AI in Embedded Systems

Class 3: Toolkit - Hardware

February 26, 2020
Louis W. Giokas

This Week's Agenda

Monday	Overview and Requirements
Tuesday	Toolkit - Algorithms
Wednesday	Toolkit - Hardware
Thursday	Systems
Friday	Application Examples

Course Description

This topic will cover a new approach to developing embedded systems which includes AI/Machine Learning(ML) as an element of the suite of tools available. As microcontroller devices become more powerful, these techniques are now within reach. We will look at the types of AI/ML algorithms available and appropriate in embedded systems. We will also look at how interaction with higher level systems, such as cloud analytics, can be integrated to create systems that evolve and improve over time and space.

Today's Agenda

- Overview
- Memory
- Processor Types
- Connectivity
- Processing Systems
- Conclusion/Next Class

Overview

- The other side of the system is the hardware. We will discuss the advances in hardware that make the implementation of AI/ML in embedded systems. This is a fast moving area driven by improvements in areas such as memory capacity, processor speed and new processor types.

Overview

- The wide range of algorithms available have a large range of processing requirements
- Different input types are also a factor
- Some approaches have small memory and processing impacts, especially when parameters change slowly and can be computed externally
- Adaptive systems have the highest requirement on memory, since they must store history locally

Overview

- Many of the advances we have seen are very recent
- Memory density continues to increase
- Memory speed continues to increase
- Power requirements decrease
- Novel processing elements are coming to the embedded space
- Higher levels of integration put diverse elements on the same piece of silicon, simplifying design

Memory

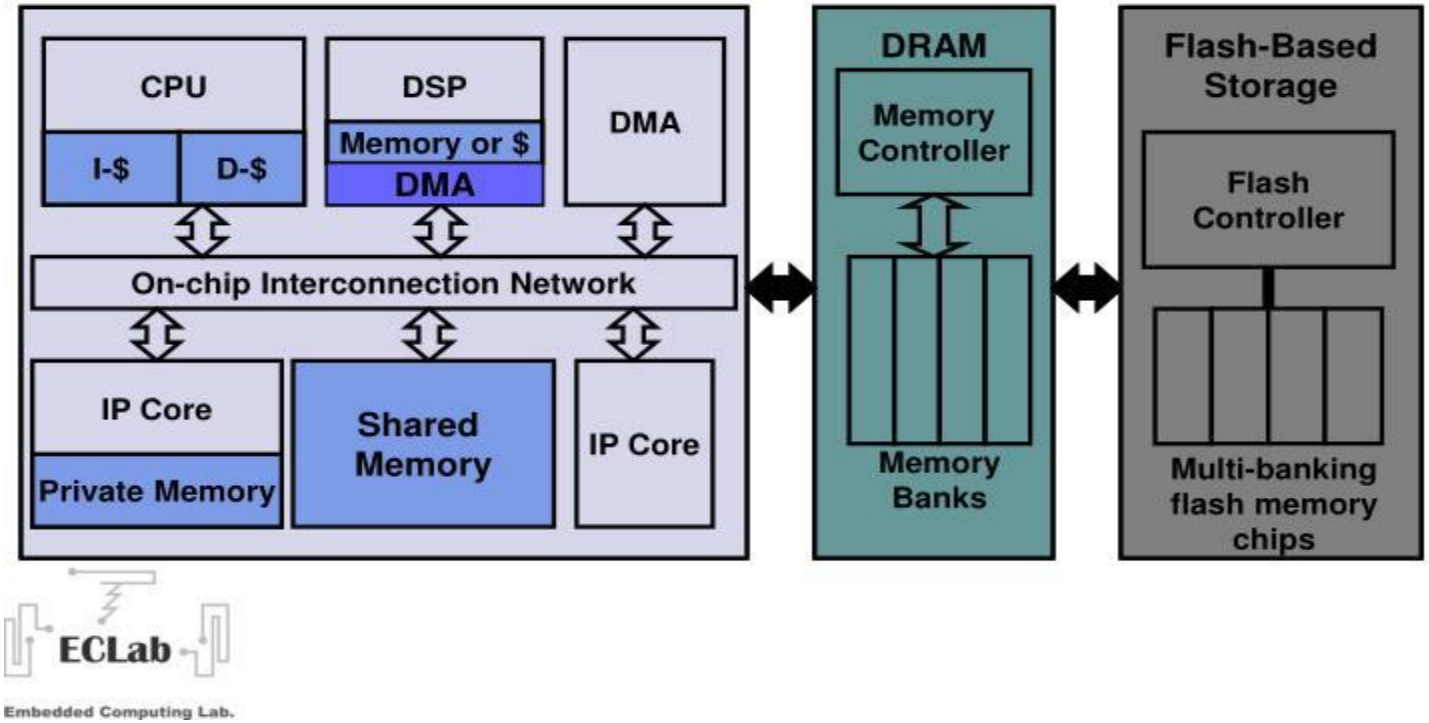
- Data is the driver for AI. As we have seen, AI systems “learn” from the data.
- In many cases we need this data only to train the system, and this can be done offline, if connectivity is available.
- Higher level algorithms need this data locally to integrate it with new measurements to learn and adapt

