Session-II
CPU Architecture

Session outline

- CPU architectures
- RISC Vs CISC
- CPU in context of today’s applications
- Multicores in context of Automotive
- Architectural approaches in multicore

CPU architecture

- Approaches to multiply numbers present at two memory locations e.g. a01 and a22 and store the result in memory location a22

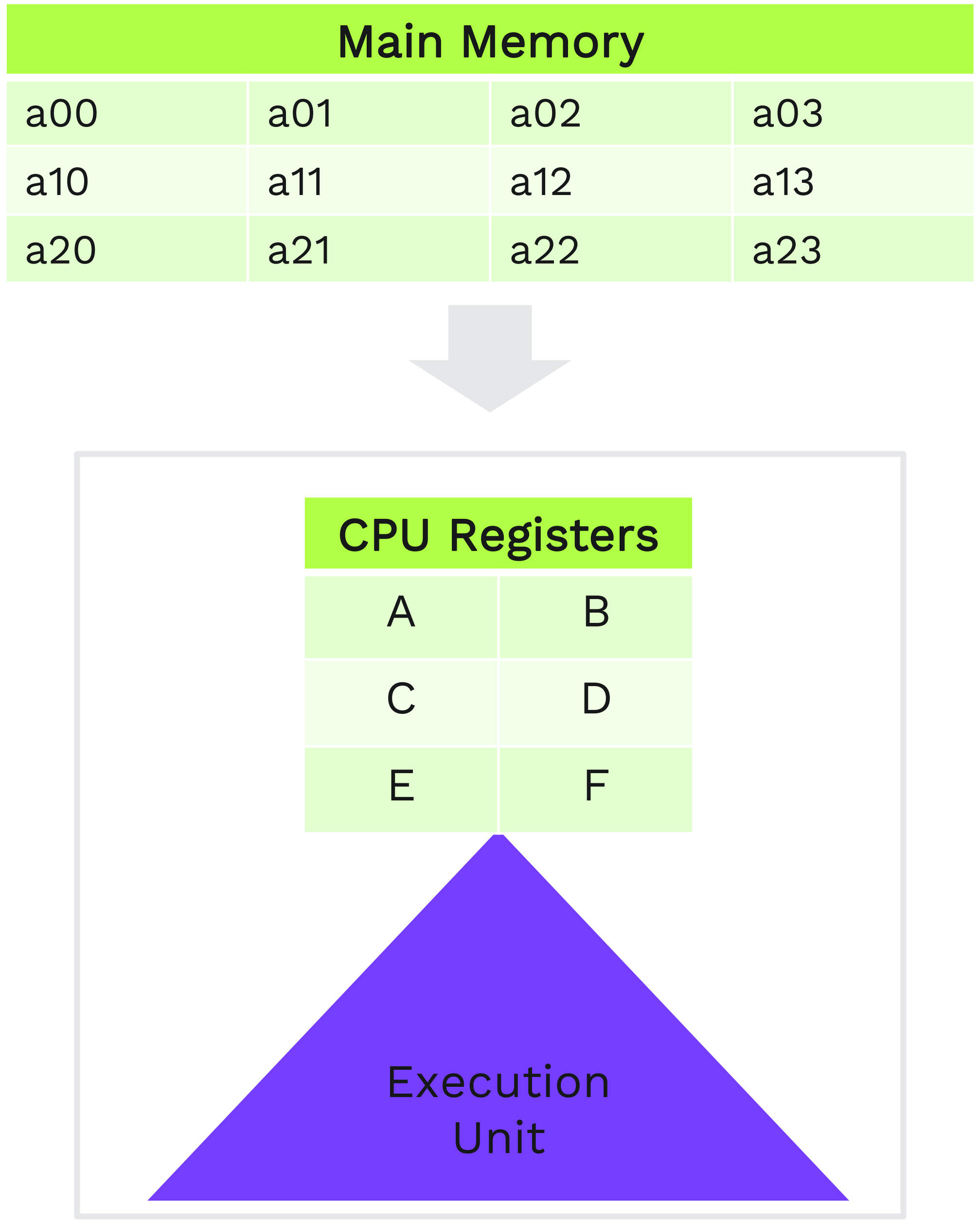
Approach 1:

```
LOAD A, a01
LOAD B, a22
PROD A, B
STORE a22, A
```

Approach 2:

```
MUL a22, a01
```

- More RAM is needed to store the assembly level instructions in approach 1 as compared to approach 2
- Approach 2 attempts to minimize the number of instructions per program, sacrificing the number of cycles per instruction.
- Approach 2 requires more complex hardware as compared to approach 1



Differentiating CISC Vs. RISC

CISC Design approach	RISC Design approach
Emphasis on hardware	Emphasis on software
Includes multi-clock complex instructions	Single-clock, reduced instruction only
Memory-to-memory: "LOAD" and "STORE“ incorporated in single instructions	Register to register: "LOAD" and "STORE“ are independent instructions
Small code sizes, high cycles per second	Low cycles per second, large code sizes
Transistors used for storing complex instructions	Spends more transistors on memory registers

KPIT

Key trends in CPU architecture

Performance:

- The increase in performance is roughly proportional to the square root of the increase in complexity.

