Thynk Loop

STM32 Fundamentals: Hands-on Workshop Series

Module 1

26th November 2024

Quick Intro…

Esakki Raja

Masters in Embedded Systems

Semiconductor, Medical, Industrial

Udhaydhithan Pa

Masters in Embedded Systems

Automotive, Medical, Industrial

Course Work

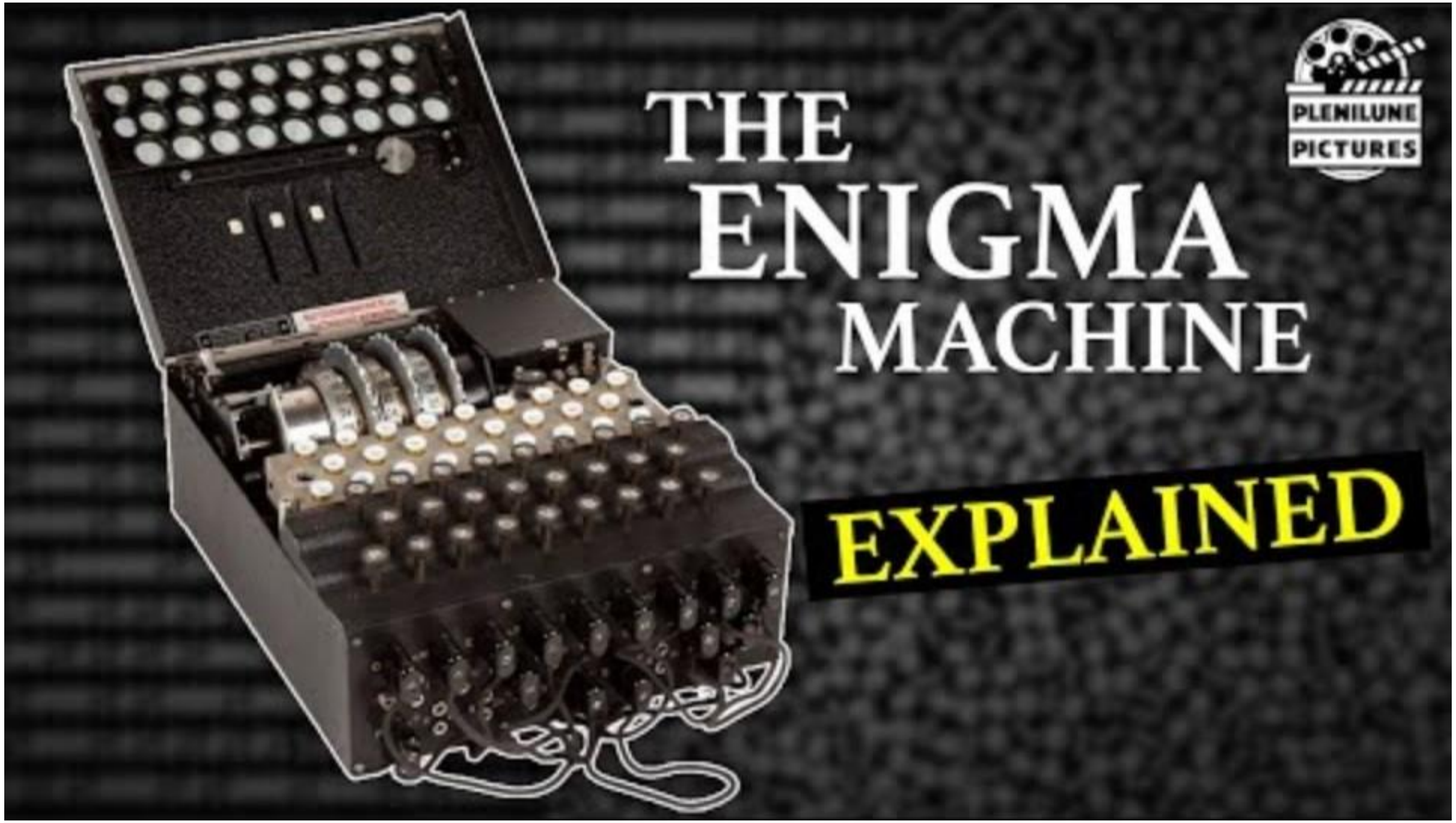
- Basics and ARM Cortex-M Architecture
- STM32 Peripherals Overview (GPIO, ADC, Timers, UART, SPI, I2C)
- Writing and Configuring GPIO Drivers
- ADC Driver Development (Single and Multi-Channel)
- Timer Driver Development (PWM, Delays, Interrupts)
- UART Driver Development (Communication and Debugging)
- SPI and I2C Driver Development (Communication Protocols)
- Debugging and Optimizing Embedded Drivers

Learning Outcome

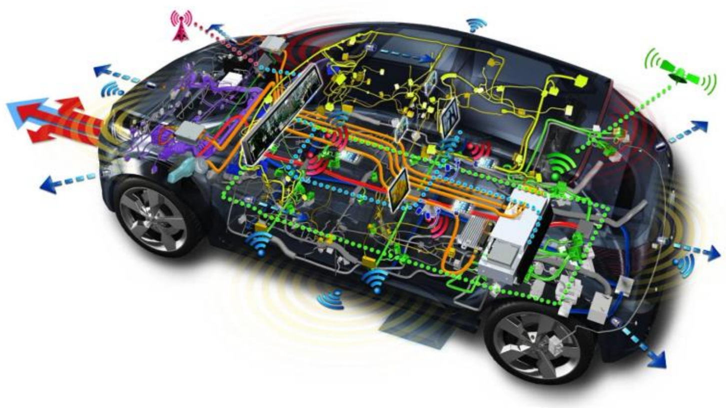
- Understand Microcontroller Fundamentals
- Master STM32 Architecture and Peripherals
- Develop Embedded Drivers
- Implement Advanced Debugging Techniques
- Use Development Tools Efficiently
- Apply Knowledge to Real-World Projects