Memory

- Critical memory parameters
  - Speed
  - Size
  - Access
- Arranged in a hierarchy
- External memory
  - Long term storage
  - High speed interfaces

Memory

Memory Hierarchy in an Embedded System



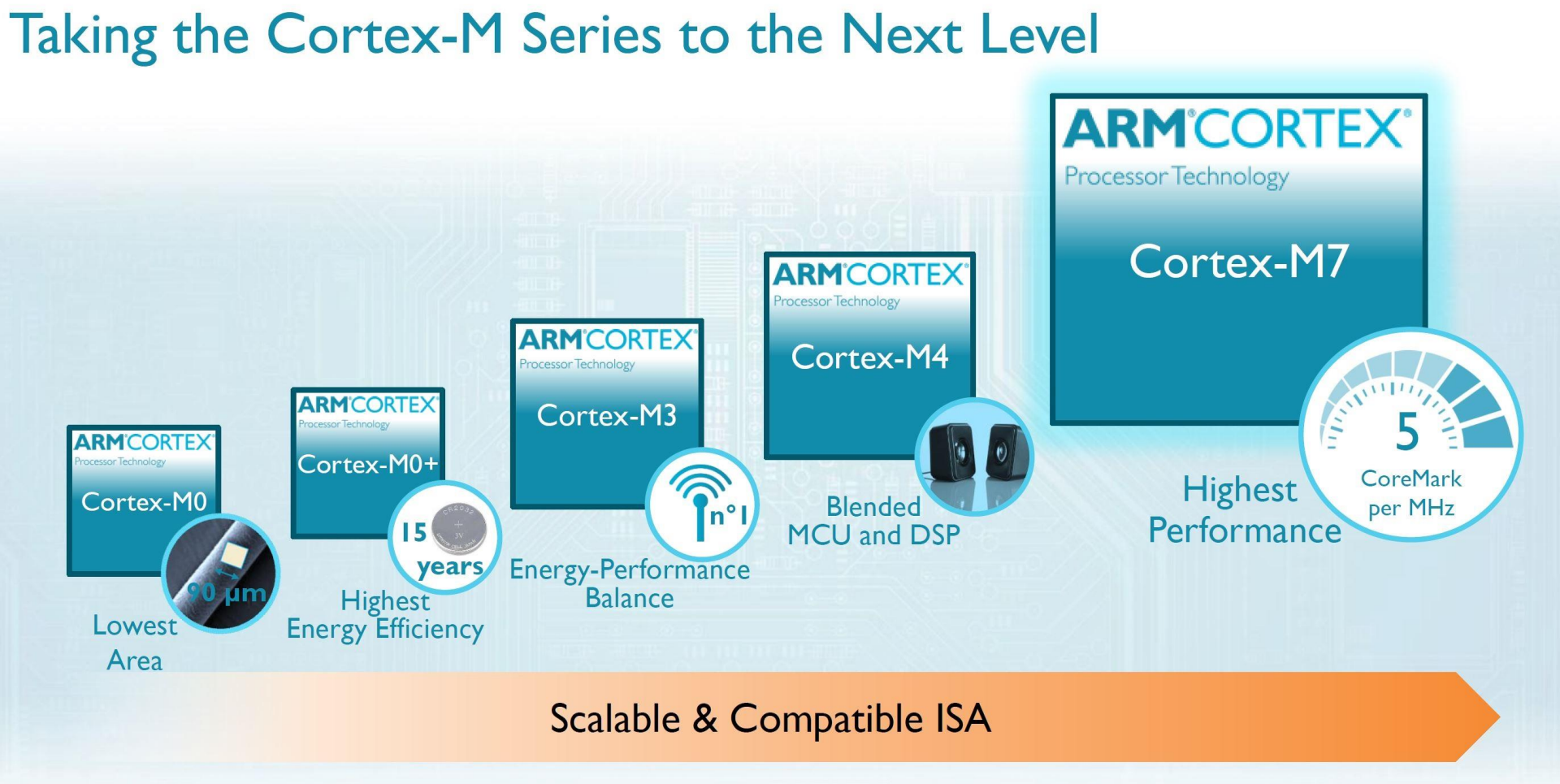
Processor Types

- Many types of processing elements are available
  - Traditional CPU
    - May contain complex instructions such as SIMD and MIMD
  - Digital Signal Processor (DSP)
  - Graphics Processing Unit (GPU)
  - Field Programmable Gate Array (FPGA)

