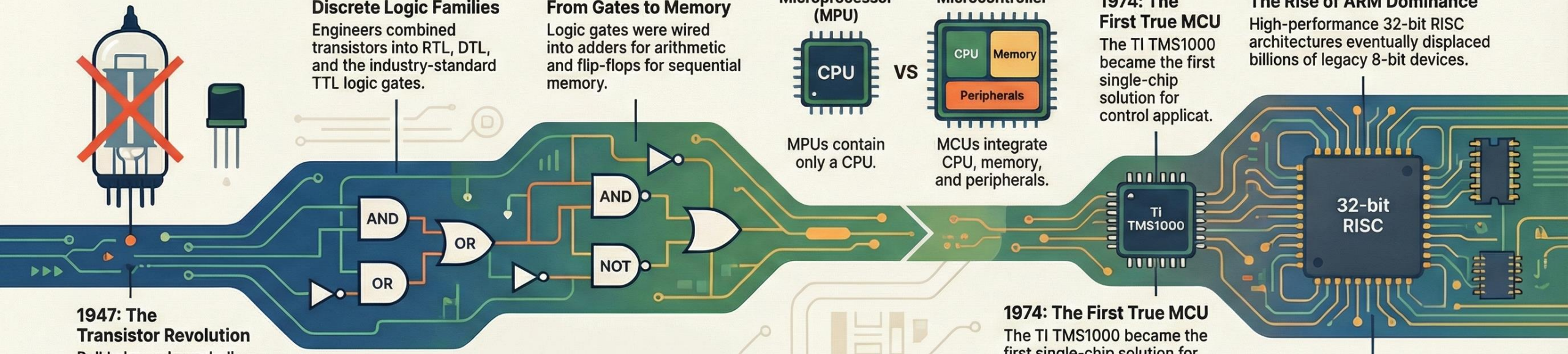
MCUs integrate CPU, memory, and peripherals.

1974: The First True MCU

The TI TMS1000 became the first single-chip solution for control applicat.

The Rise of ARM Dominance

High-performance 32-bit RISC architectures eventually displaced billions of legacy 8-bit devices.

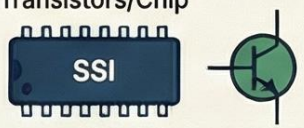


1974: The First True MCU

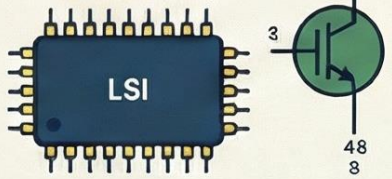
The TI TMS1000 became the first single-chip solution for control applications.

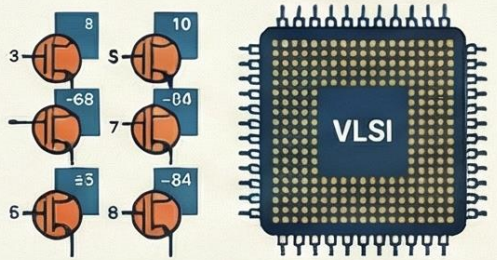
> 10,000 Transistors/Chip

1–20 Transistors/Chip



200–10,000 Transistors/Chip





1960s → 1970s

SSI (Small-Scale)
e.g., 7400 Series Logic Gates

LSI (Large-Scale)
e.g., Intel 4004 Microprocessor

VLSI (Very Large-Scale)
e.g., Intel 8051 Microcontroller

Evolution of

ICs, CPU and MCU

Architectural progress shaping
Embedded Systems innovation

Agenda

- Historical Development of Integrated Circuits
- Historical Development of Integrated Circuits first CPU and MCU

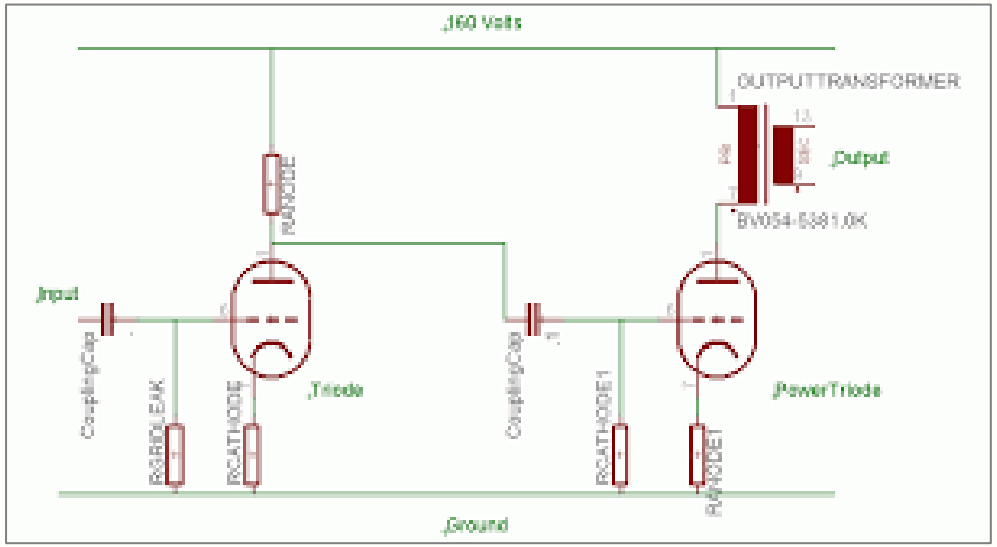
Historical Development of Integrated Circuits

Quick Preview - Journey to the first CPU on a chip

1. Electronic Valves (Vacuum Tubes), Point contact diodes — ~1904–1950s
2. First Bipolar Junction Transistor (BJT) — 1947
3. Metal-Oxide-Semiconductor Field-Effect Transistor (MOSFET) — 1959
4. Logic Gates (Combinational Logic) — 1960s
5. Adder / Subtractor ICs — Late 1960s–1970s
6. Flip-Flops and Memory ICs (Sequential Logic) — 1970s
7. Microprocessor (CPU on a Chip) — 1971
8. MCU (CPU, Memory, I/Os) - 1974

Electronic Valves (Vacuum Tubes) — 1904–1950s

Technology: Thermionic emission in vacuum.
Examples: Triode tubes used in early computers like ENIAC as switches
Limitations: Very large; Consumed high power; Generated heat; Frequent failure;
This motivated the search for solid-state devices.



Why couldn't engineers just keep using vacuum tubes?

Point-Contact Diode — 1904

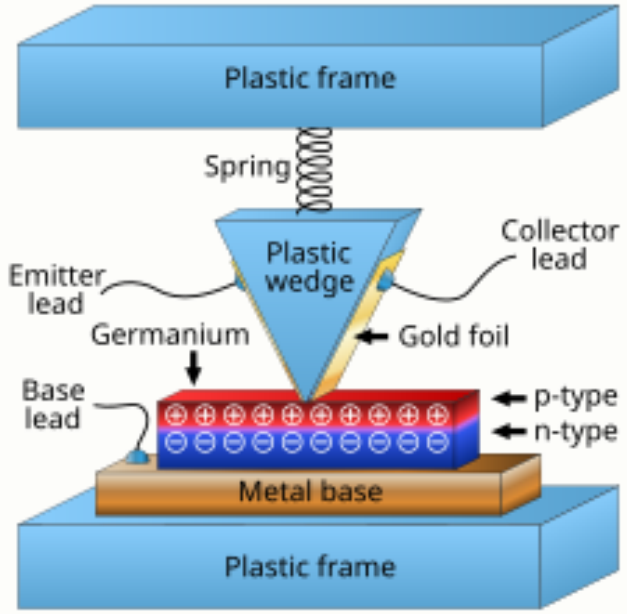
The **point-contact diode** is one of the earliest semiconductor devices used for detecting radio signals.

Invention Year: 1904
Inventor: Jagadish Chandra Bose
(demonstrated crystal detector using galena-PbS)
Later developed and patented by Greenleaf Whittier Pickard in 1906
How it worked :A fine metal wire (“cat’s whisker”) lightly touches a semiconductor crystal forming a rectifying junction

Applications:
Used as a radio detector in early crystal radios
Converted RF signals → audio signals
Importance in electronics evolution
One of the first semiconductor devices
Demonstrated rectification without vacuum tubes
Conceptually led to later semiconductor diodes and transistors

1N34A – Historic packaged Germanium Diode 1946

The First Electronic Switches: Vacuum Tubes to the Point Contact Transistor (1947)



Invented at Bell Labs 1947
By John Bardeen, Walter Brattain, and William Shockley.

Key idea: Semiconductor device that controls current using PN junctions.

Advantages over vacuum tubes:
Much smaller; Low power consumption; More reliable; No warm-up time

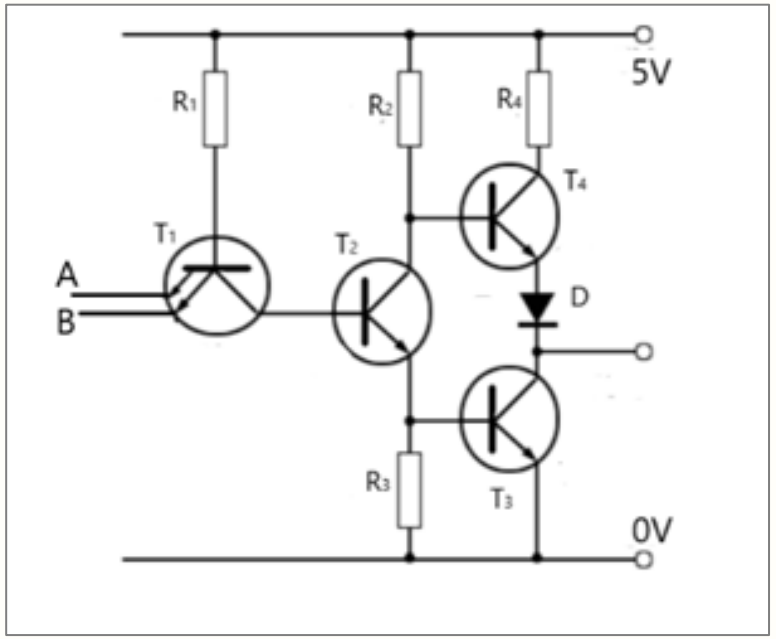
Applications: Early transistor radios; Digital switching circuits; Foundation of modern electronics

The first commercially available packaged point-contact transistor was the Raytheon CK703, announced in late 1948.

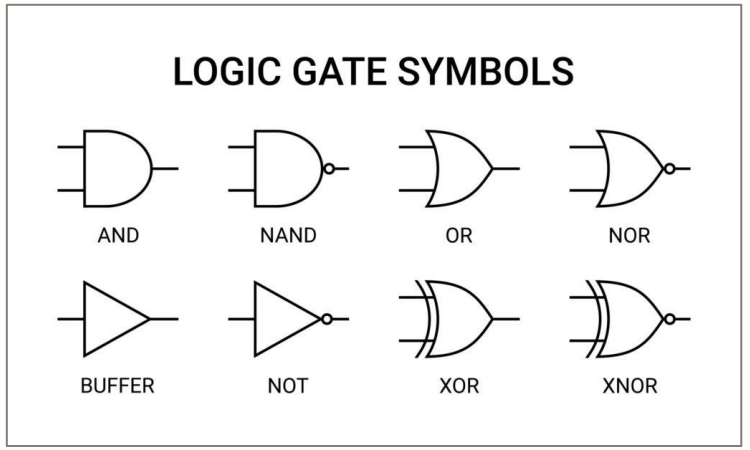
Yes, now we know Point Contact Transistors can replace electronic valves as switches. Bit more reliable, smaller than electronic valves..