Before We Start…

Enigma Machine – Birth Of Complex Machines

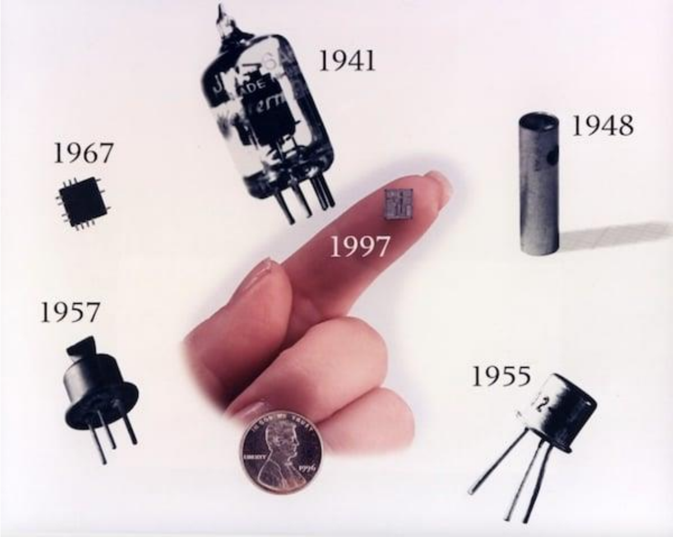


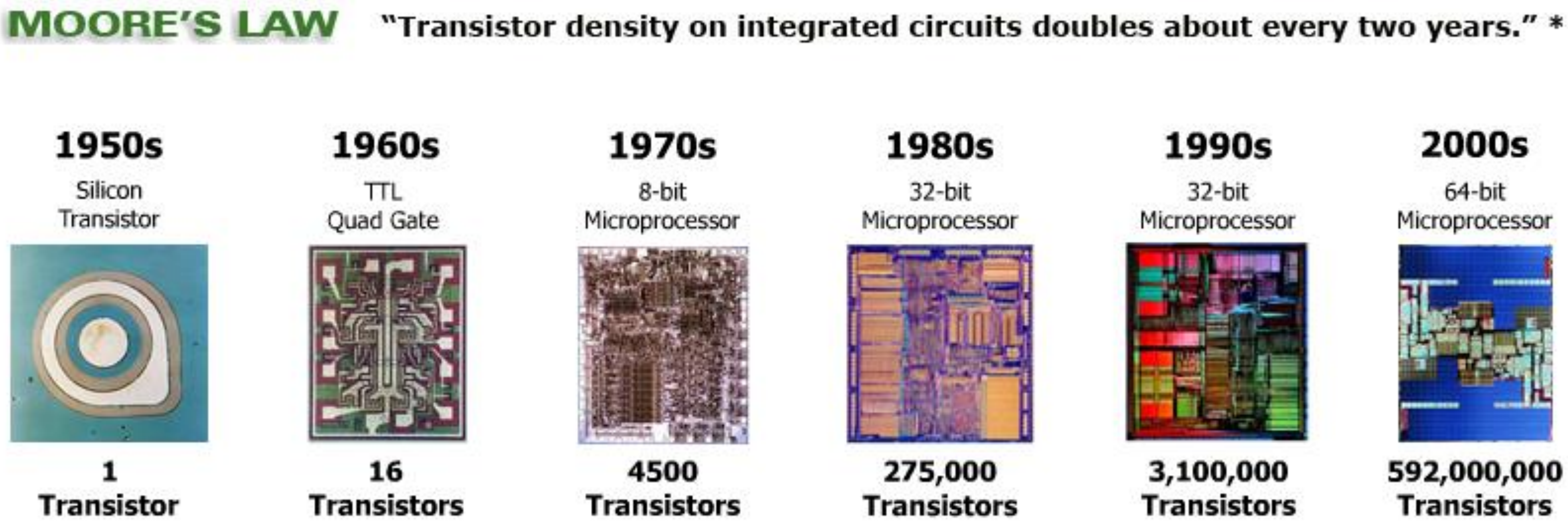
Where do we use embedded…



" Embedded systems are customized computing solutions crafted for specific tasks within larger devices, ensuring real-time operation and efficient execution of functions through dedicated hardware and software. "