Processor Types

- CPU
  - Most common used in embedded systems is the ARM family of processors
  - These are RISC machines with extended instructions in some models
  - Many vendors and processor configurations available
  - Include extensive interrupt processing and power management
  - Typically use the M series

Processor Types

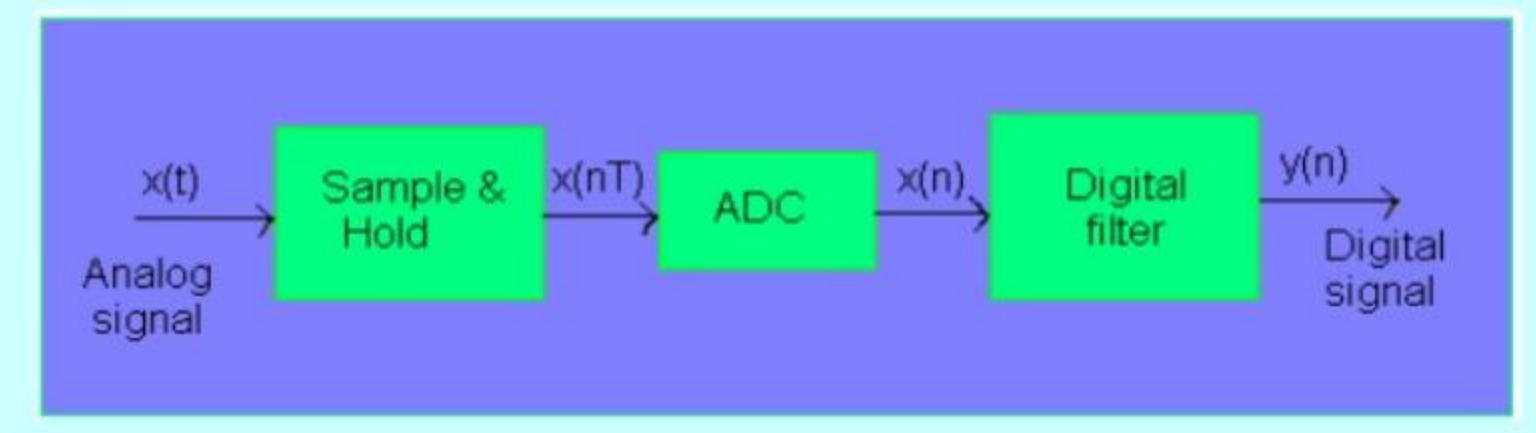


Processor Types

- DSP
  - Signal is converted from analog to digital
  - Once converted, digital filtering techniques can be applied
    - Matrix operations such as Convolution
    - Finite Impulse Response (FIR) filters
    - Fast Fourier Transform (FFT)
  - Purpose built instructions are included to directly implement these techniques, simplifying programming