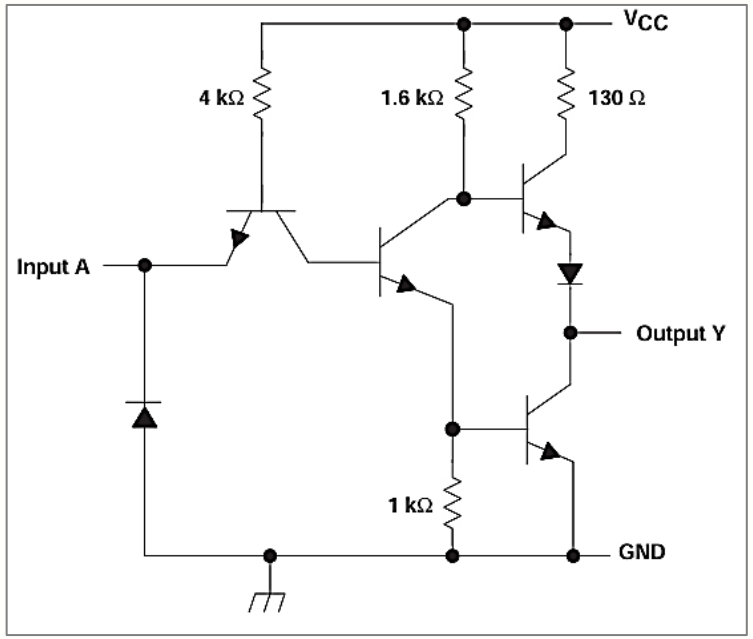
So… What more possible with Transistors?

Applications using Transistors – Logic Circuits



TTL "2 input NAND gate" (Totem pole Output)



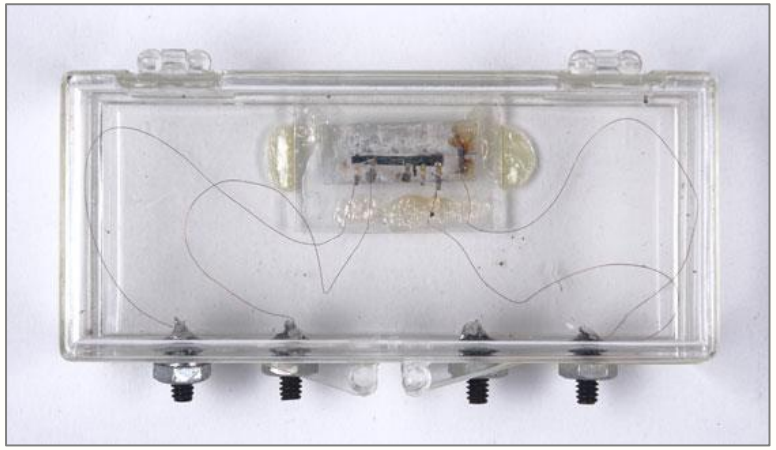


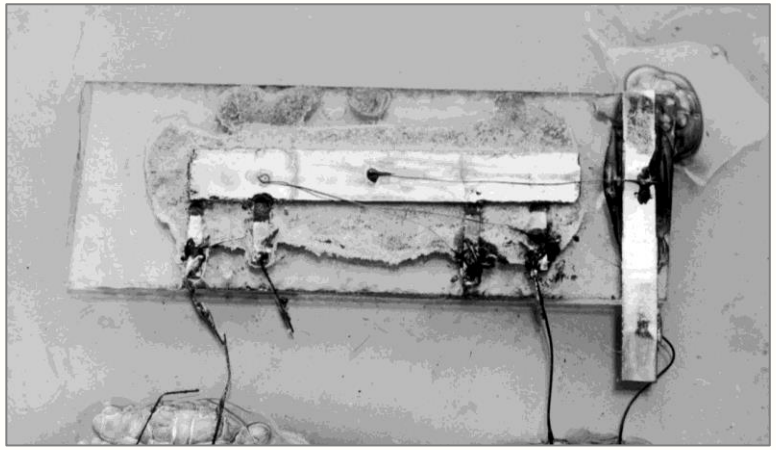
TTL "NOT gate" (Totem pole Output)

Okay, transistors are cool switches and we can create whole Logical circuits with it But..

What problem appears when circuits need 100’s or 1000’s of transistors?

First Integrated Circuit (Germanium IC) — Jack Kilby (1958)

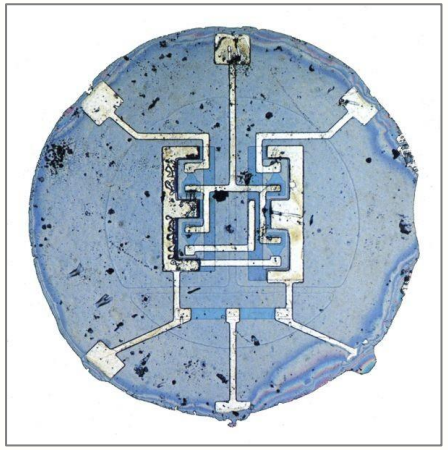




Inventor: Jack Kilby
Company: Texas Instruments
Year: 1958
Material: Germanium
Key idea
All components (transistor, resistor, capacitor for an oscillator) built on **one semiconductor piece**.

Prototype
Oscillator circuit
Connected using **fine gold wires**
Limitations
Difficult to mass produce
Not ideal for large-scale manufacturing
Despite limitations, this proved **monolithic integration was possible**.

Practical Silicon IC — Robert Noyce (1959)



Inventor: Robert Noyce
Company: Fairchild Semiconductor
Year: 1959

Material: Silicon

Key breakthrough: Used the planar process developed by Jean Hoerni.

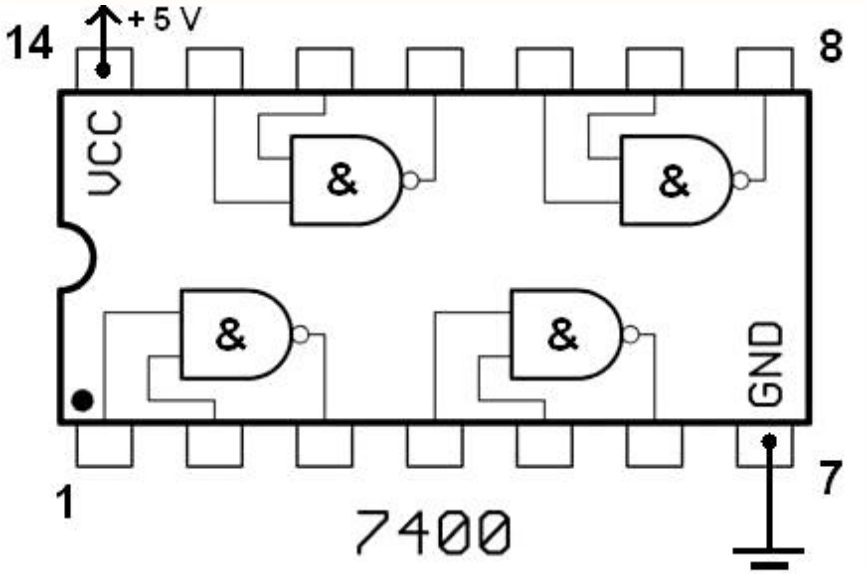
Advantages
Easier mass production
Reliable metal interconnections
Scalable manufacturing

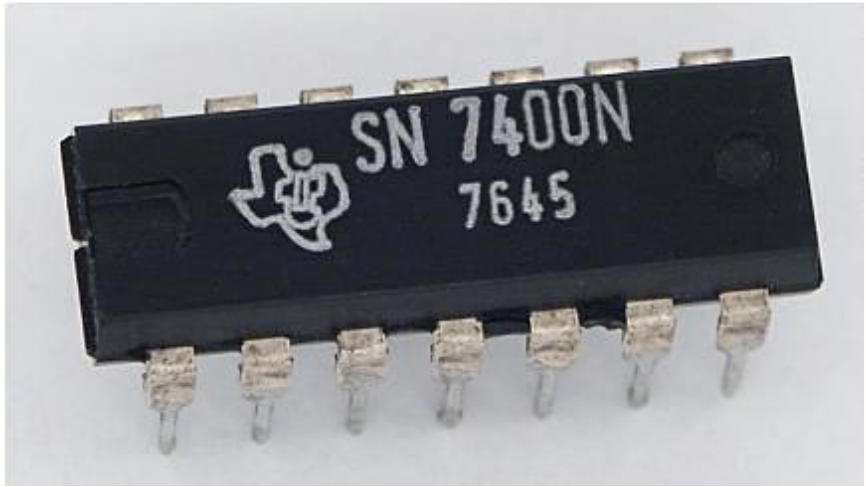
This design became the foundation of modern IC fabrication.

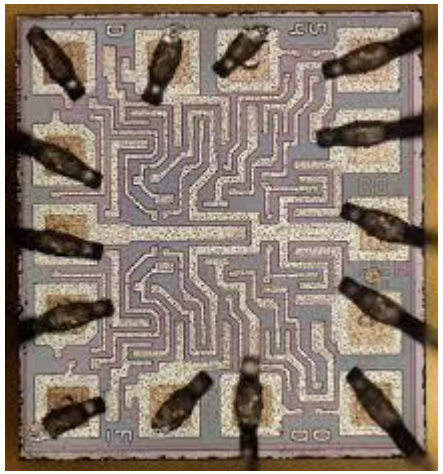
Logic IC families using Discrete Transistors

Logic Family	Full Name	Basic Gate	Power Consumption	Speed	Noise Margin	Cost (1960s)	Typical IC Example
RTL	Resistor-Transistor Logic	NOR	High	Medium	Low	Very low	Fairchild μL900
DTL	Diode-Transistor Logic	NAND	Medium	Medium	Good	Low	930 series
TTL	Transistor-Transistor Logic	NAND	Medium	Fast	Excellent	Moderate	Texas Instruments 7400 (1964)

Transistor–Transistor Logic ICs (TTL) — Early 1960s





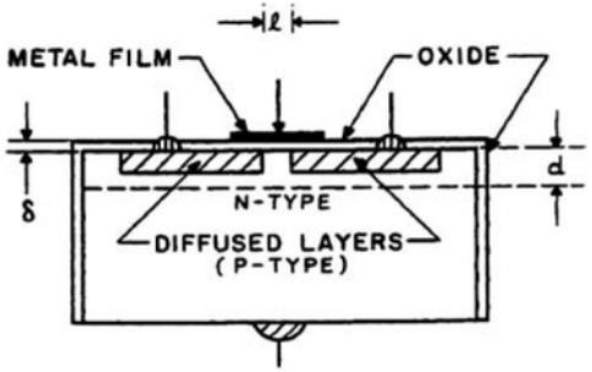


Die image of
7400 -Quad 2input NAND

Introduced: ~1961–1963
Developed by: Texas Instruments
Key Features:

- Logic built using BJTs
- Fast switching speeds
- Operated typically at 5 V
- Higher power consumption

Innovation of the MOSFET — 1959



MOSFET (Metal–Oxide–Semiconductor Field Effect Transistor)
Invented in 1959 at Bell Labs by
Mohamed Atalla, Dawon Kahng
Their contributions were complementary and together solved a major challenge in semiconductor technology.