The Evolution & Moore’s Law





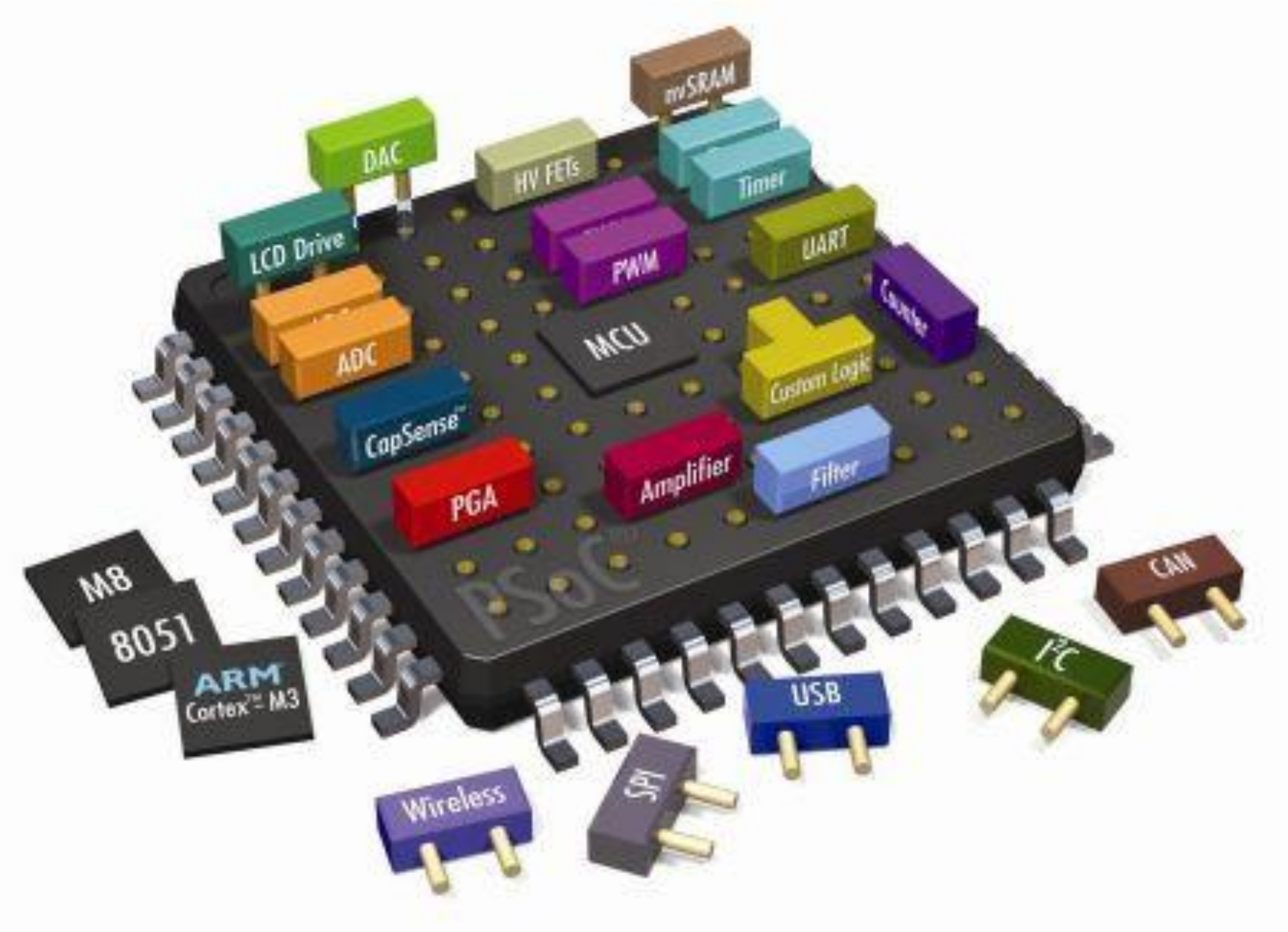
MOORE’S LAW **“Transistor density on integrated circuits doubles about every two years.” ***

1950s	1960s	1970s	1980s	1990s	2000s
Silicon Transistor	TTL Quad Gate	8-bit Microprocessor	32-bit Microprocessor	32-bit Microprocessor	64-bit Microprocessor
1 Transistor	**16 Transistors**	**4500 Transistors**	**275,000 Transistors**	**3,100,000 Transistors**	**592,000,000 Transistors**

Characteristics of Embedded Systems

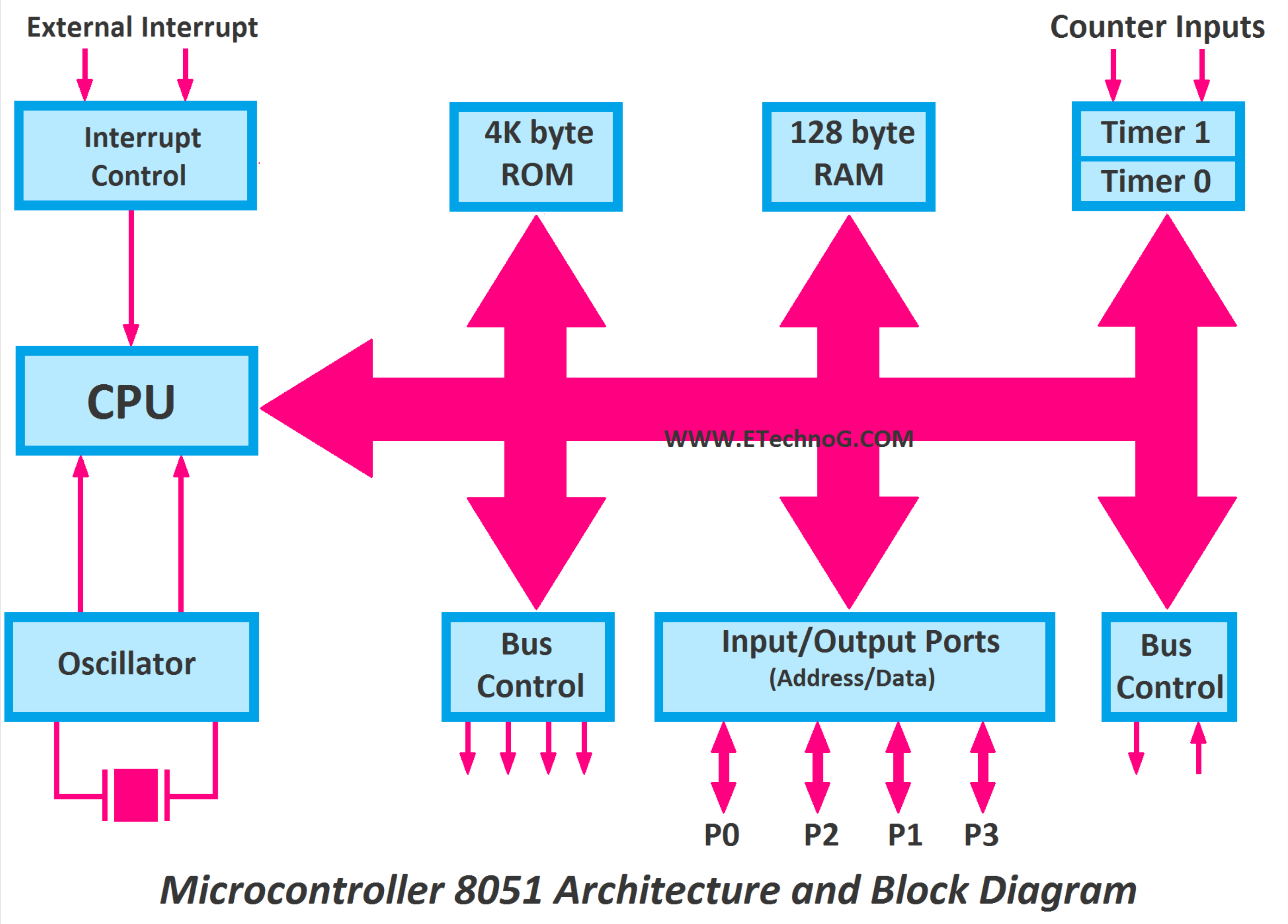
- Real-time Operation
- Memory Constraints
- Power Constraints
- Dependability and Reliability