Processor Types

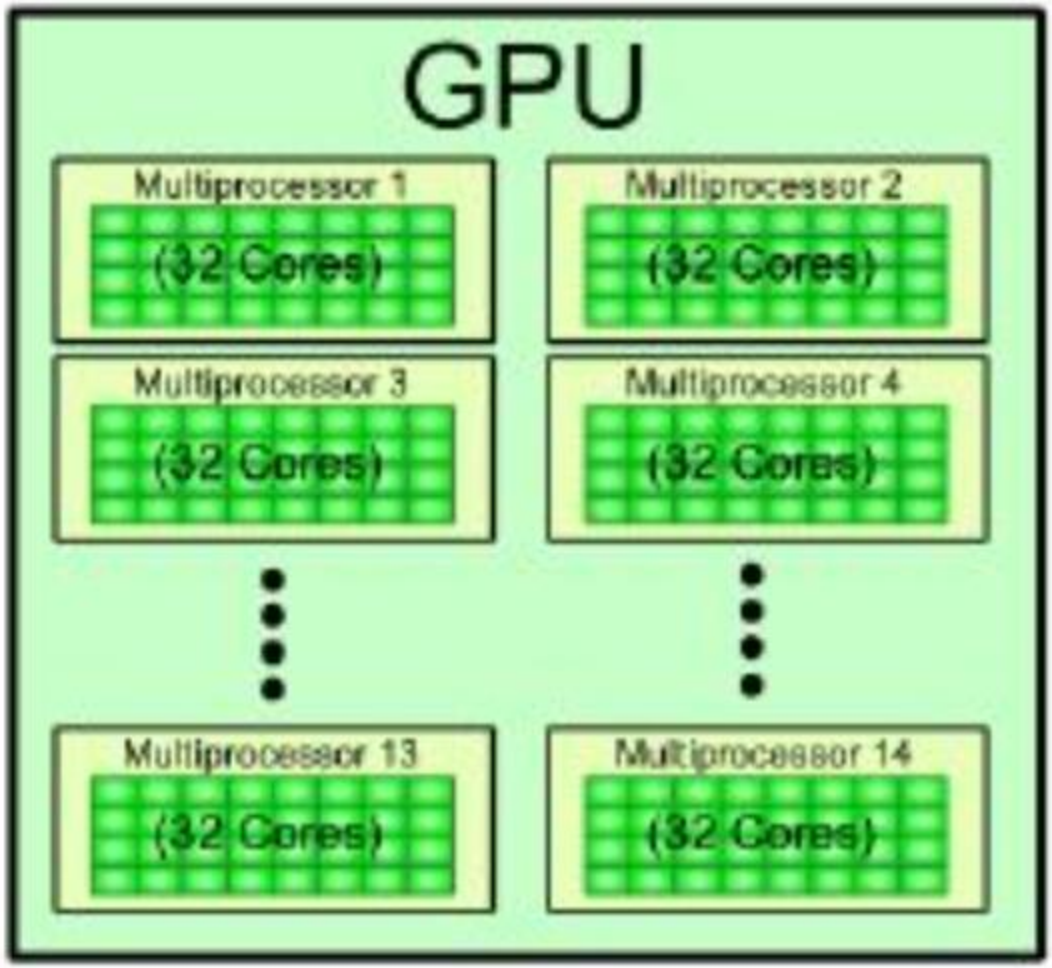
Typical DSP Processing Flow



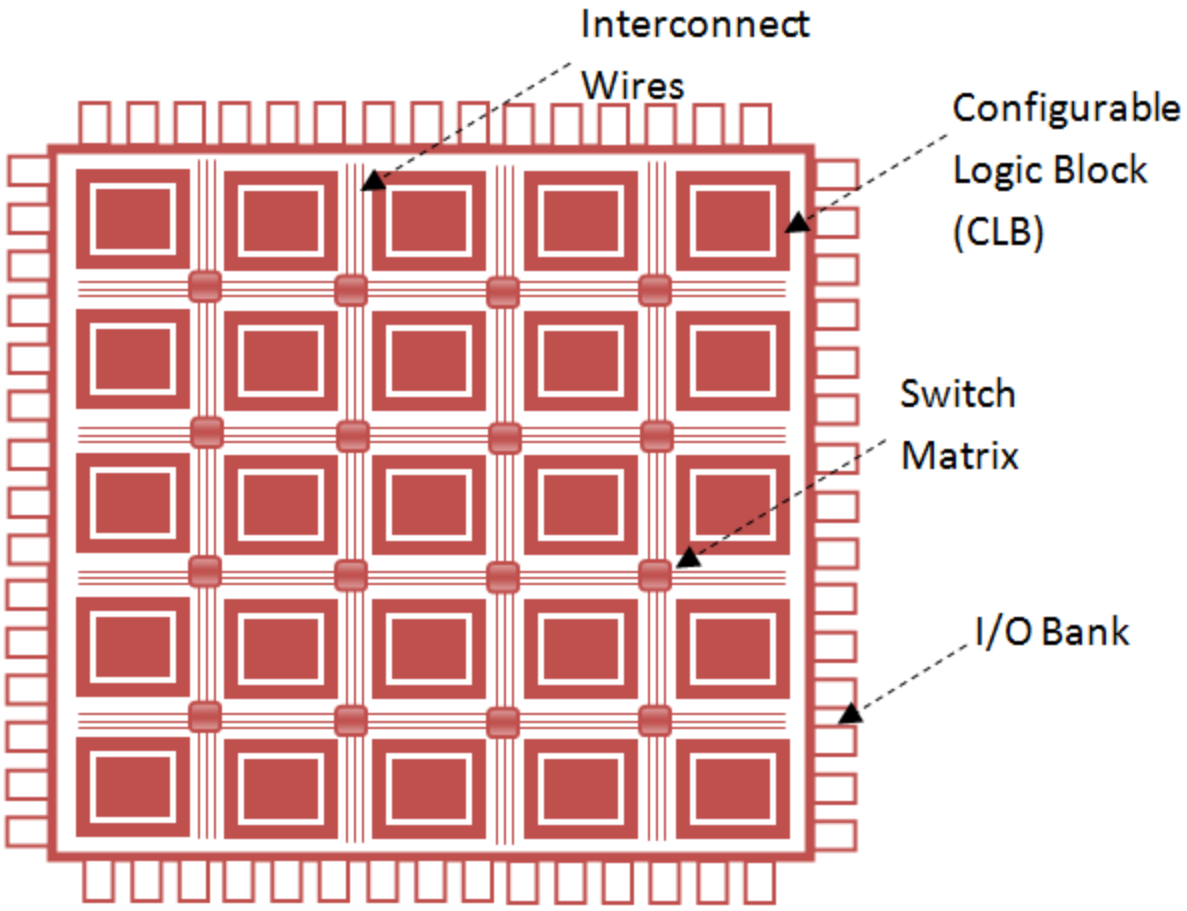
Processor Types

- GPU
  - Highly parallel architectures with simplified processors arranged in a grid with high speed memory access
  - Appropriate for a number of types of algorithms, including Neural Networks
  - Often used in a coprocessor arrangement with traditional CPUs and DSPs

Processor Types



Processor Types



Basic FPGA Architecture

Connectivity

- Connectivity options determine system configuration and are driven by the application environment
  - Some applications will have little or no connectivity
  - Local and global connectivity options will extend the types of algorithms that can be used as well as overall system architecture
  - Speed and capacity of communication links vary widely
    - Higher values allow for more complex algorithms

Connectivity

- Protocols typically encountered
  - Bluetooth
  - Zigbee
  - Wi-Fi
  - Ethernet
  - TCP/IP
  - Cellular (dedicated data)
  - 5G