That is: – 2x the core logic means a √2 ~= 1.4x performance increase

Power Dissipation:

- Power dissipation: P = CxVxVXf

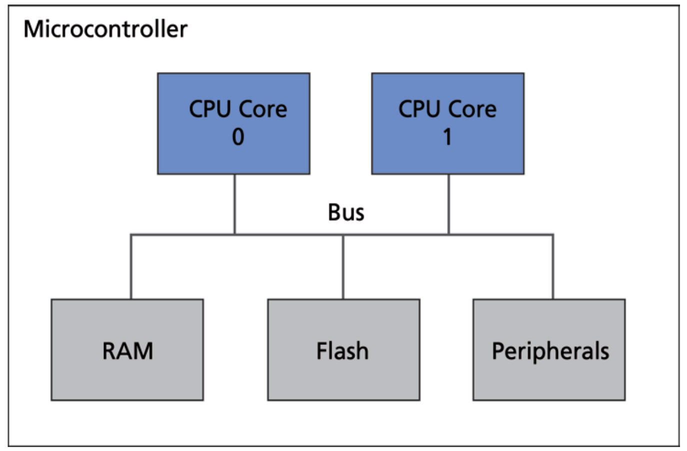
To reduce power dissipation, reduce clock speed and add more CPU cores.

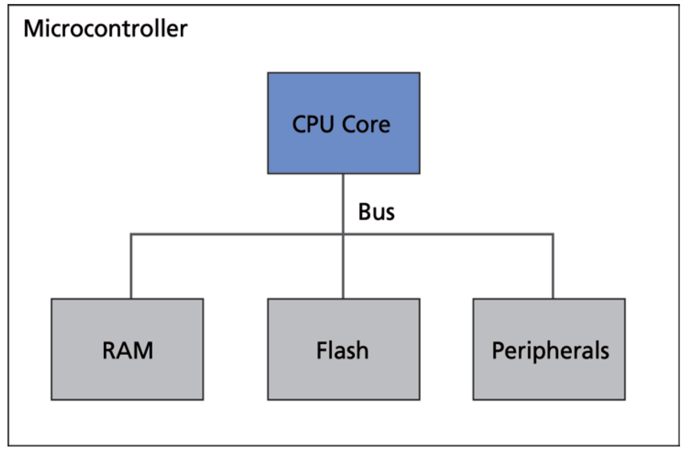
CPU in context of today’s application

Typical application requirements are

- Enhanced throughput or increased computational power
- Low Latency
- Better power management
- Safety

Scalable → Adaptive → Power Efficient → Reliability

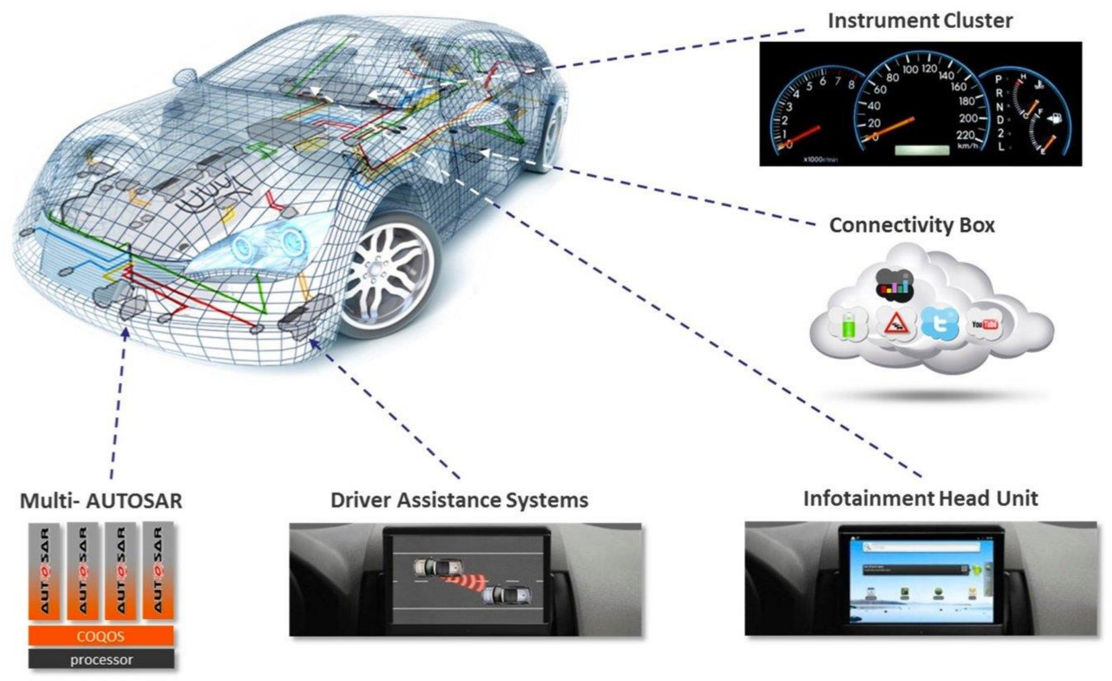




Main drivers in automotive

There are two main drivers:

- Performance and
- Safety



New driver assistance systems, digital instrument clusters, and head units with infotainment all require advanced computing power inside the vehicle

- Multicore architecture increases the throughput and performance of Automotive applications
- Failures happening in software can be better detected and handled in multi-core systems

Architectural approaches in multicore

Multi-core systems may implement architectures such as

- VLIW,
- superscalar,
- vector, or
- Multithreading

Very-Long Instruction Word (VLIW) architectures

VLIW has been described as *a natural successor to RISC,* because it moves complexity from the hardware to the compiler, allowing simpler, faster processors.

Goal of the hardware design:

- reduce hardware complexity
- to shorten the cycle time for better performance
- to reduce power requirements

KPIT

Relevance of using multicore in Automotive

Three main points to be taken into consideration for multicore architectures to be used efficiently in the future in the automotive sector:

A. Critical and Non- critical tasks must run parallel without affecting each other

B. Multi OS support; the multicore system needs to be able to run multiple operating systems at the same time.
For example, AUTOSAR (AUTomotive Open System ARchitecture) for safety-critical functions, GenIVI (based upon Linux) for automotive infotainment, Android for user apps

C. Efficient shared use of processor resource