॥ अप्पो दीपो भव ॥
महात्मा गांधी मिशन विश्वविद्यालय
MGMUNIVERSITY

Jawaharlal Nehru Engineering College

AVR Microcontroller

DEMP: CA-4

Prof. Deepali Naglot ma’am

Presented by:

1)	Abhishek Chavan	202301103006
2)	Vedant Waghchaure	202301103007

AVR History

- AVR was developed in the year 1996 by **Atmel Corporation.**
- The architecture of AVR was developed by Alf-Egil Bogen and Vegard Wollan.
- AVR derives its name from its developers and stands for **A**lf-Egil Bogen **V**egard Wollan **R**ISC microcontroller.
- Also known as Advanced Virtual RISC.
- The first microcontroller to hit the commercial market was **AT90S1200** in the year 1997.

Categories of AVR

1. **TinyAVR**

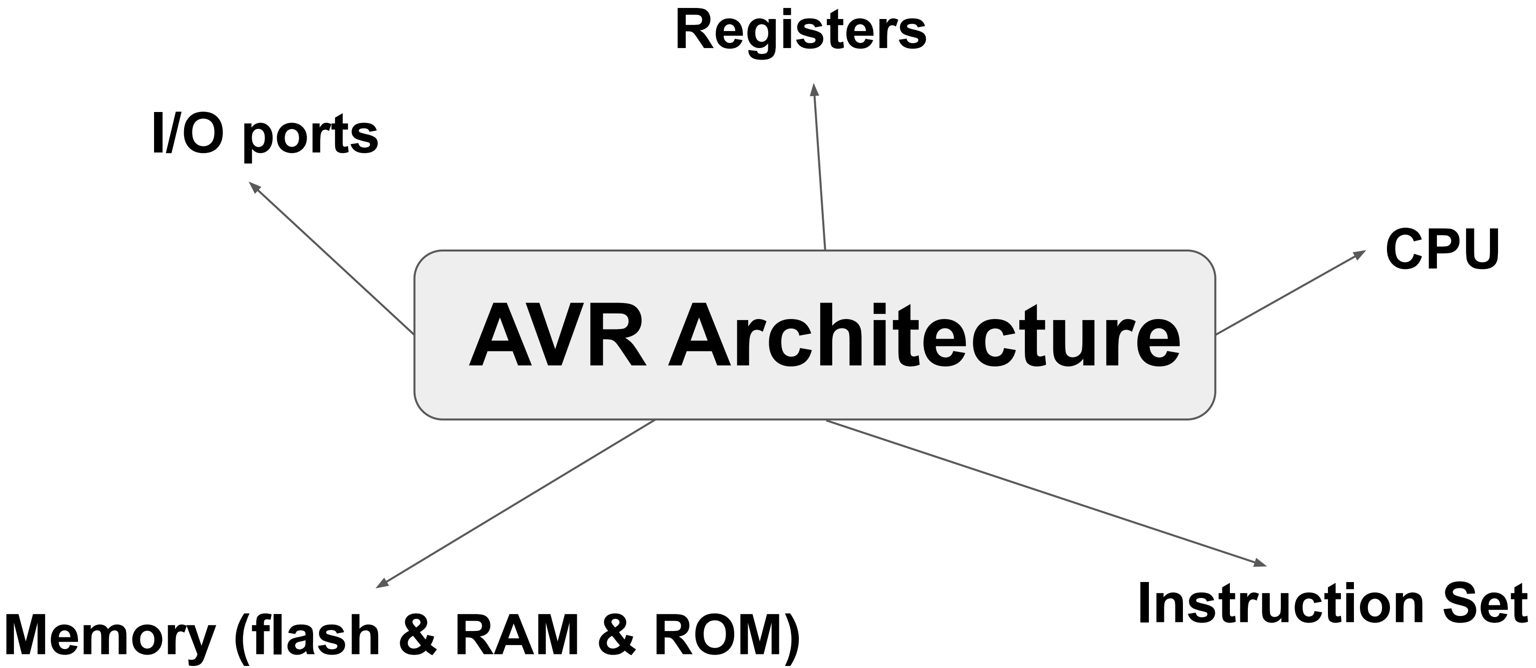
- Less memory.
- Small size.
- Suitable only for simpler applications.

2. **MegaAVR**

- These are the most popular ones.
- Having good amount of memory (up to 256 KB) higher number of inbuilt peripherals.
- Suitable for moderate to complex applications.

3. **XmegaAVR**

- Used commercially for complex applications, which require large program memory and high speed.