Into Microcontroller

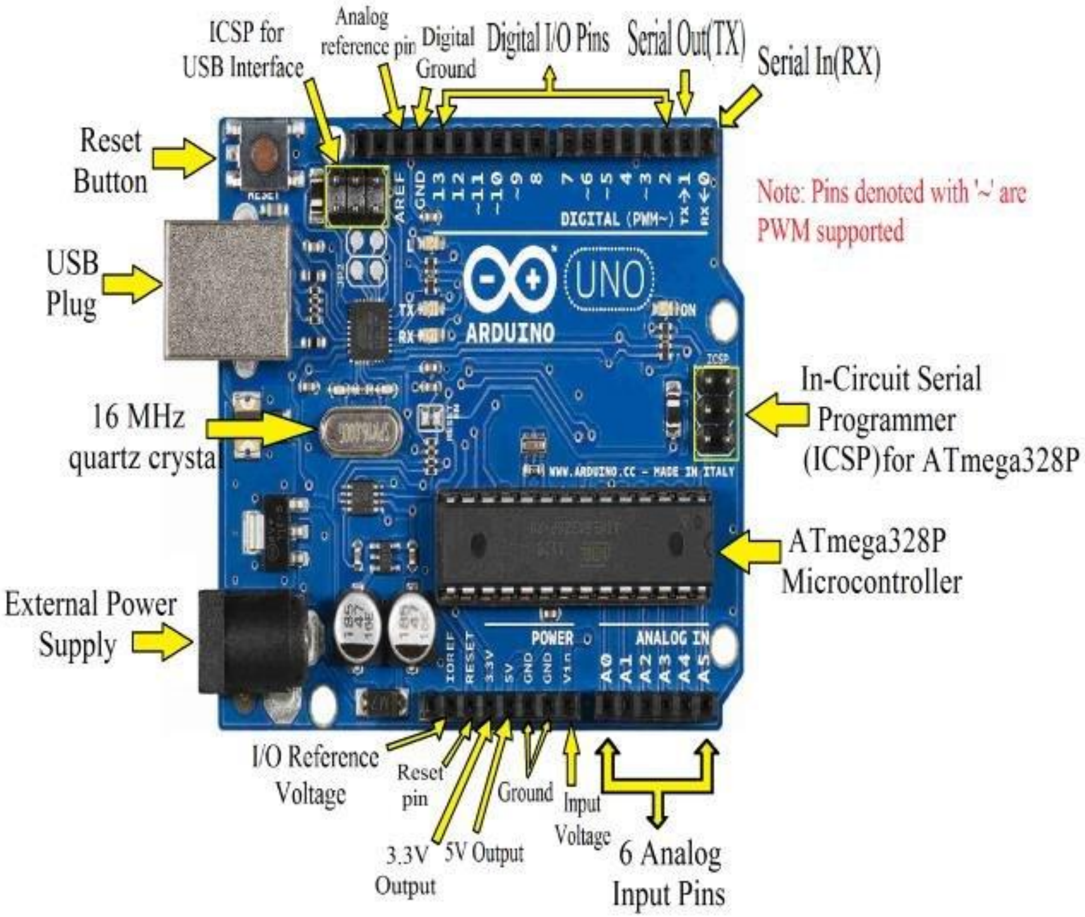


- Central Processing Unit (CPU)
- Memory (RAM, ROM, Flash)
- Input/Output Ports
- Peripherals (ADC, Timers, PWM)

