

# Lecture 2 - ARM Controllers architecture

---

# Topics

---

- ARM series introduction
- ARM features

# ARM company history

---

- Architectural ideas that you see in ARM controllers really came in the 1980s
- Acorn had a computer called BBC which was based on an 8 bit microprocessor
  - MOS Technology 6502 CPU



## ARM company history

---

- For next iteration of BBC micro, Acorn tried to replace the 6502 with a more powerful controller based on smaller instruction set
  - Led to the first commercial RISC processor
  - ARM1 circa 1985
- Later version of BBC micro used the RISC based microprocessor

## ARM company history

---

- ARM company was founded in 1990
- Stands for
  - Advanced RISC Architecture
  - Initially own by Acorn and Apple

# Why ARM architecture?

---

- Why are we using ARM for teaching purposes?
- ARM dominates a lot of embedded system applications
  - Most popular category of microcontrollers

## Examples of ARM processors in products

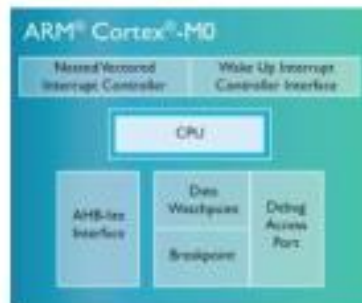
---

- Mainly used in battery operated devices
- ARM7 - Apple iPod
- ARM9 - Sony Erricson, BenQ TVs
- ARM11 - Original iPhone
- Generally, ARM processors are used when you need reasonably powerful computational capabilities

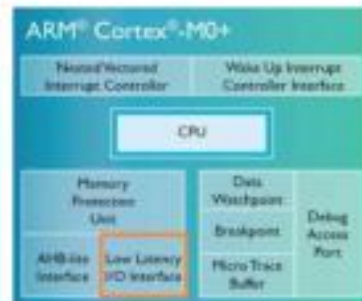
# ARM Cortex products

## ARM Cortex-M Product Line

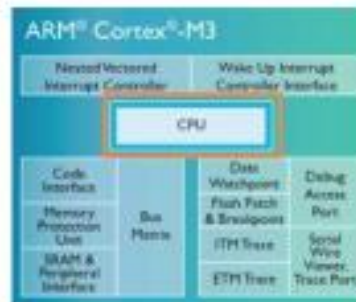
Consistent 32 bit processor architecture across all applications



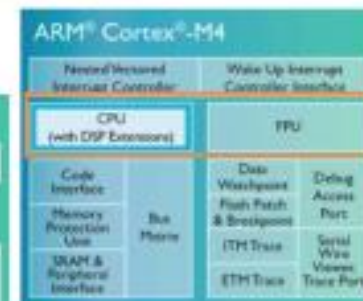
Lowest cost  
Low power



Lowest power  
Outstanding energy efficiency



Performance efficiency  
Feature rich connectivity



Digital Signal Control  
Processor with DSP  
Accelerated SIMD  
Floating point

'8/16-bit' Traditional application space

'16/32-bit' Traditional application space

ARM



# Features of ARM processors

---

- RISC based architecture
- Despite this, they are quite powerful
  - Advanced architecture ideas
  - Not like your Arduino devices!
- Not just one product, but a family
  - There is a common instruction set for backward compatibility
  - Anything in ARM7, can be run in ARM9 for example

# Design Philosophy of ARM processors

---

- Small processor to lower power consumption
  - Needed for Embedded systems
- High code density
  - Helps limit memory and physical size
- Interfacing with low-cost memory
- Reduced size for processor
  - Allows more room for ASIC

# RISC Architecture Features

---

- Instructions
  - Reduced set
  - Single cycle
  - Fixed length
- Pipelined
  - Decode in one stage
  - No need for micro-programming

# RISC Architecture Features

---

- Registers
  - Large number of general purpose registers
- Load/Store Architecture
  - Only LOAD and STORE instructions can access memory and registers
  - All other instructions can only work with registers

# ARM vs RISC Architecture - Differences

---

- Strictly speaking, ARM processors are not RISC architectures
  - Certain instructions require variable number of cycles for execution
  - Barrel shifter
  - Thumb mode → 16 bit instruction set
- ARM is mostly RISC