Processor Evolution

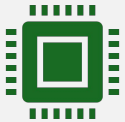
By: Bilal Ahmed

CISC vs RISC

Feature	**CISC (Complex Instruction Set Computer)**	**RISC (Reduced Instruction Set Computer)**
Instruction Set	Large, complex instructions (multi-step in one)	Small, simple instructions (one step each)
Instruction Length	Variable (may be short or long)	Fixed (usually 32 bits)
Execution Time	Multiple clock cycles per instruction	Typically one clock cycle per instruction
Hardware Complexity	More complex control unit, often microprogrammed	Simpler control unit, hardwired
Code Density	Higher (fewer lines of assembly code)	Lower (more instructions needed)
Performance	Optimized for fewer instructions but slower execution per instruction	Optimized for faster execution per instruction
Power Consumption	Higher	Lower
Typical Use	Desktops, laptops, legacy systems (x86)	Mobile devices, embedded systems (ARM, MIPS, RISC-V)

CISC Implementations

x86 / x86-64 (Intel, AMD) → PCs, laptops, servers



VAX (Digital Equipment Corp.) → Minicomputers in the 1980s

System/360 and System/370 (IBM mainframes) → Large enterprise systems

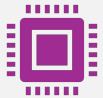
Z/Architecture (IBM) → Modern mainframes

RISC Implementations

ARM → Smartphones, tablets, IoT devices, embedded systems

MIPS → Routers, older game consoles (PS1, PS2, Nintendo 64)

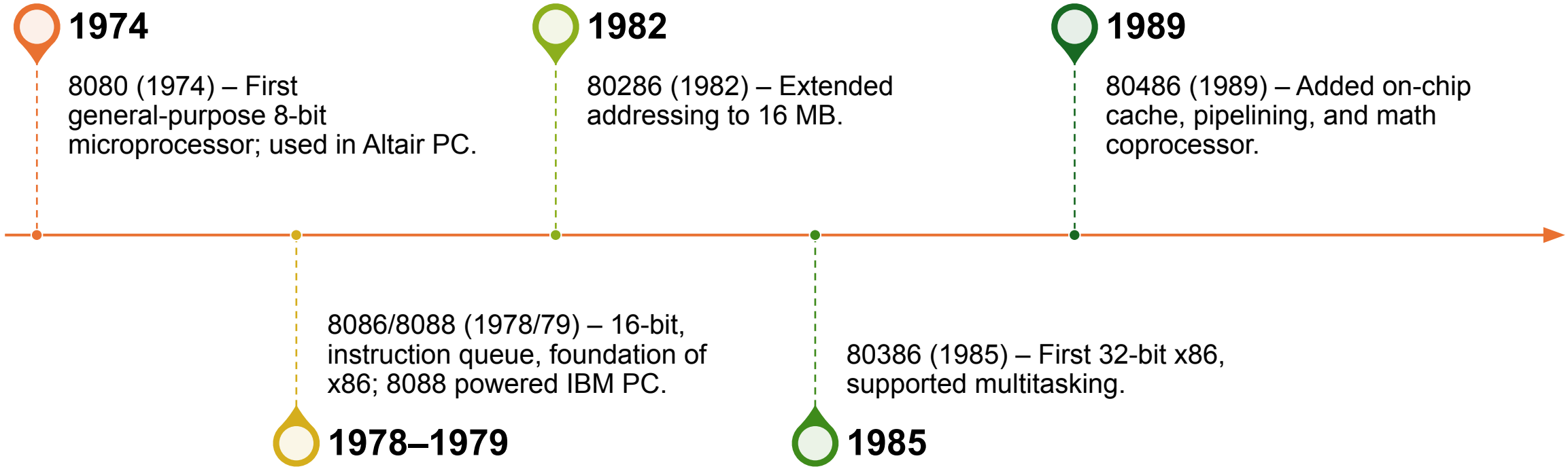
SPARC (Sun Microsystems) → Workstations, servers



PowerPC (IBM/Motorola/Apple) → Older Macs, game consoles (Xbox 360, PS3)

RISC-V → Open-source ISA, used in research, IoT, and emerging processors

Intel x86 Evolution



Intel x86 Evolution

Pentium (1993) – Superscalar: multiple instructions executed in parallel.