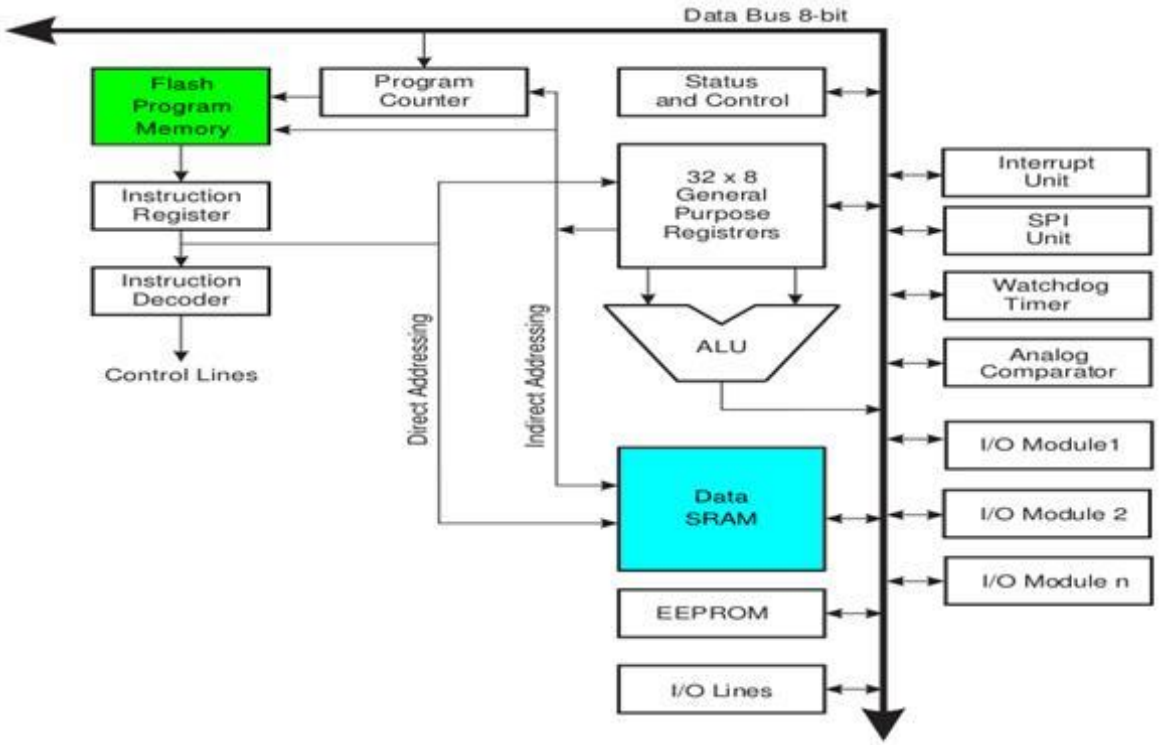
Old School – 8051 Architecture



ATMega 328P - Arduino



Note: Pins denoted with '~' are PWM supported



What is Cortex?

"Cortex" refers to a family of processors designed by ARM Holdings.

1. ***Cortex-A Series:*** Designed for high-performance applications, such as smartphones and tablets, with advanced features like out-of-order execution.

2. ***Cortex-R Series:*** Tailored for real-time systems, like automotive control and industrial applications, prioritizing predictability and reliability.

3. ***Cortex-M Series:*** Optimized for microcontroller and low-power embedded applications, commonly used in IoT devices, sensors, and other power-sensitive applications.

What is Cortex?