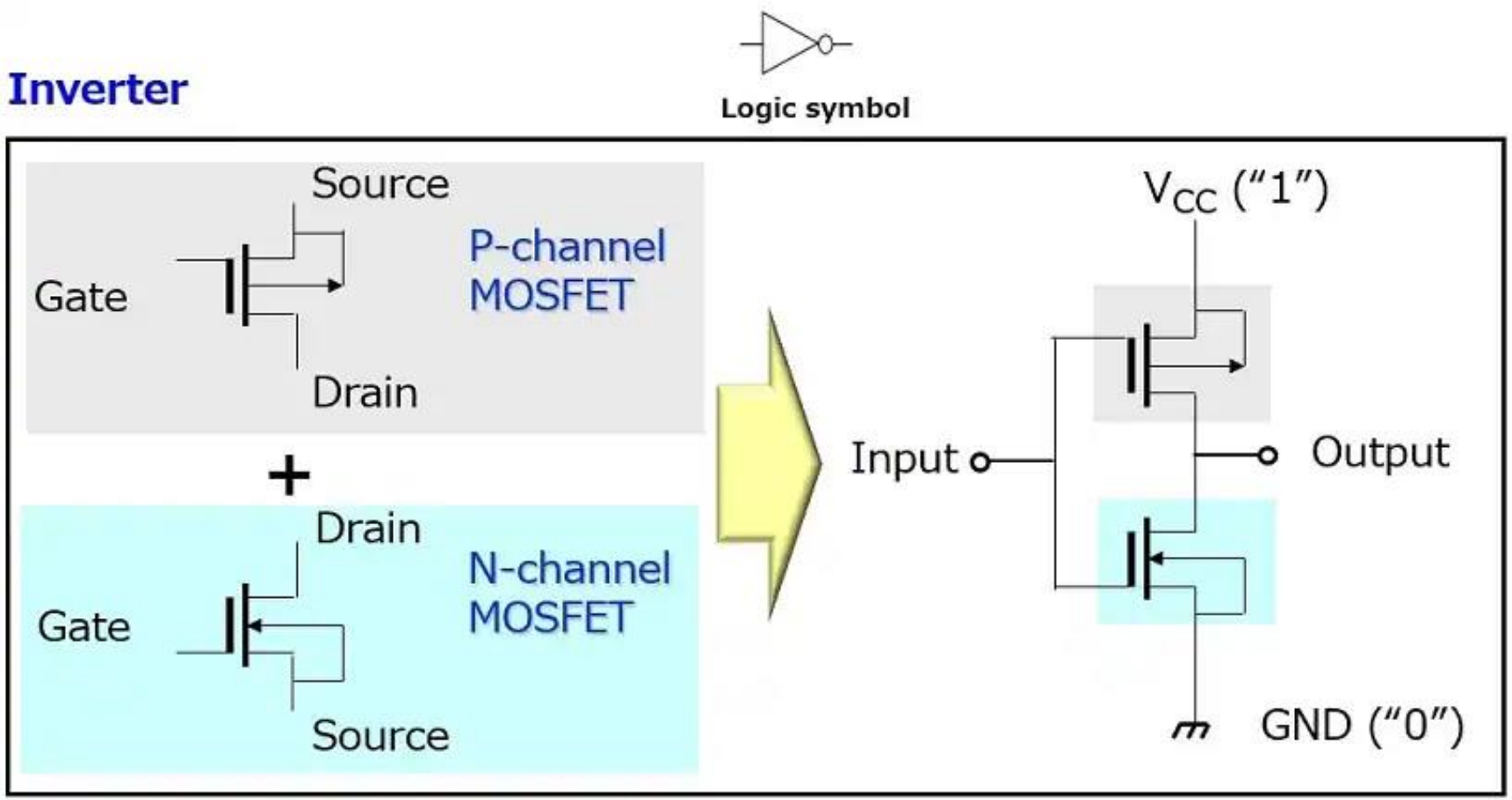
Together they created the first successful MOSFET in 1959 at Bell Labs.

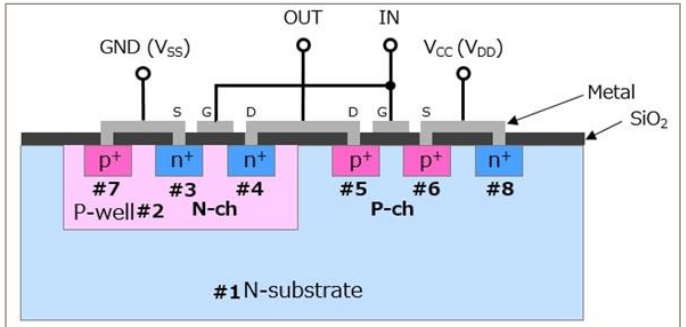
Device structure

Metal Gate
↓
Silicon Dioxide (Insulator)
↓
Silicon Substrate

Gate voltage → controls channel conductivity.