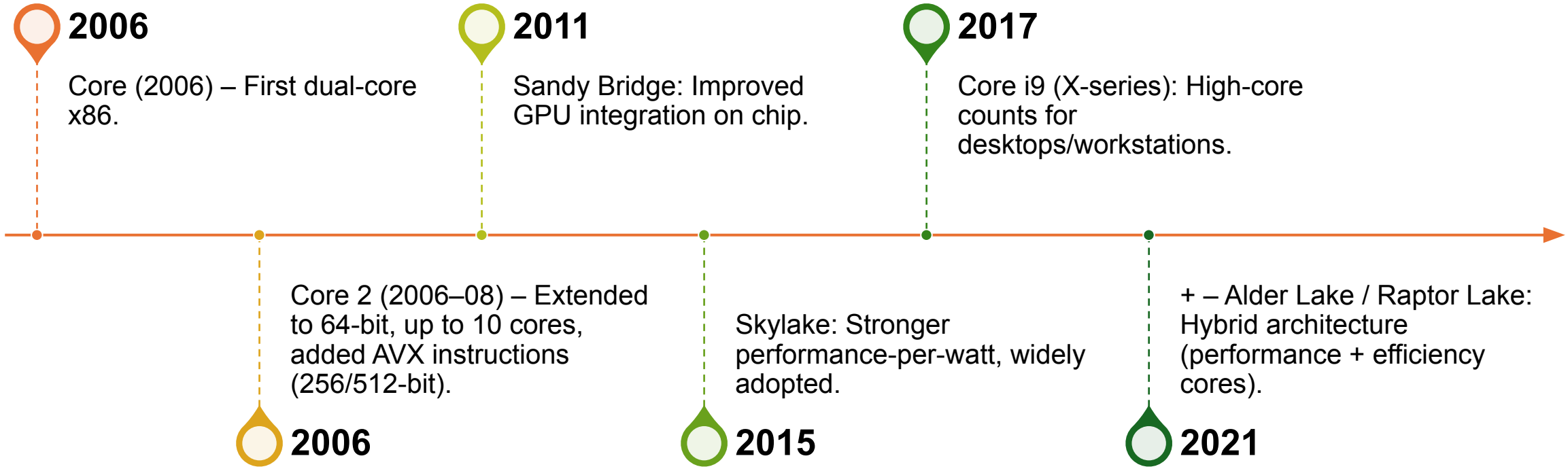
Pentium Pro (1995) – Advanced branch prediction, register renaming, speculative execution.

Pentium II (1997) – Added MMX for multimedia.

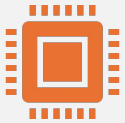
Pentium III (1999) – Added SSE instructions for vector/matrix operations.

Pentium 4 (2000) – Enhanced multimedia performance.

Intel x86 Evolution



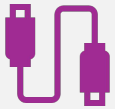
Embedded Systems



Embedded systems: Electronics + software integrated within a product (not general-purpose computers).