AVR Architecture

Registers

I/O ports

CPU

Memory (flash & RAM & ROM)

Instruction Set

- Types of registers

1. **General purpose registers**: Can store both data and addresses.
2. **Special purpose registers:** program counter, stack pointer, status register.

- Memory

1. Program Memory

Program memory holds ***interrupt function addresses***, ***16 bit*** and double word (32 bit) ***opcode***, and ***static data tables.***

2. Data memory

Used for data storage and different from program memory, which has ***128 bytes of SRAM.***

AVR Instruction SET

118 Powerful Instructions - Most Single Clock Cycle Execution

All arithmetic operations are done on registers RO - R31 Mostly instructions take one cycle for execution

Example:
ADD Rd, Rr

Rd: Destination (and source) register in the Register File
Rr: Source register in the Register File

I/O Pins

General purpose I/O pins:

- Ports are simply gates through which CPU connects with the outer world.
- Each port has 3 control registers associated with it.
    - ***DDRx, PORTx, PINx.***

ADVANTAGES

1) Ease of Use: AVR microcontrollers are known for their simplicity and user-friendly architecture, making them accessible for beginners.

2) Wide Range of Options: The AVR family includes a variety of models with different features, memory sizes, and processing speeds, allowing for flexibility in project design.

3) In-System Programming (ISP): Many AVR microcontrollers support ISP, enabling developers to program and reprogram the chip while it is still mounted in the circuit.

4) Rich Development Tools: AVR offers robust development environments, such as Atmel Studio and Arduino IDE, along with extensive libraries and community support.

5) Low Power Consumption: AVR microcontrollers are designed for low power usage, making them suitable for battery-powered applications.

6) Built-in Peripherals: Many AVR models come with integrated peripherals like ADCs, timers, and communication interfaces (UART, SPI, I2C), reducing the need for additional components.

7) High Performance: AVR microcontrollers often provide good processing speed relative to their power consumption, making them efficient for various applications.

8) Flash Memory: The use of flash memory for program storage allows for easy updates and reprogramming without needing to replace the chip.

DISADVANTAGES

1) **Limited Processing Power**: Compared to more advanced microcontroller families (like ARM Cortex), AVR microcontrollers may have lower processing speeds and limited performance for complex applications.

2) **Limited Memory**: Some AVR models have restricted RAM and flash memory capacities, which can be a constraint for larger applications requiring more resources.

3) **Less Advanced Features**: Compared to newer microcontroller architectures, AVR may lack some modern features such as advanced debugging capabilities or more sophisticated peripheral options.

4) **Power Consumption at Higher Speeds**: Although AVR microcontrollers are generally low power, their power consumption can increase significantly at higher clock speeds.

5) **Older Architecture**: The AVR architecture is somewhat older, and while it’s reliable, it may not be as optimized as newer architectures for certain tasks.

6) **Limited Floating Point Support**: AVR microcontrollers generally do not have hardware floating-point units, which can slow down calculations involving floating-point arithmetic.

7) **Inefficiencies in Larger Applications**: As project complexity increases, managing and structuring code can become cumbersome compared to more modern platforms that support better abstraction and modularity.

8) **Toolchain Limitations**: While there are good development tools available, they may not be as advanced or feature-rich as those available for other architectures.

9) **Niche Applications**: For very specialized applications (like advanced robotics or high-performance computing), AVR might not be the best choice compared to alternatives.

10) **Community Fragmentation**: While there is a large community around AVR, it has become somewhat fragmented with the rise of alternative platforms, potentially leading to fewer recent resources and updates.

APPLICATIONS

1) Embedded Systems: AVR microcontrollers are widely used in embedded systems for tasks like controlling devices and processing data.

2) Consumer Electronics: They can be found in household appliances, remote controls, and other consumer gadgets for automation and control.

3) Automotive Applications: AVR microcontrollers are utilized in automotive systems for tasks like engine control, dashboard displays, and sensor integration.

4) Robotics: AVR is popular in hobbyist and educational robotics projects for controlling motors, sensors, and communication modules.

5) IoT Devices: Many IoT applications utilize AVR microcontrollers for connectivity, data processing, and device control in smart home devices and sensors.

6) Home Automation: Used in systems for controlling lighting, heating, security, and appliances remotely.

7) Educational Projects: Widely used in educational settings for teaching electronics, programming, and embedded system design.

8) Medical Devices: AVR microcontrollers can be found in medical devices for monitoring and control applications, such as blood pressure monitors and glucose meters.

Future Trends in AVR Technology

The future of AVR microcontrollers looks promising with advancements in technology. Innovations such as IoT integration and enhanced wireless communication capabilities are expected to expand their applications and improve overall functionality.