Logic Circuits using MOSFETs – NOT Gate

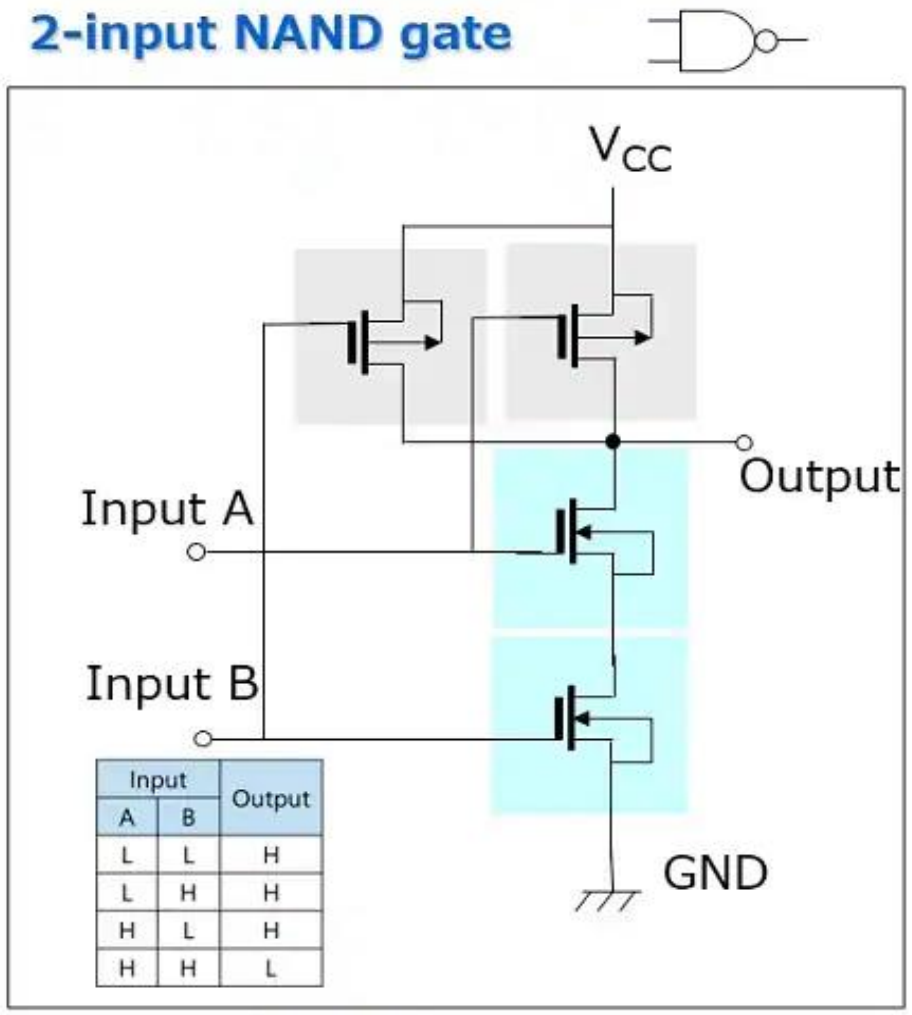


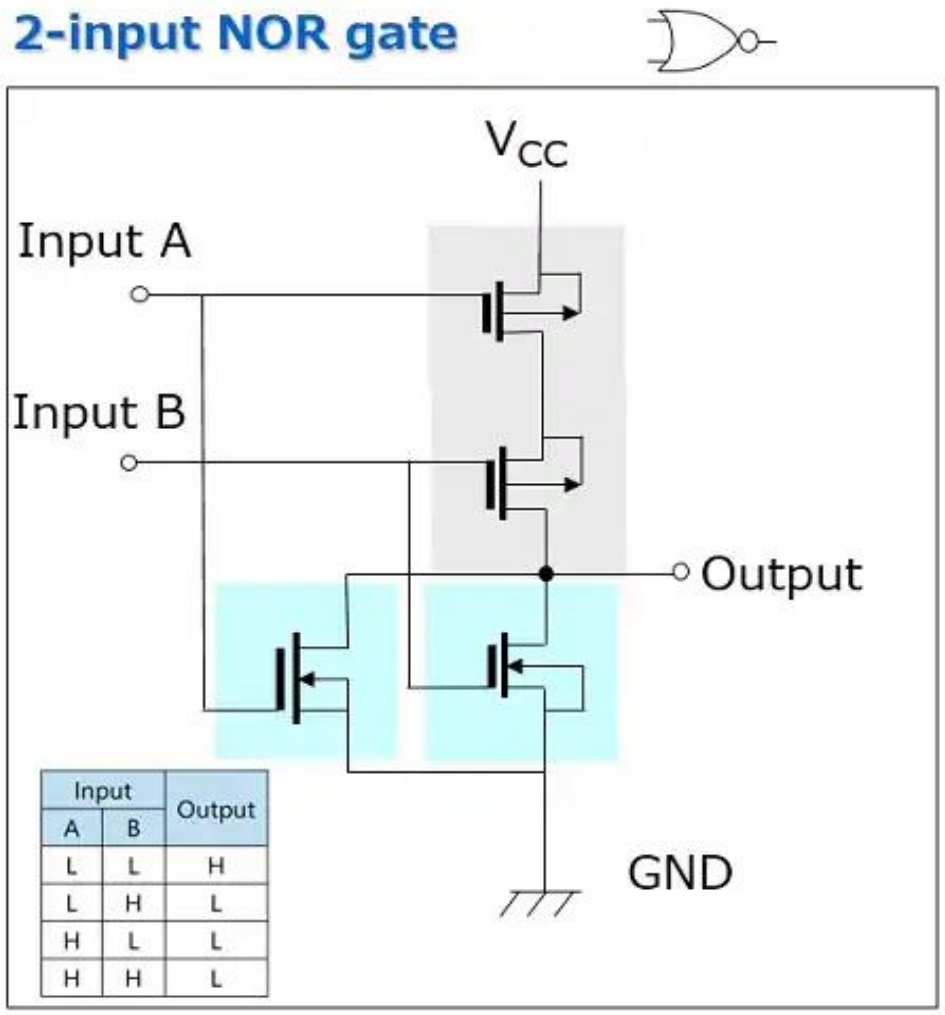


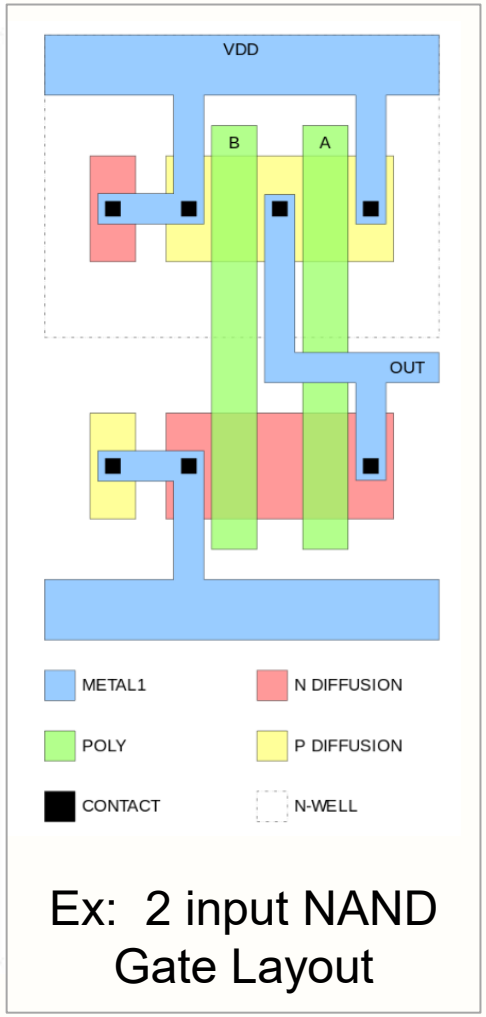
Ex: Die Cross section of a NOT gate

Image Courtesy: Toshiba

Logic Circuits using MOSFETs – NAND, NOR



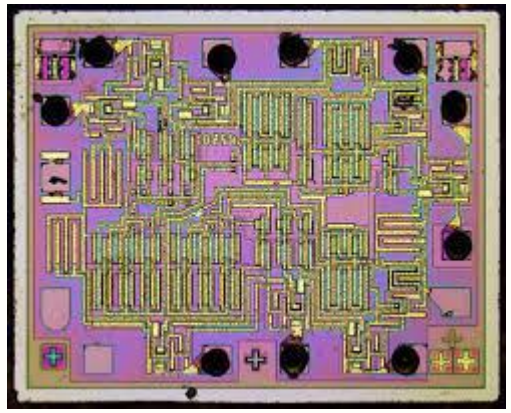




Ex: 2 input NAND Gate Layout

Image Courtesy: Toshiba

CMOS Logic ICs — Late 1960s



First Commercial Mfr: RCA
Year of Introduction: 1968
Product Family: CD4000 Series

CMOS Logic ICs Marketed under the name COSMOS (Complementary Symmetry MOS).

Key Innovations:

RCA engineers developed complementary MOS technology (CMOS) using both: NMOS transistors PMOS transistors

This resulted in: Very low static power consumption; High noise immunity; Wide operating voltage range (3V–15V).

Fundamentals: NMOS vs PMOS die area difference

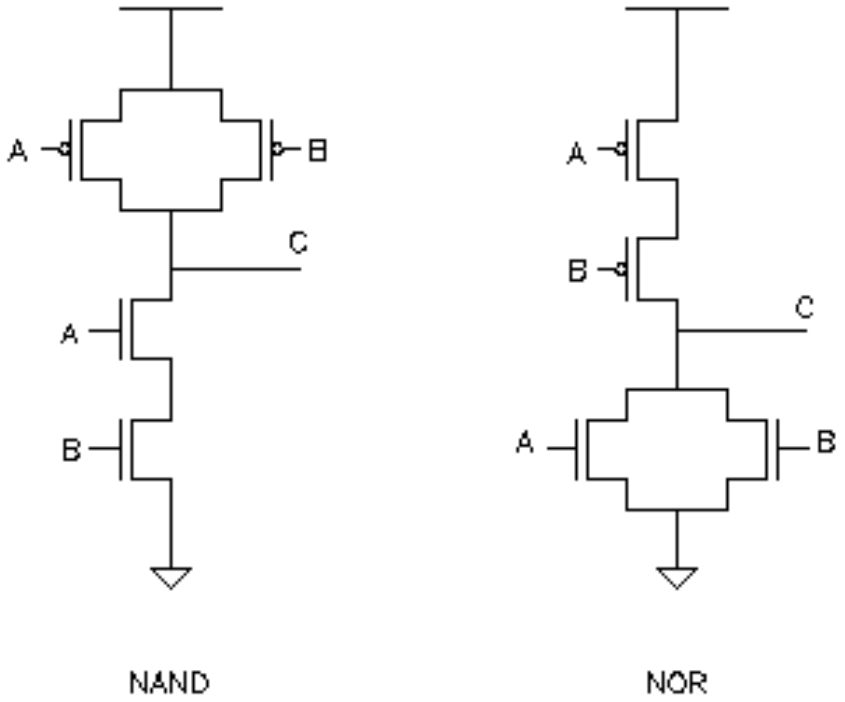
- **Charge Carrier Mobility:** NMOS transistors use electrons with higher mobility (~1400 cm^2/Vs), allowing smaller sizes, while PMOS uses holes with lower mobility (~450 cm^2/Vs), requiring wider channels and roughly doubling die area.

- **Switching Speed & Sizing:** NMOS switches faster; PMOS needs to be upsized, increasing area by 1.5–2x in balanced CMOS gates.

- **Layout Impact:** PMOS sizing largely determines overall CMOS area.

Fundamentals: Why NAND is preferred in VLSI designs

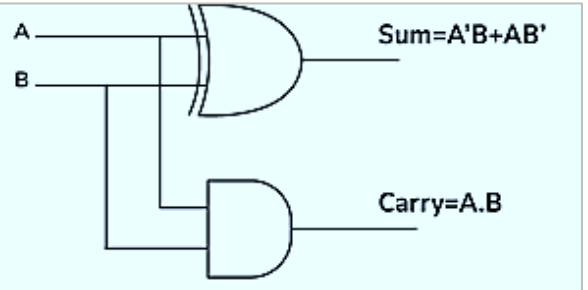
- CMOS NAND gates use parallel PMOS and series NMOS, allowing smaller PMOS sizes and a compact layout, while CMOS NOR gates have series PMOS and parallel NMOS, requiring wider PMOS and larger area.
- Higher electron mobility in NMOS enables smaller NAND designs; lower hole mobility in PMOS leads to bigger NOR transistors and increased die size.
- **NAND gates are more area-efficient** (6-8 λ^2) compared to NOR gates (8-12 λ^2), making NAND preferable in VLSI standard cell libraries.



Now that we have TTL, CMOS Logic ICs moving to mass manufacturing in a nice tiny IC package, what more we can do?

Combinational Logic: Adders and Subtractors

The first arithmetic circuits were built entirely from gates. A subtractor is simply an adder with one input inverted (using XOR gates to create 2's complement). The same full-adder block can perform both addition and subtraction by controlling the carry-in and inverting one operand.

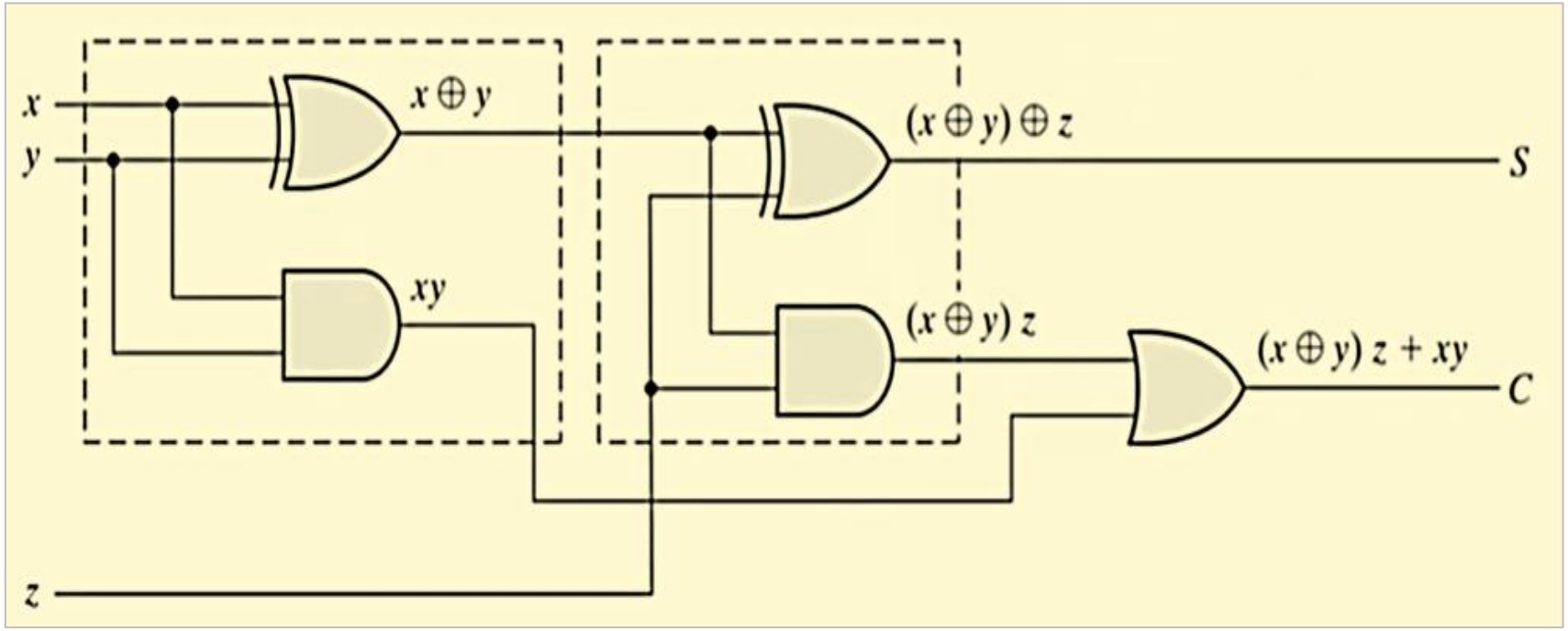


Truth Table

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Half Adder

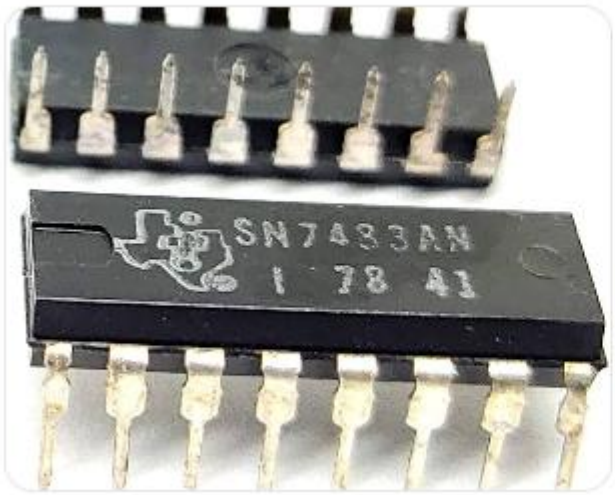
(Adds two bits)

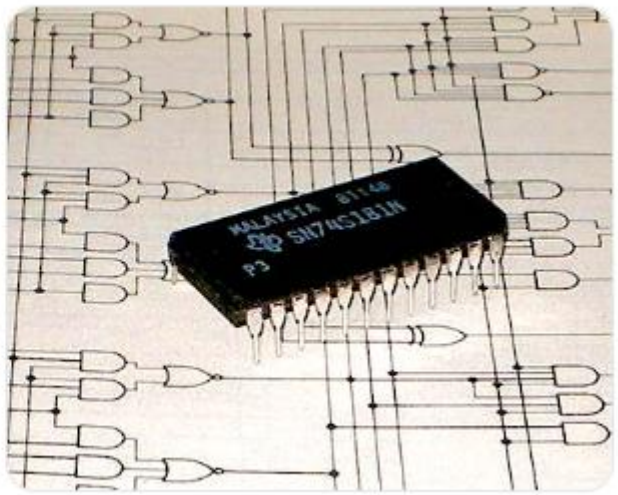


Full Adder

(Adds two bits and Carry)

Arithmetic ICs (Adders / ALU) — Late 1960s–1970s





- Examples
- **7483** → 4-bit adder
- **74181** → Arithmetic Logic Unit
- These chips implemented **basic arithmetic operations for CPUs**.