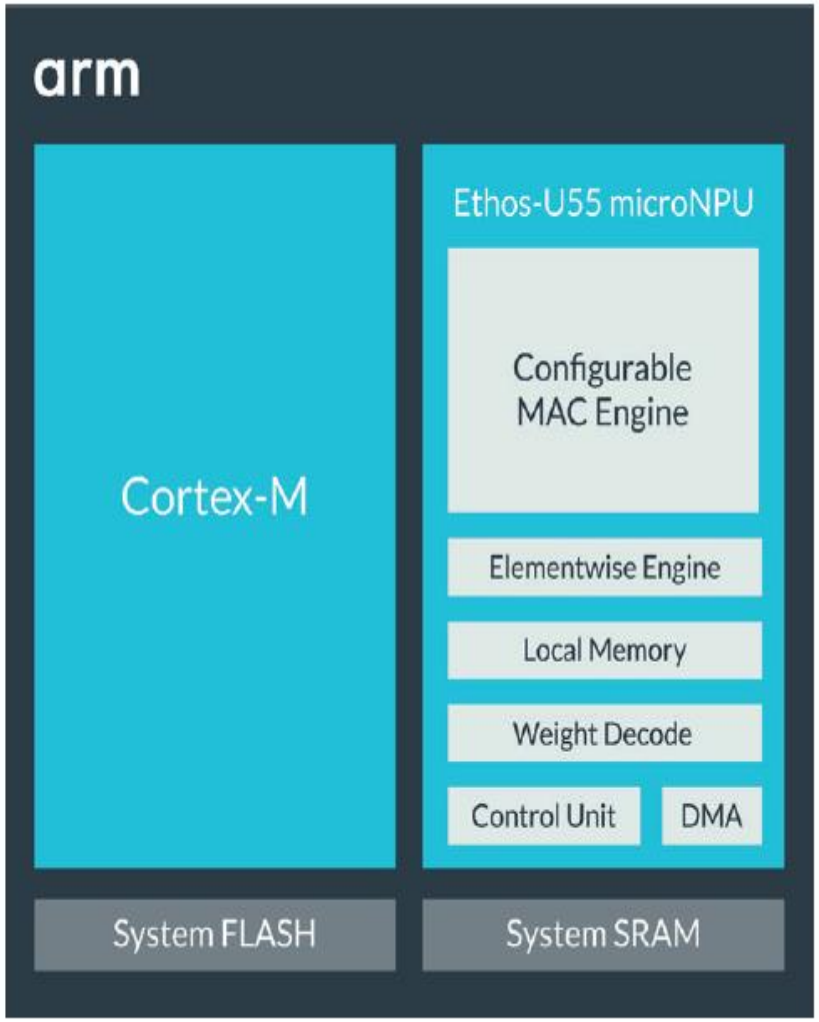
Processing Systems

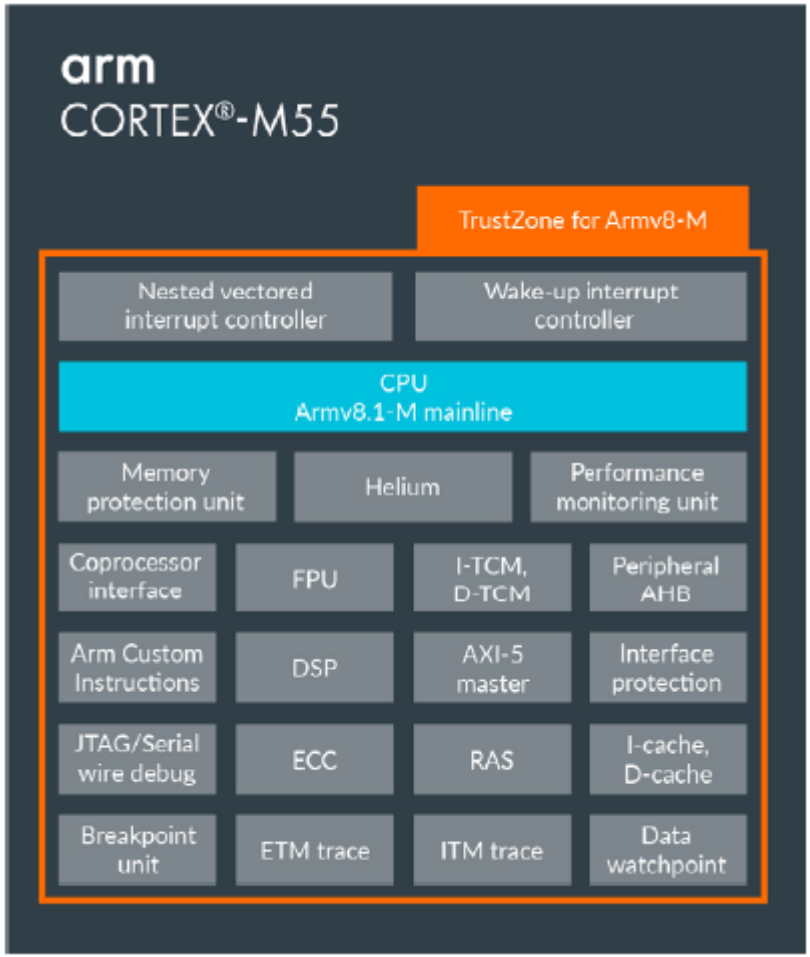
- New processing systems, especially from ARM. Combine a number of the elements we have discussed on a single piece of silicon
- ARM provides IP and reference designs, that are then implemented, often with enhanced features, by a large number of chip manufacturers
  - Some examples are MicroSemi, NXP, Cypress, STMicro