Cortex - A

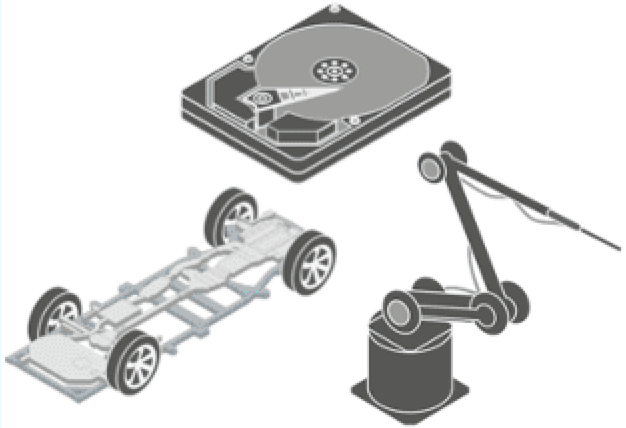
Highest performance

Optimised for
rich operating systems

Cortex - R

Fast response

Optimised for
high performance,
hard real-time applications

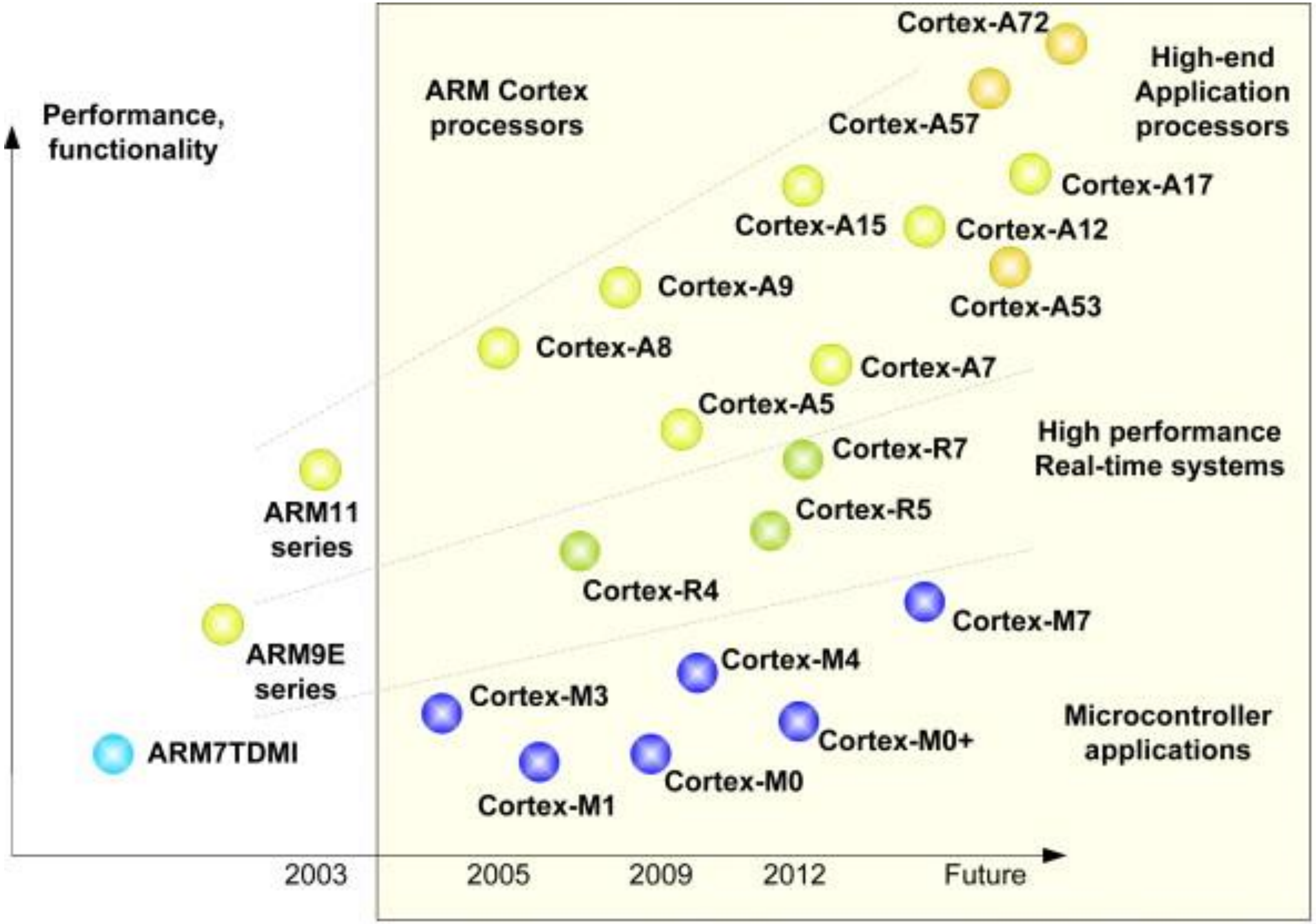


Cortex - M

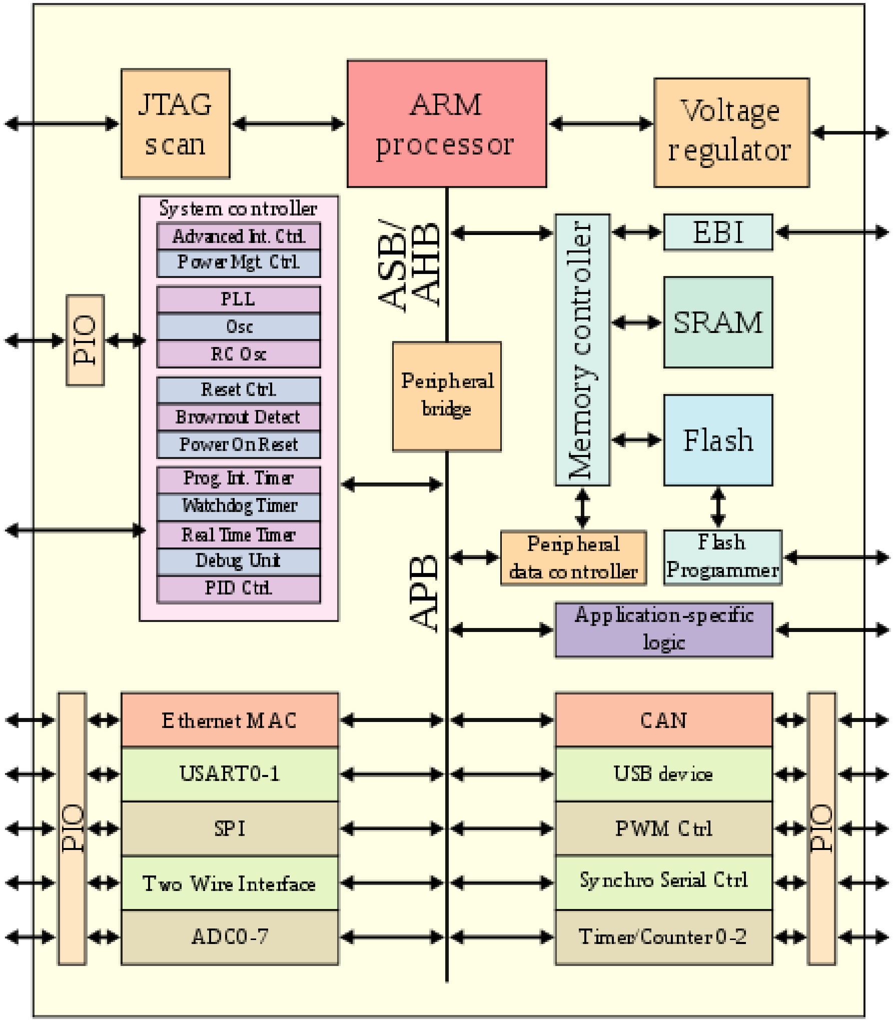
Smallest/lowest power

Optimised for
discrete processing and
microcontrollers

A Little Dive-in

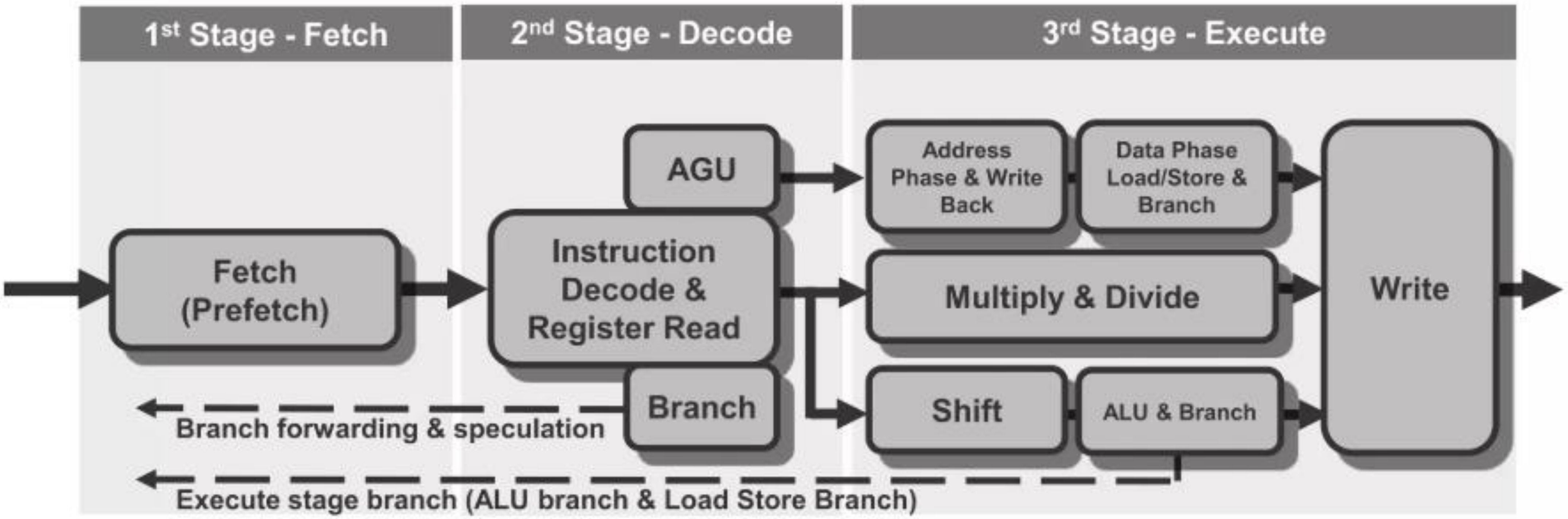


ARM Architecture

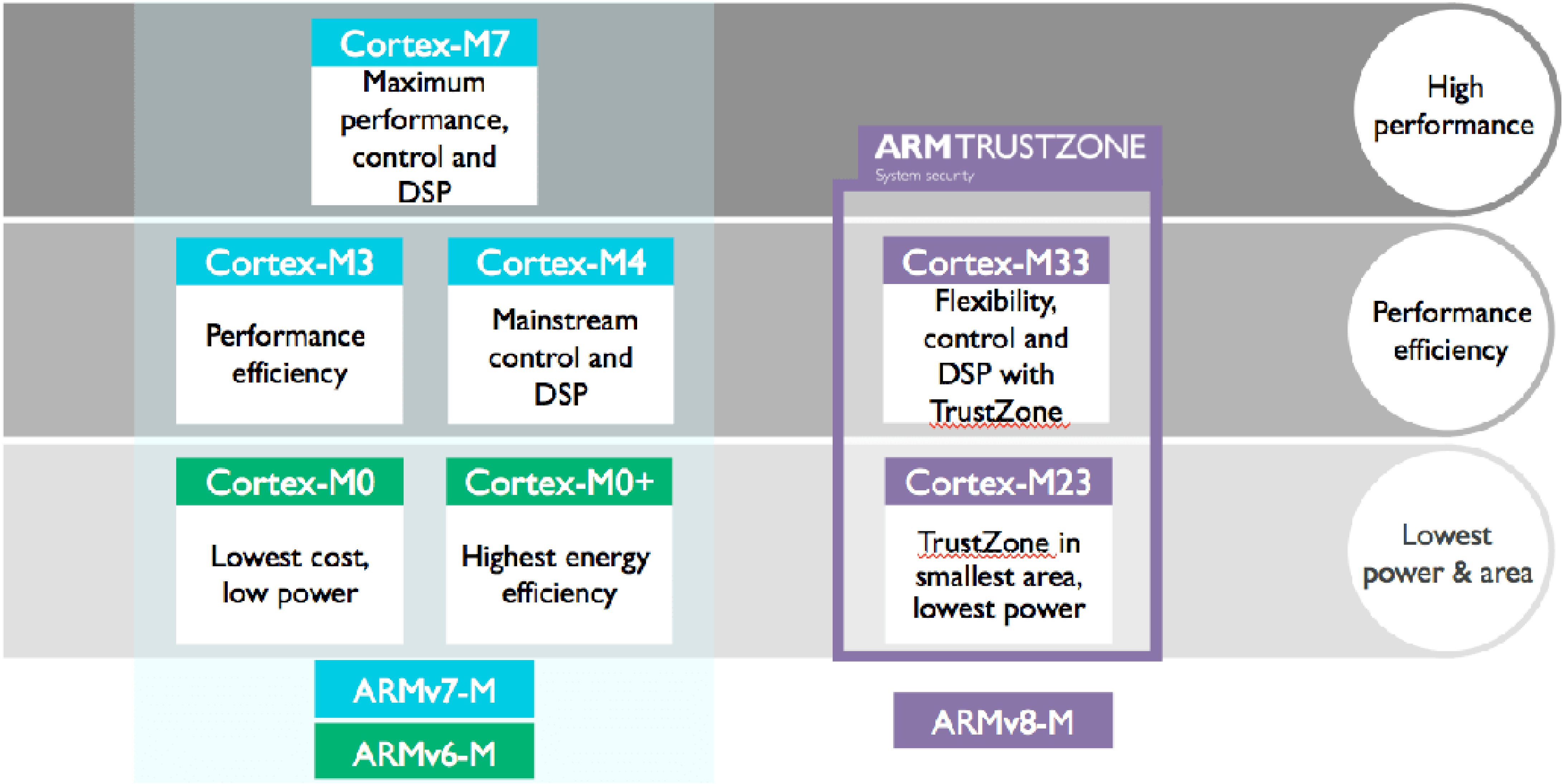


- Harvard Architecture:
- 32-bit RISC Architecture:
- Pipeline Architecture:
- Thumb and Thumb-2 Instruction Sets:
- Low Power Features:
- Interrupt Handling:
- Memory Protection Unit (MPU):
- Floating-Point Unit (FPU):
- Debugging and Trace Support:
- Peripheral Integration:

What is Pipeline


1st Stage - Fetch
2nd Stage - Decode
3rd Stage - Execute
Fetch (Prefetch)
AGU
Instruction Decode & Register Read