DM74LS181
4-Bit Arithmetic Logic Unit

General Description

The DM74LS181 is a 4-bit Arithmetic Logic Unit (ALU) which can perform all the possible 16 logic operations on two variables and a variety of arithmetic operations.

Features

- Provides 16 arithmetic operations: add, subtract, compare, double, plus twelve other arithmetic operations
- Provides all 16 logic operations of two variables: exclusive-OR, compare, AND, NAND, OR, NOR, plus ten other logic operations
- Full lookahead for high speed arithmetic operation on long words

Okay. We have Arithmetic and Logic ICs possible now..

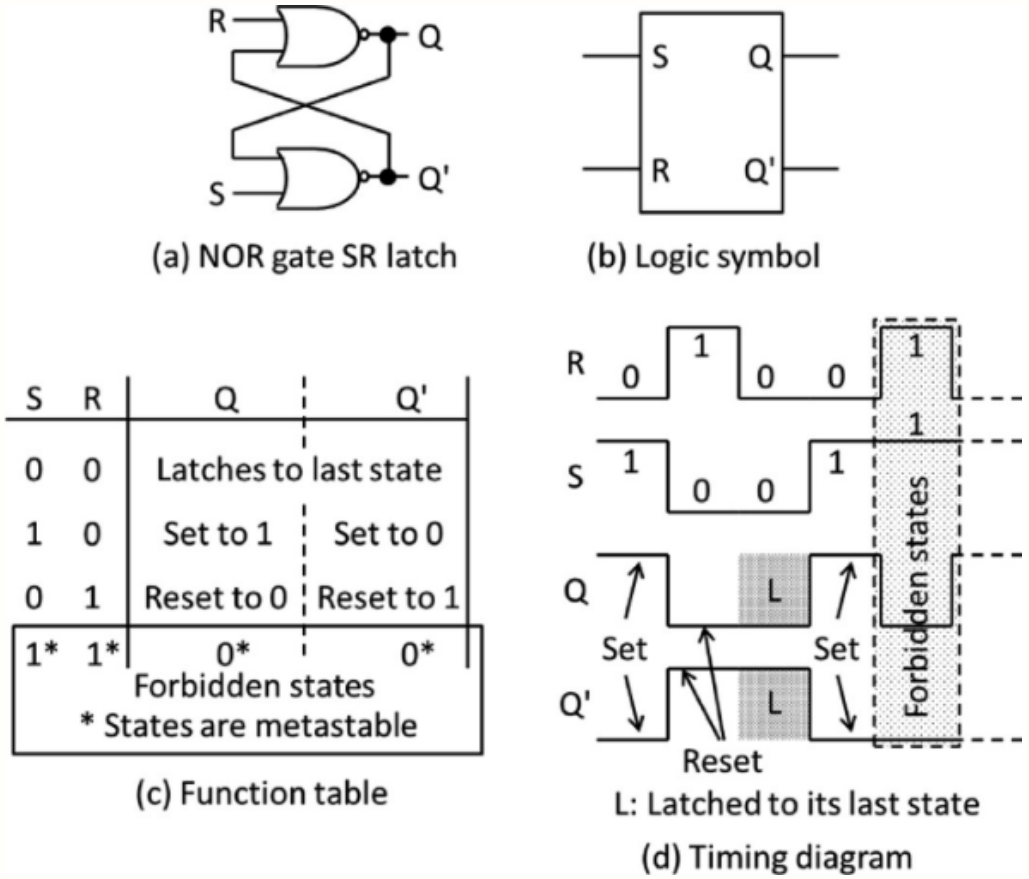
But can a circuit remember like we have memory?

Sequential Logic: SRLatch

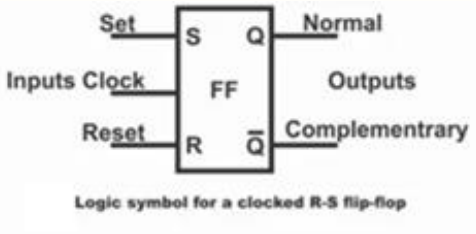
- Combinational circuits have no memory. In order to store data, we need feedback — the birth of sequential logic.

SR Latch (Set-Reset)

- Built from two cross-coupled NOR (or NAND) gates.
  - S = 1 → Q = 1 (Set)
  - R = 1 → Q = 0 (Reset)
  - S = R = 0 → holds previous state

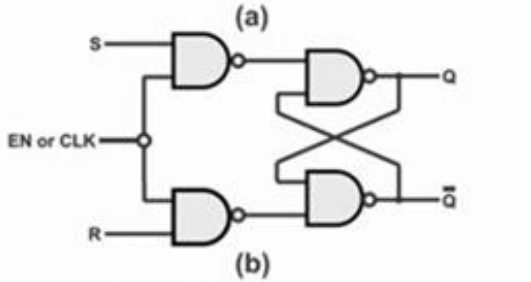


Sequential Logic: Flip Flops



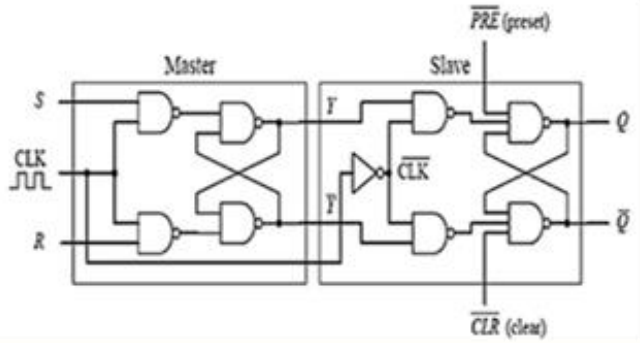
Logic symbol for a clocked R-S flip-flop

Mode of operation	INPUTS			INPUTS		
	CLK	S	R	Q	$\bar{Q}$	Effect on output Q
Hold	⎍	0	0			No change
Reset	⎍	0	1	0	1	Reset or cleared to 0
Set	⎍	1	0	1	0	Set to 1
Prohibited	⎍	1	1	1	1	Prohibited-do not use

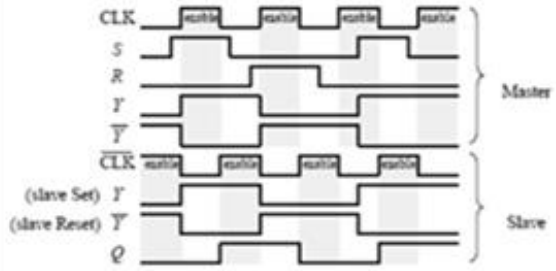


(a) Truth table for a clocked R-S flip-flop
(b) Wiring a clocked R-S flip-flop using NAND gates.

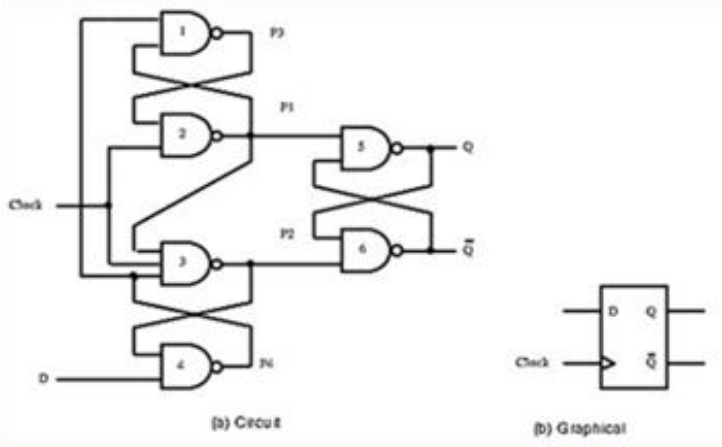
Pulse Triggered Master Slave SR Flip Flop



PRE	CLR	CLK	S	R	Q	$\bar{Q}$	Mode
0	1	X	X	X	1	0	preset
1	0	X	X	X	0	1	cleared
0	0	X	X	X	1	1	not used (race)
1	1	⎍	0	0	0	0	hold
1	1	⎍	0	1	0	1	Reset
1	1	⎍	1	0	1	0	Set
1	1	⎍	1	1	1	1	not used (race)