Processing Systems

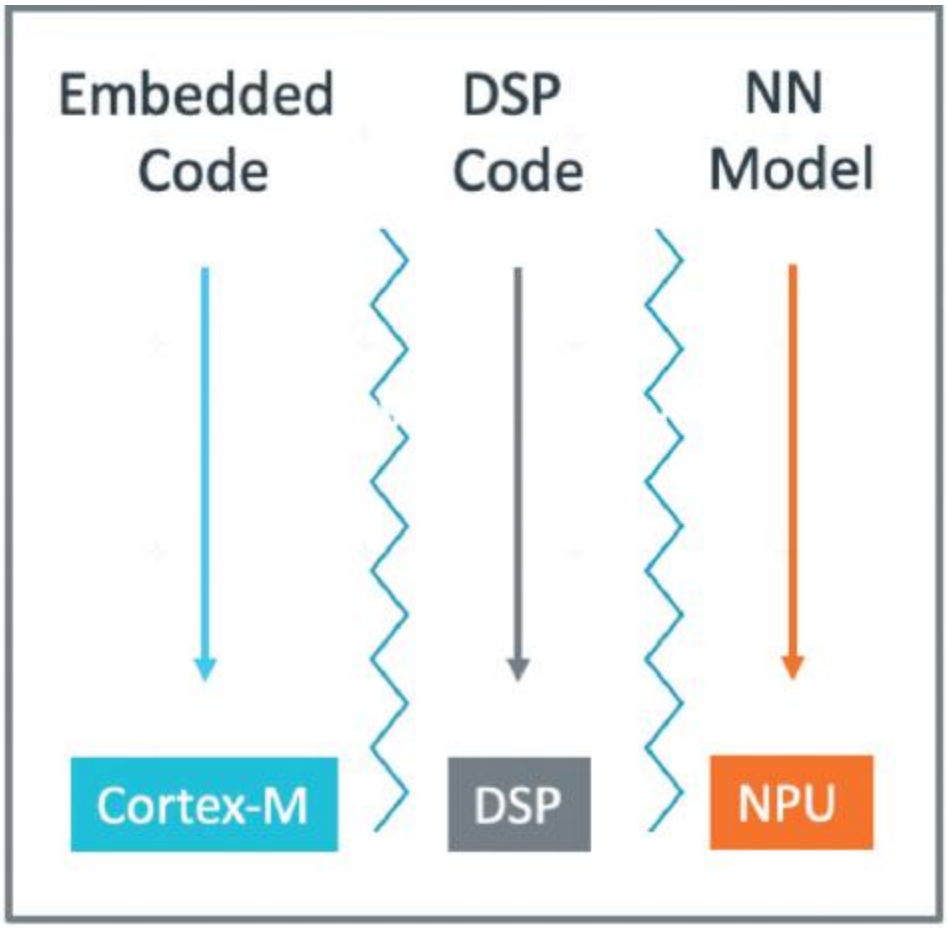
- Example: ARM
  - Corstone-300: Reference Design
    - Ethos-U55 microNPU (Neural Processing Unit)
    - Cortex-M55 (AI-capable Cortex-M processor)
  - These designs enable the implementation of NN in an embedded systems context
  - Have elements that provide extensive acceleration for NN and DSP functions