Branch
Address Phase & Write Back
Data Phase Load/Store & Branch
Multiply & Divide
Shift
ALU & Branch
Write
Branch forwarding & speculation
Execute stage branch (ALU branch & Load Store Branch)

Difference Between Architecture



ARM & It’s Memory

Flash Memory (Program Storage):

- Starting address: `0x0800 0000`
- Ending address: Varies based on the microcontroller model, determined by the Flash size.

SRAM (Data Storage):

- Starting address: It depends on the specific microcontroller model and its memory configuration.
- Ending address: Varies based on the microcontroller model, determined by the SRAM size.

System Memory (Bootloader Memory - Optional):

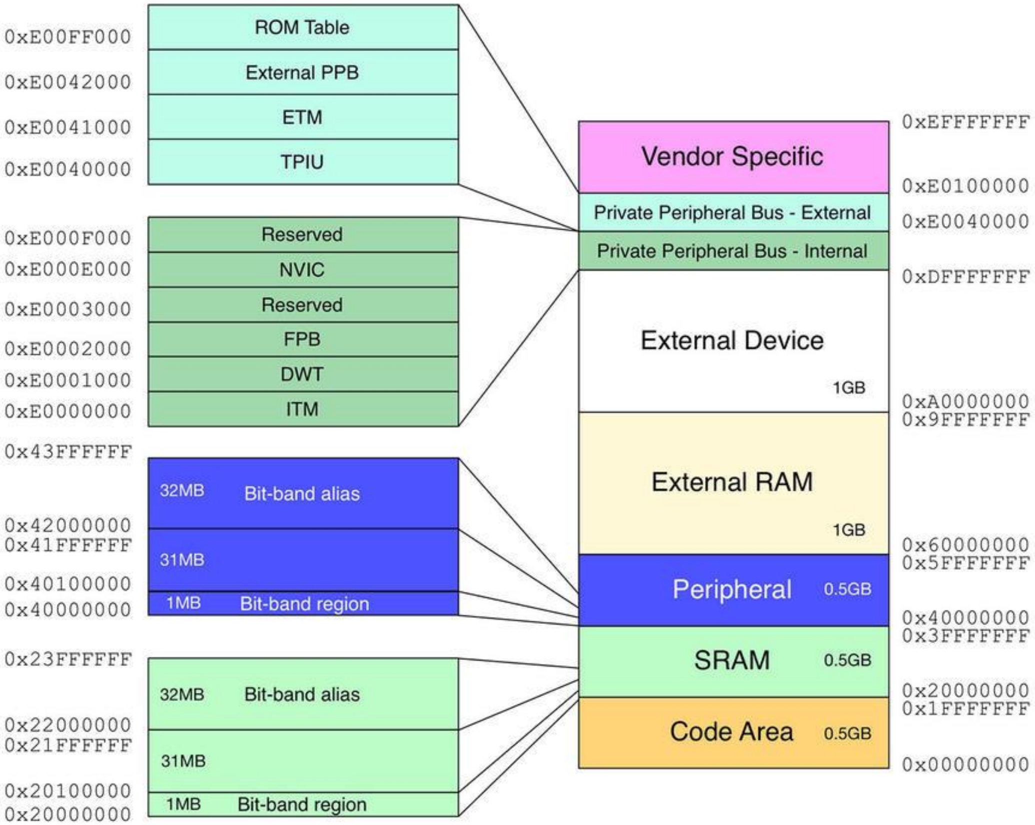
- Starting address: `0x1FFF 0000`
- Ending address: `0x1FFF FFFF`

Peripheral and Register Memory:

- Addresses vary based on the specific peripherals used in the microcontroller. For example, GPIO registers, USART registers, etc.

EEPROM (if available):

- Starting address: Varies; not all STM32 microcontrollers have EEPROM.
- Ending address: Varies based on the EEPROM size.



ARM Thumb 1

- **Purpose**: Reduces code size, ideal for memory-limited systems.
- **Instruction** Length: 16-bit instructions.
- **Key Points:**
  - More compact than ARM (32-bit instructions).
  - Fewer operations and registers.
  - Suitable for simple applications with limited functionality.

ARM Thumb 2

- **Purpose**: Combines the benefits of Thumb (16-bit) and ARM (32-bit).
- **Instruction Length**: Mix of 16-bit and 32-bit instructions.
- **Key Features**:
  - Supports both 16-bit and 32-bit instructions for flexibility.
  - **Improved Performance**: Allows complex operations with 32-bit instructions while maintaining compactness.
  - **Backward Compatible**: Works with older Thumb code.