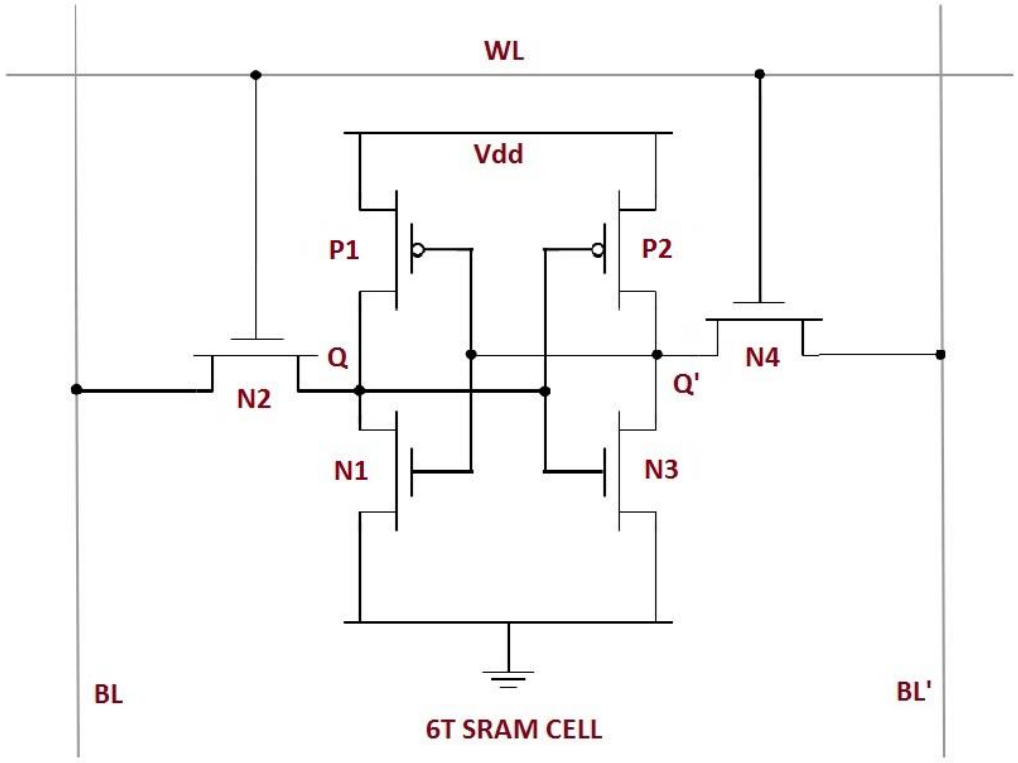


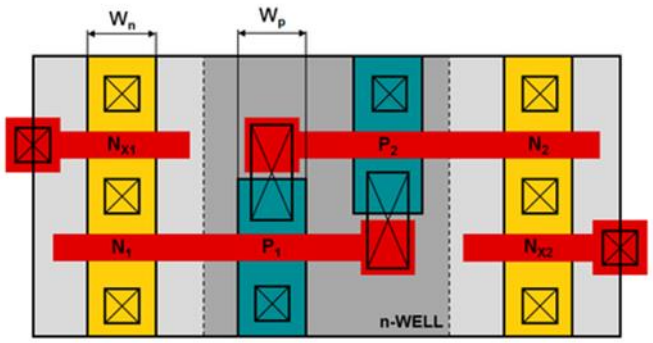
Positive edge triggered D flip-flop



Sequential Logic: 6T SRAM Cell

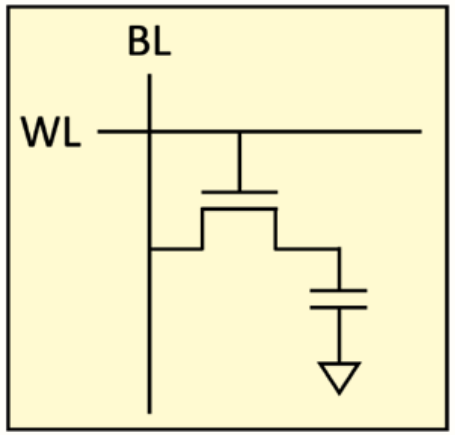
- The basic SRAM (Static Random Access Memory) cell uses a cross-coupled inverter pair, which functions as a bistable latch or flip-flop.
- Specifically, it behaves like an SR flip-flop (Set-Reset flip-flop) in its core structure, where the two inverters hold the state (0 or 1) indefinitely as long as power is supplied.
- This is implemented in the standard 6T SRAM cell (six transistors: four for the flip-flop latch and two for access).

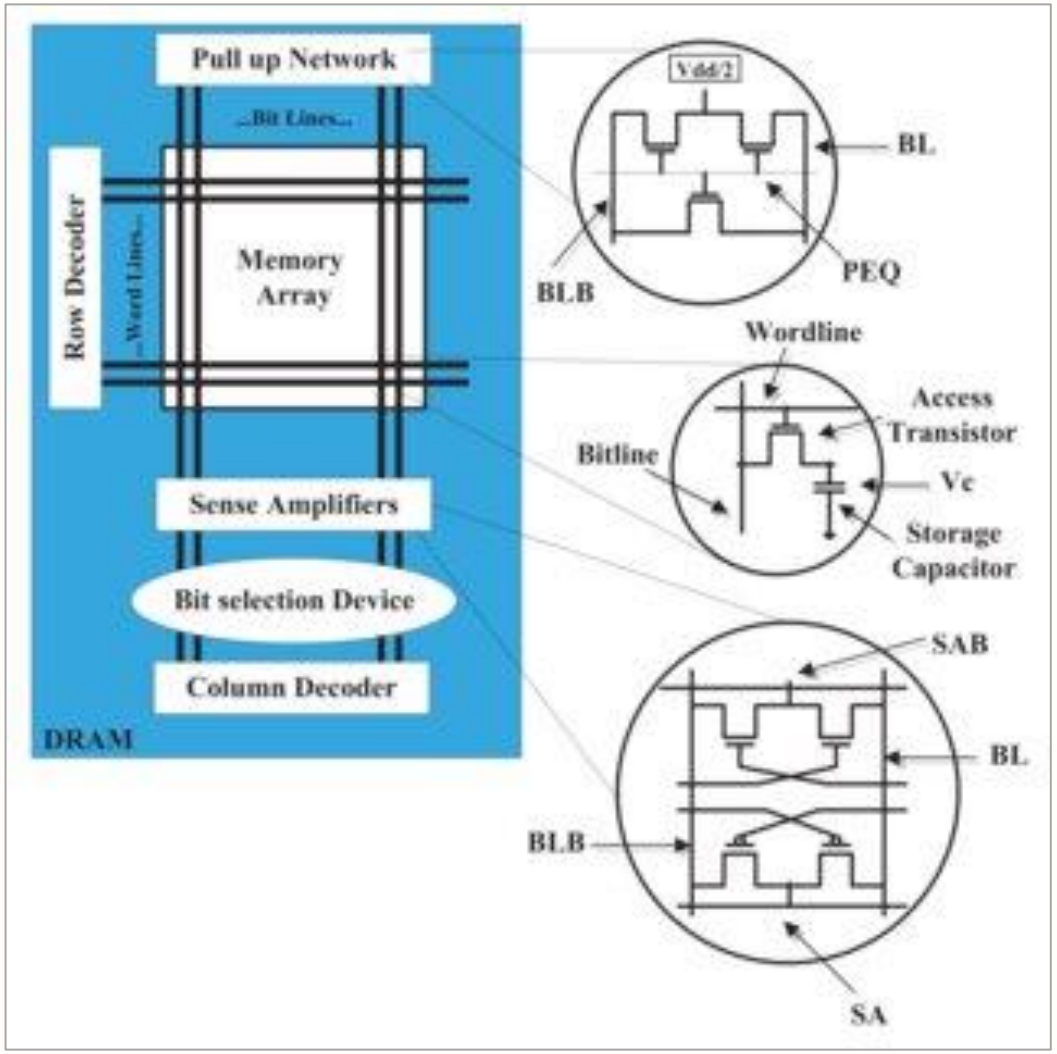




Dynamic RAM (DRAM) – 1-Transistor + 1-Capacitor Cell

- Invented in the early 1970s (Intel 1103, 1970). One transistor charges a tiny capacitor. The charge leaks, so the cell must be refreshed every few milliseconds.
- Much smaller and cheaper than SRAM but requires refresh circuitry.





Flip-Flops and Memory ICs — 1970s

Examples

- 7474 D Flip-Flop
- SRAM
- DRAM

These enabled registers, buffers

Okay. We have ALU, SRAM which is volatile.

But for a programmable computer, we would need a nonvolatile memory as well, isn’t it?

Non-volatile Program Memory – Mask ROM

Adopts a NAND structure for increased integration (1 transistor cell).

Data Write Method

- Information written in the wafer process
- ‘1’ : **Ions implanted in the transistor**
- ‘0’ : No ion implantation

Data Read Method

- Word line potential of the Read Cell is 0V
- Word line potential of non-Read Cell is Vcc
- Voltage is supplied to the Bit Line
- ‘1’ is determined if current flows

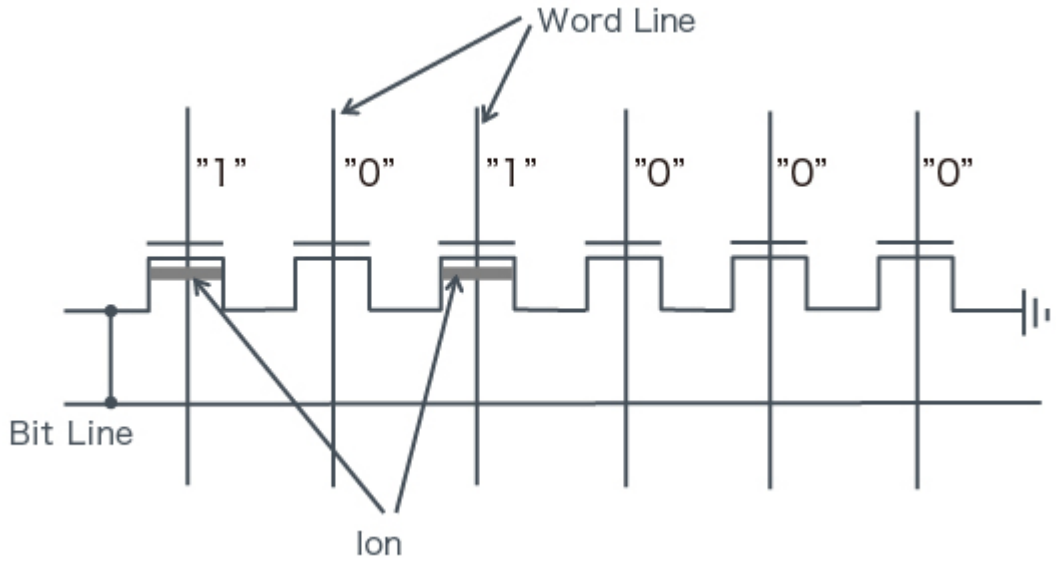


Image Courtesy: ROHM

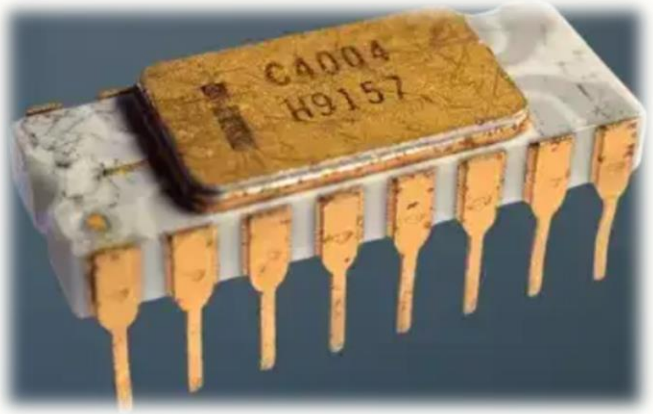
So, What was the first application for which a first Microprocessor was created?