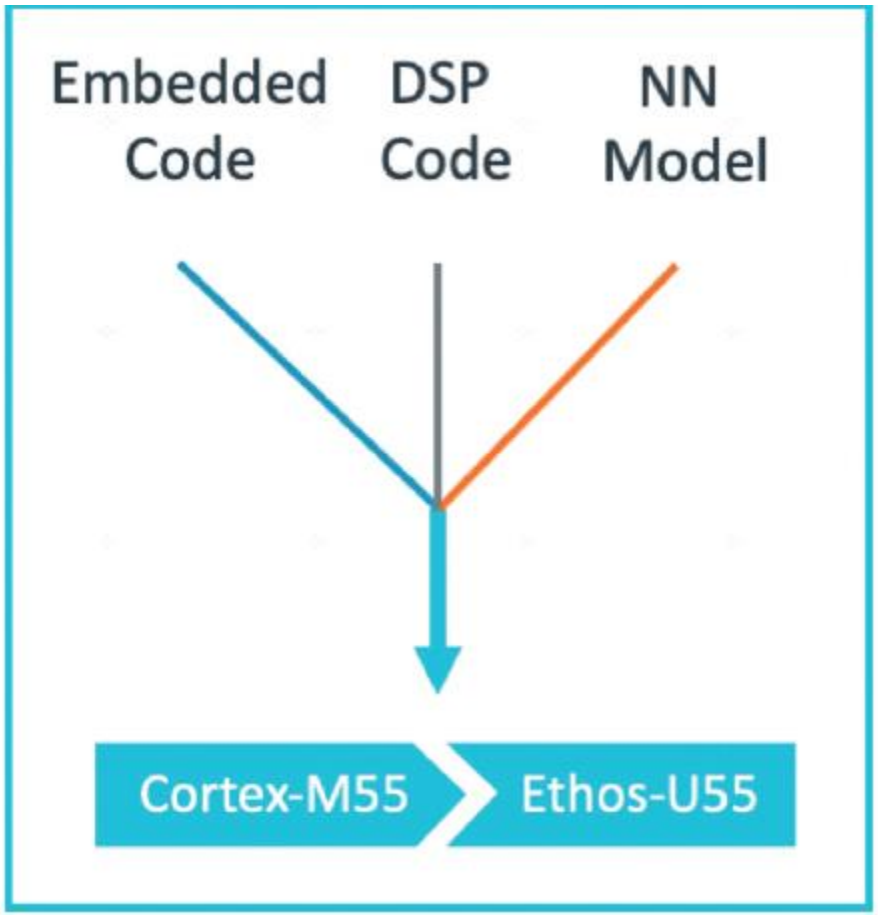
Processing Systems





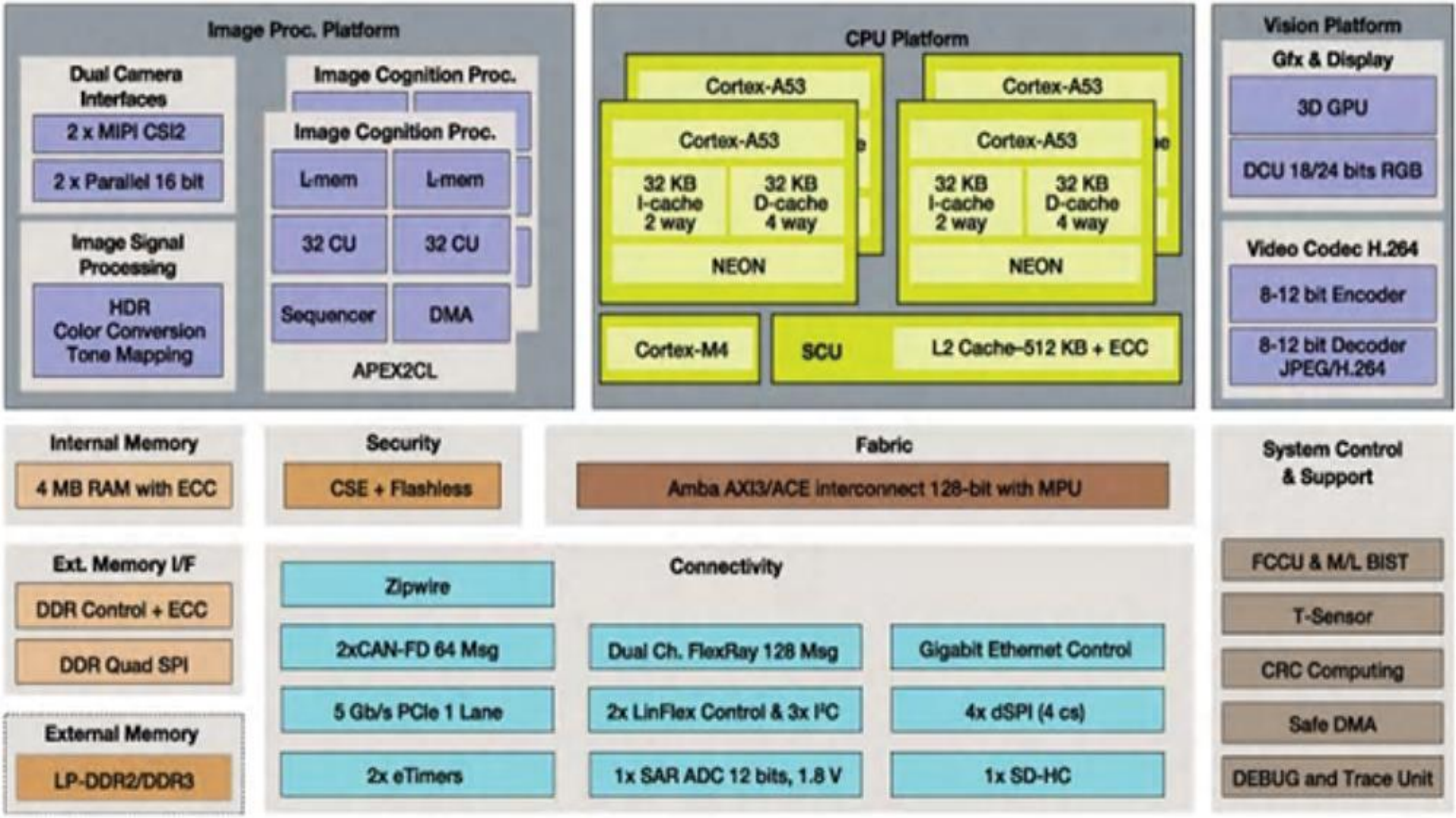
Processing Systems





Two views of code development and mapping

Processing Systems



Conclusion/Next Class

- Today we have reviewed the hardware Toolkit of the embedded AI landscape
- We reviewed many aspects including memory, processor types, connectivity and processing systems
- Tomorrow we will systems approach tp embedded AI