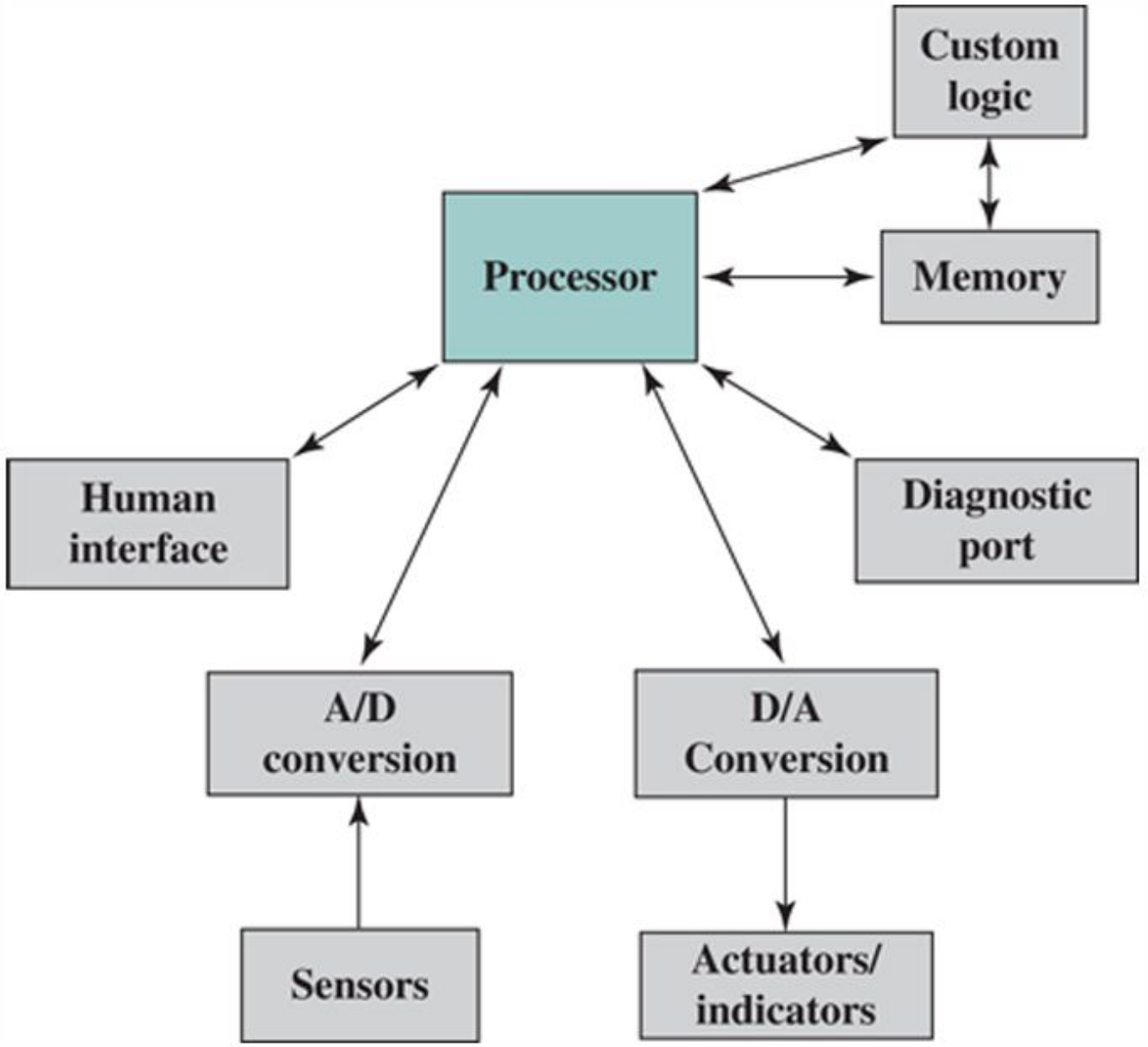
Millions of general-purpose computers sold yearly.

Billions of embedded systems produced yearly.



Trend: Most electrically powered devices today use embedded systems; in the future, nearly all will.

Embedded System Organization



Embedded System Organization

- **Interfaces**: Allow measurement, control, and interaction with the environment.
- **Sensors & Actuators**: Enable sensing, manipulation, and communication → embedded systems are usually **reactive** (work in real-time with environment).
- **Human Interface**: Can be minimal (flashing light) or complex (robotic vision); sometimes none.
- **Diagnostic Port**: Used to check both the computer and the controlled system.
- **Hardware**: May use FPGA, ASIC, or other special hardware for higher performance/reliability.
- **Software**: Fixed-function, application-specific.
- **Optimization Goals**: Energy efficiency, small code size, fast execution, reduced weight/size, and low cost.

Internet of Things

- **IoT = interconnection of smart devices** (sensors, appliances, gadgets) via wireless + cloud.
- Devices **sense, act, communicate, self-modify** → enable larger systems (e.g., smart cities).
- Mostly **deeply embedded, low-bandwidth devices**; few need high-bandwidth (e.g., cameras).
- **IoT evolution:**
  1. IT (PCs, servers, wired)
  2. OT (machines with embedded IT, wired)
  3. Personal tech (smartphones, wireless)
  4. IoT (billions of sensors/actuators, wireless).

ARM Architecture

- The **ARM architecture** is built on **RISC (Reduced Instruction Set Computer) design principles**, meaning it uses a **small set of simple instructions** that execute very efficiently.
- Instead of manufacturing chips itself, **ARM Holdings (Cambridge, UK)** only **designs processors and processor architectures** and then **licenses** them to other companies.
- **Processor License** → A company uses ARM's ready-made processor design in its chip.
- **Architecture License** → A company creates its own chip design, but it must follow ARM's architecture rules.
- Because of this business model, ARM technology is used by hundreds of companies worldwide.