Historical Development of CPU and MCU

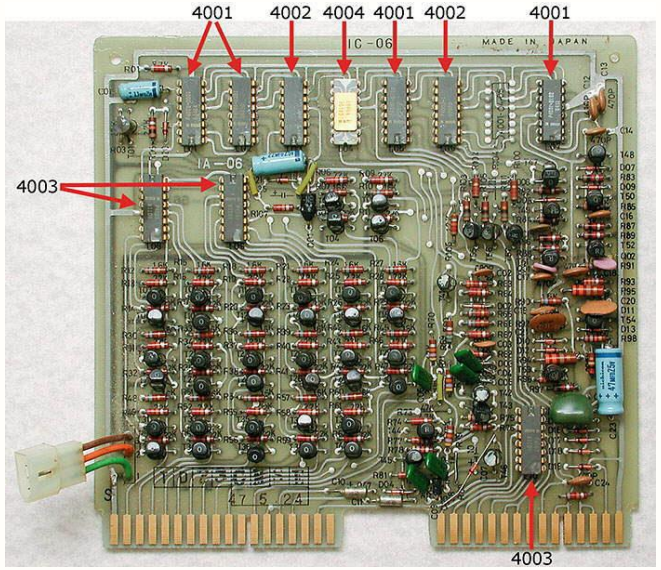
First requirement for a Microprocessor

- In 1969, a Japanese company called **Busicom** wanted to build a new programmable calculator.
- Their initial design required 12 different custom chips to implement the logic for the calculator.
- They approached a young company called **Intel** to manufacture those chips.
- At Intel, **an engineer named Ted Hoff** looked at the design and realized something interesting.
- Instead of building many fixed-function chips, what if we build one programmable chip that could run instructions stored in memory?
- That single idea led to the creation of the **Intel 4004 in 1971**.
- “That chip had about **2300 transistors** and ran at **740 kHz**.

World's first Microprocessor, Intel 4004 (1971)



Intel 4004 Schematic



Intel 4004 was a **4-bit CPU** consisting of ALU, Control Unit, Registers and Instruction Decoder.

- To build a calculator, it also needed
- Intel 4001 – Mask ROM(Program Memory),
- Intel 4002 (RAM)
- Intel 4003 – Shift Register I/O (Peripheral interface) MCS4_Data_Sheet_Nov71.pdf

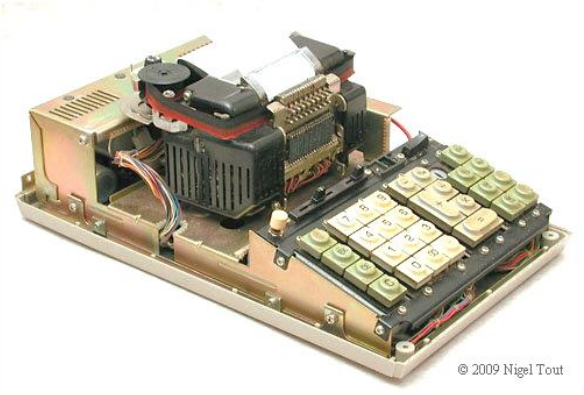


Image Courtesy: Busicom 141-PF

Program Memory Used Before Flash (1970s–1980s)

MASK ROM

The program was permanently encoded during chip fabrication. The ROM pattern was defined by the metal mask layer.

Characteristics

Very cheap in large production; Very fast; Not reprogrammable

Examples

Intel 8048
Intel 8051 (original versions)

Manufacturers produced custom chips with the user's firmware burned into ROM.

PROM

The Programmable Read Only Memory (PROM) allowed one-time programming by the user.

Operation

Internal fuse links are blown using a PROM programmer. Once programmed, it cannot be erased.

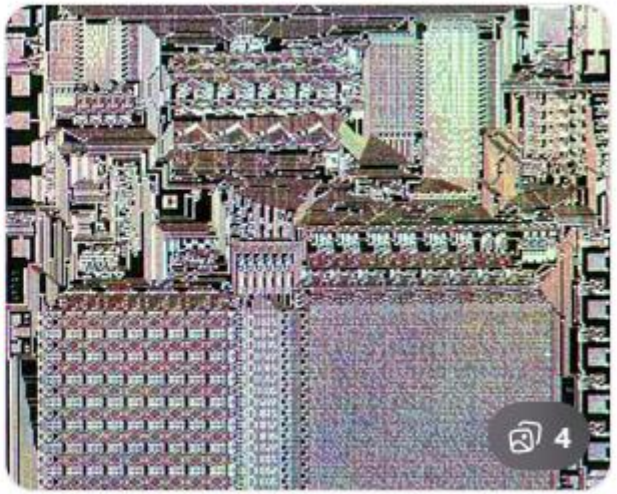
Usage

Low-volume products

UVEPROM

Programmable using programmer and erase by exposing the die to UV Lamp for 20~30 minutes
EEPROM : Electrically erasable but very costly.

World's first general purpose Microcontroller, TMS 1000 (1974)



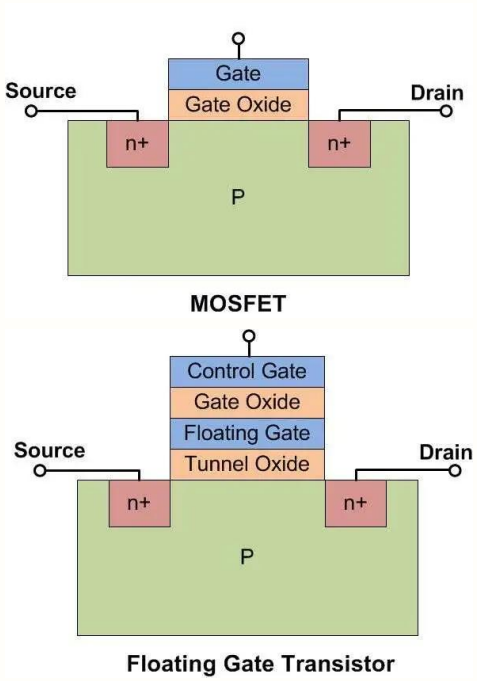
The world's first commercially available microcontroller was the TMS1000, developed by Gary Boone of Texas Instruments (1974).

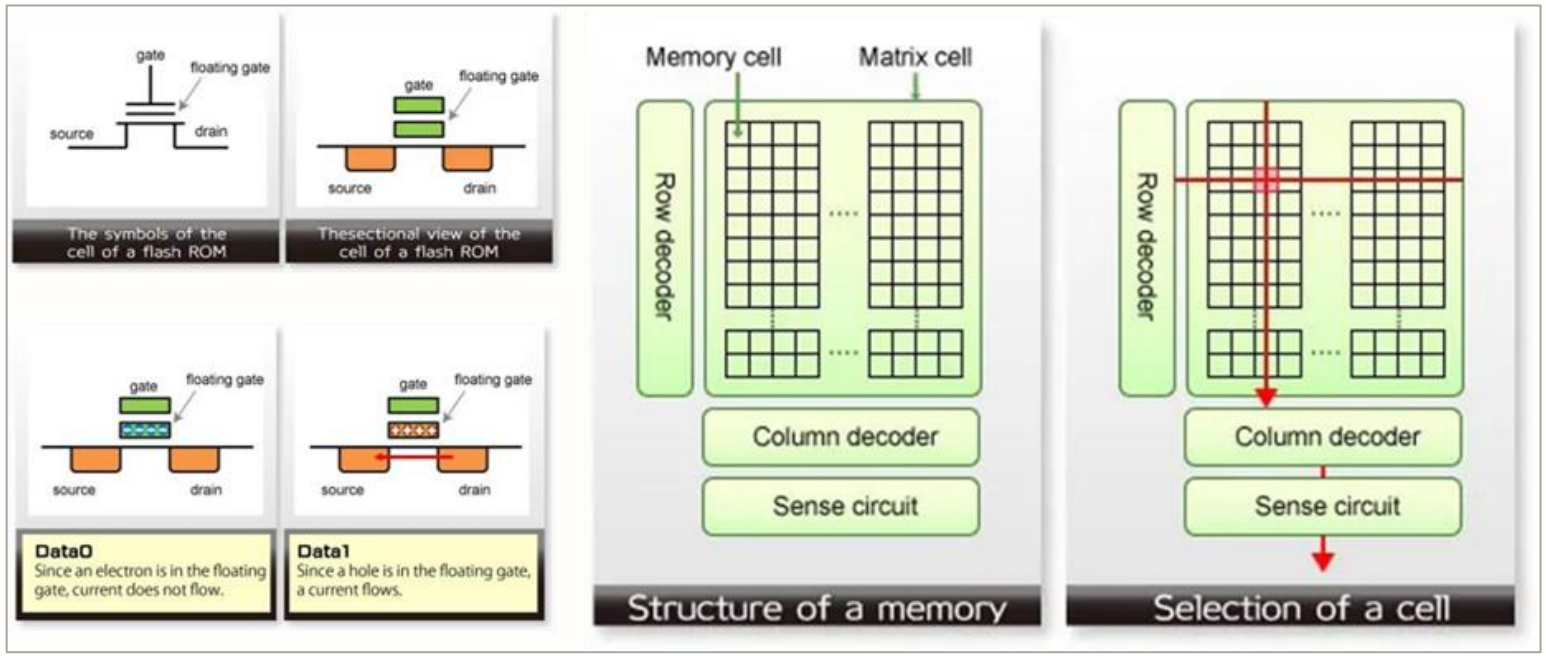
The TMS1000 integrated multiple components on one chip, including: Program ROM (1K Byte), RAM (64 Bytes), Control unit and I/O

1976_TI_TMS_1000_Series_Data_Manual_MOS_LSI_One-Chip_Microcomputers_197602.pdf

Flash: 1-Transistor with floating gate structure

- The 1-transistor floating-gate memory cell was invented by: **Fujio Masuoka**, Working at **Toshiba (1980~1984).**
- Masuoka and his team developed a floating-gate MOSFET memory cell, where charge stored on an isolated floating gate represents digital data (0 or 1).
- Masuoka named it **Flash** memory because entire blocks of memory can be erased in a single flash operation rather than one byte at a time.





Summary - The March of Integration SSI to VLSI

Integration Level	Approx. Transistors per IC	Typical Logic Complexity	Example Devices
SSI (Small Scale Integration)	**1 – 100**	Basic logic gates	NAND, NOR gates (7400 series)
MSI (Medium Scale Integration)	**100 – 3,000**	Adders, counters, multiplexers	4-bit adders, decoders
LSI (Large Scale Integration)	**3,000 – 100,000**	Simple processors, memory chips	Early microprocessors, RAM
VLSI (Very Large-Scale Integration)	**100,000 – millions**	Microprocessors, complex ICs	CPUs, DSPs
ULSI (Ultra Large Scale Integration)*	**Millions – billions**	Modern processors / SoCs	Smartphone SoCs, GPUs

What this Journey Teaches Engineers

Every chip we design today sits on top of decades of engineering effort.

That's why this history should leave us with three things: **Gratitude** for those engineers, **Responsibility** for what we build next, and the **Assertiveness** to push technology further."

Haven’t anyone heard of Moore’s law that held the torch for the five decades of growth in integration of number of transistors into an IC?

Moore's Law

Moore's Law states that:

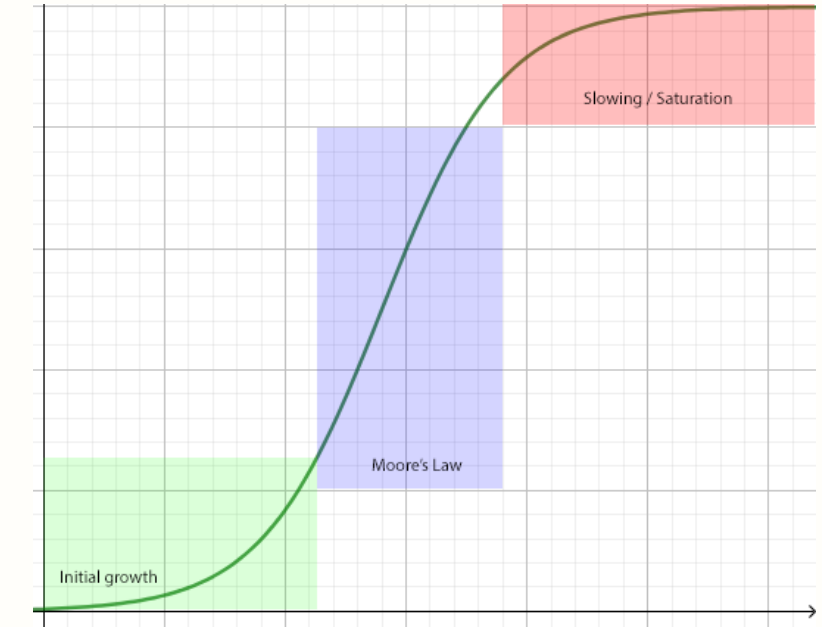
The number of transistors on an integrated circuit doubles approximately every 18–24 months, leading to exponential growth in computing power and reduction in cost per transistor. It was proposed by Gordon Moore in 1965.

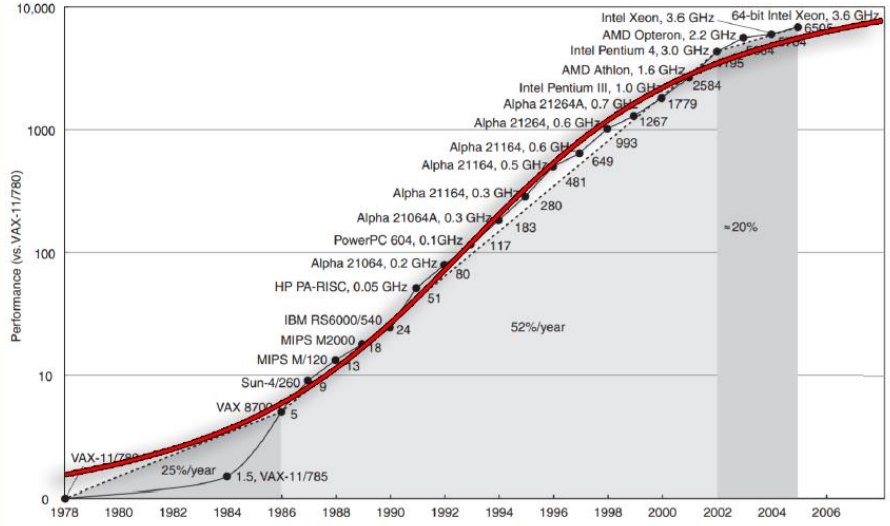
Moore observed that:

Transistor density in ICs was doubling roughly every year.

He later revised the estimate to every two years.

At the time, he worked at Fairchild Semiconductor, before co-founding Intel.





Courtesy: G.Mancusi

Does Moore's Law Still Hold Today?

Short answer: Partially but slowing down.
The trend continues but **not as smoothly as before**.

Reasons for Slowdown

1. Physical Limits
Transistors are now only a few **nanometers** in size.
Challenges include: Quantum tunneling; Leakage currents; Heat dissipation.

2. Rising Fabrication Costs

Advanced nodes (3 nm, 2 nm) require extremely expensive equipment.
Example lithography machines from ASML cost **>$150 million each**.

3. Power Density Limits

Increasing transistor counts no longer automatically improves performance due to power and heat constraints.

How Industry Extends Moore's Law (For Processors)

Simple scaling down of process node cannot be the future.

1. Advanced Transistor Architectures and Lithography

Gate-All-Around (GAA) FETs and Nanosheets enabling scaling to 2nm and below. Tools like ASML's High-NA Extreme Ultraviolet (EUV) Lithography allow finer patterns (sub-10nm resolution), extending scaling cost-effectively to achieve 1.4nm by 2027. Backside Power Delivery Network (BSPDN) improving efficiency by 20-30%

2. 3D Integration and Chiplet Designs

3. New Materials and System-Level Optimizations

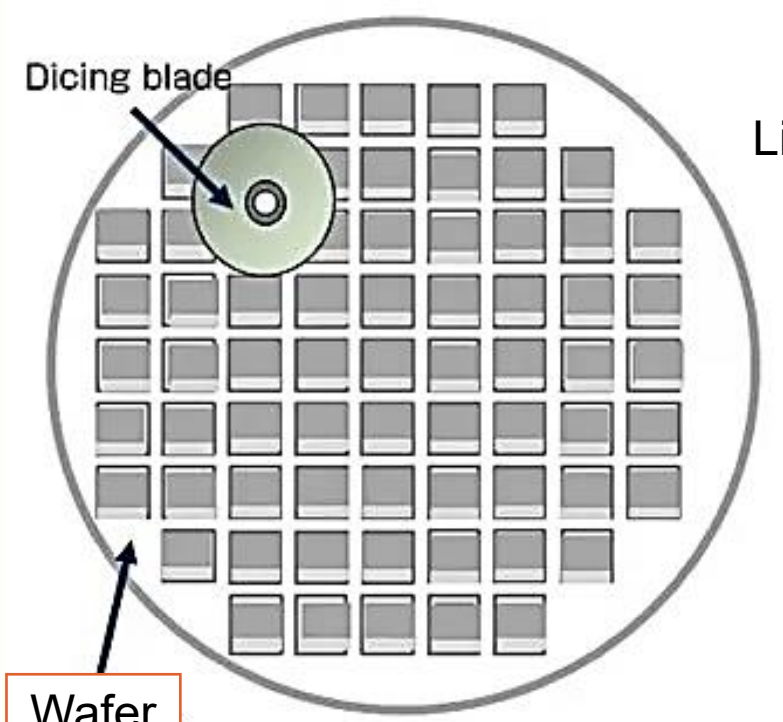
2D Materials and Novel Semiconductors: Transition metals like MoS_2 or graphene replace silicon for channels, enabling sub-1nm scaling with lower leakage (research at IMEC/TSMC). This extends efficiency for low-power MCUs.

Advanced Packaging and Co-Design: Techniques like silicon interposers (TSMC InFO) and co-optimization of hardware/software (e.g., AI-specific accelerators) deliver "effective" scaling. Intel's "Tick-Tock" evolved into "Process-Architecture-Optimization" to integrate these.

Domain-Specific Architectures: Specialized chips (e.g., Google's TPUs, Arm's Ethos NPUs) offload tasks, achieving Moore-like gains through efficiency rather than raw transistor count.

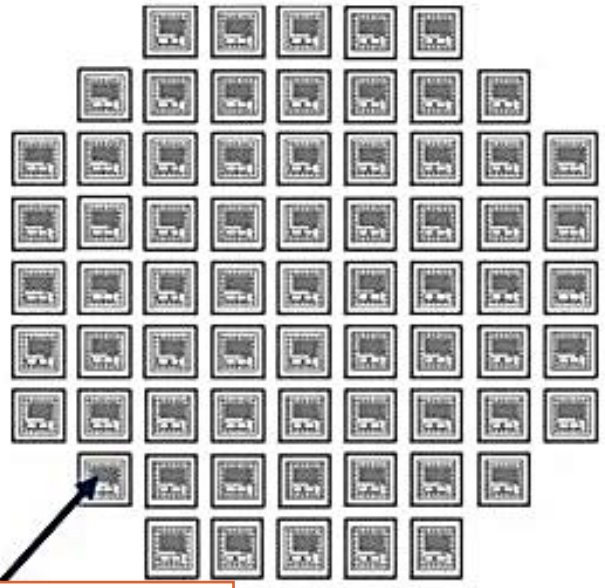
Now that we have got a glimpse of the journey of computing evolution, do we know how ICs are manufactured?

So, what steps are involved to create a new MCU?



200nm (8-inch) Wafer /
300mm (12-inch) Wafers

Lithographic process



Die for the IC

Smaller the area per DIE, more the ICs that can be obtained

Packed Die
as ready to use IC

So, what steps are involved to create a new MCU? (1)

1. Product Definition / Market Requirements

- Target application (automotive, IoT, industrial)
- Performance requirements
- Cost targets
- Power consumption targets
- Reliability requirements

Output → **Product Requirement Document (PRD)**

2. Technology Node Selection (Process Node)

Decision on fabrication technology such as:

- **180 nm**
- **130 nm**
- **90 nm**
- **65 nm**
- **40 nm**
- **28 nm**

Example foundries:

- TSMC
- UMC
- GlobalFoundries

Typical MCU nodes today: **180 nm → 40 nm**

Reasons:

- Lower cost
- Higher reliability
- Embedded Flash compatibility

Output → **Process Design Kit (PDK)**

So, what steps are involved to create a new MCU?(2)

3. System Architecture Design

Define system-level architecture:

- CPU core (ARM / RISC-V)
- Memory architecture
- Peripheral set
- Bus structure (AHB/APB)

Output → **MCU Block Diagram**

4. IP Selection / Development

Many parts of the chip are **licensed IP blocks**.

Examples:

- CPU core
- USB controller
- ADC
- SRAM macros
- Flash macros

Vendors include:

- Arm Ltd.
- Synopsys
- Cadence Design Systems

5. RTL Design

Hardware coded using:

- Verilog
- VHDL
- SystemVerilog

Design modules:

- CPU integration
- Bus interconnect
- Peripherals
- Memory controllers

Output → **RTL code**

So, what steps are involved to create a new MCU? (3)

6. Functional Verification

Ensure logical correctness.

Methods:

- Simulation
- Test benches
- Assertions
- Coverage analysis

Output → **Verified RTL**

7. Logic Synthesis

Convert RTL → **gate-level netlist**

Mapping to:

- Standard cells
- Flip-flops
- Logic gates

Output → **Netlist**

8. Design for Test (DFT)

Add hardware for manufacturing test:

- Scan chains
- Built-in self-test (BIST)
- Boundary scan (JTAG)

Purpose:

Ensure chips can be **tested after fabrication.**

So, what steps are involved to create a new MCU? (4)

9. Physical Design (Chip Layout)

Transform logic into physical layout.

Steps:

1. Floorplanning
2. Power grid design
3. Placement
4. Clock tree synthesis
5. Routing

Output → **GDSII layout**

10. Sign-off / Physical Verification

Final checks before manufacturing:

- DRC (Design Rule Check)
- LVS (Layout vs Schematic)
- Timing analysis
- Power analysis
- Signal integrity

11. Tape-Out

Final design files sent to foundry.

Output → **Photomasks**

So, what steps are involved to create a new MCU? (5)

12. Wafer Fabrication

Manufacturing using:

- Photolithography
- Ion implantation
- Etching
- Metal deposition

Output → **Silicon wafers**

13. Wafer Probe Testing

Test each die on wafer.

Bad dies are discarded.

14. Packaging

Convert die → usable IC.

Examples:

- QFN
- QFP
- BGA

15. Final Test and Qualification

Electrical tests:

- Functional test
- Speed test
- Power test
- Reliability testing

16. Productization

Final MCU released with:

- Datasheet
- Development boards
- SDK / compilers
- Reference designs

Example MCU families:

- STM32 microcontrollers
- PIC microcontrollers
- AVR microcontrollers

Summary of steps involved in creating a new MCU