Features of ARM Chips

- **High performance** while maintaining **low power consumption** → perfect for **portable and battery-powered devices**.
- **Small die size** → saves chip area and cost.
- **Energy efficiency** → major reason ARM dominates smartphones and IoT devices.
- Found in:
  - **Apple's iPod and iPhone**
  - **Almost all Android smartphones**
  - **Game consoles & handhelds**
  - Many **consumer electronics** and **embedded systems**.
- By **2016**, ARM's partners shipped **16.7 billion ARM-based chips**, making it the **most widely used processor architecture in the world**.

ARM Architecture Evolution

- **Early 1980s – Acorn Computers (UK):**
  - Acorn got a contract from the **BBC** for the *BBC Computer Literacy Project*.
  - This led to the development of **Acorn RISC Machine (ARM)** processors.
- **1985 – ARM1:**
  - The **first commercial RISC processor (ARM1)** became operational.
  - Used for **research** and as a **coprocessor** in BBC machines.
- **VLSI Technology partnership:**
  - Acorn partnered with **VLSI Technology** for **fabrication** of ARM chips.
  - VLSI also marketed ARM chips to other companies → ARM started being used in **embedded systems**.

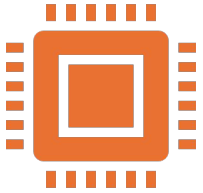
ARM Architecture Evolution

- **Need for wider development:**
  - ARM's design proved ideal: **low power, small, cheap, efficient** → perfect for embedded applications.
  - But Acorn alone couldn't take it further.
- **Formation of ARM Ltd. (1990):**
  - A new joint venture was created with **Acorn, VLSI, and Apple** as partners.
  - **Acorn RISC Machine** was renamed to **Advanced RISC Machines**.
- **2016 – Acquisition:**
  - ARM Holdings was acquired by **SoftBank Group (Japan)** for about $32 billion.

ARM Instruction Set Architecture (ISA)

- The **ARM instruction set** is highly regular, designed for **efficient processor implementation** and **fast instruction execution**
- All instructions are **32 bits long** and follow a **consistent, regular format**, which simplifies both hardware design and software development.
- This regularity makes the **ARM ISA** well-suited for implementation across a wide range of products, from **embedded systems** to **high-performance computing devices**.

ARM Products



Cortex-A:

Application processors for mobile and consumer devices.

High clock speeds (>1 GHz).

Includes MMU (supports full OS like Linux, Android, Windows).

Available in 32-bit and 64-bit versions.

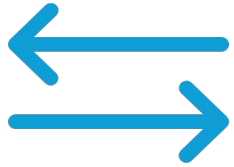
Cortex-R:

Designed for real-time systems needing fast response.

Runs at several MHz with very low latency.

No MMU, but has MPU, cache, and memory protection features.

Used in automotive braking, storage controllers, networking, printers.



Cortex-M:

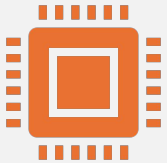
For microcontrollers with low power and small size.

Fast, predictable interrupt handling.

Has MPU but no MMU.

Common in IoT devices, wireless sensor networks, automotive electronics.

ARM-based Boards



STM32 Nucleo boards (STMicroelectronics) – widely used for learning ARM Cortex-M microcontrollers.

TI LaunchPad series – based on MSP430 (low power) and ARM Cortex-M.

mbed boards (by ARM/partners) – designed for rapid prototyping.

General Purpose & Hobby Boards

Arduino boards – (UNO, Mega, Nano, Due) – very popular for beginners, AVR/ARM-based.

Raspberry Pi – (Pi 4, Pi Zero, Pico) – Linux-capable SBCs (ARM Cortex-A) and microcontrollers (Cortex-M0+ in Pico).

BeagleBone Black – ARM Cortex-A8, used for Linux-based embedded education.

FPGA + Embedded Education + Low-Cost Learning Kits

Xilinx Zynq boards (e.g., ZedBoard, PYNQ) – FPGA + ARM Cortex-A cores.

Intel/Altera DE series (DE0, DE10) – FPGAs with soft-core CPUs for embedded teaching.

ESP32 / ESP8266 boards – WiFi + Bluetooth microcontrollers, very popular in IoT education.

BBC micro:bit – beginner-friendly ARM Cortex-M0 board for kids and schools.