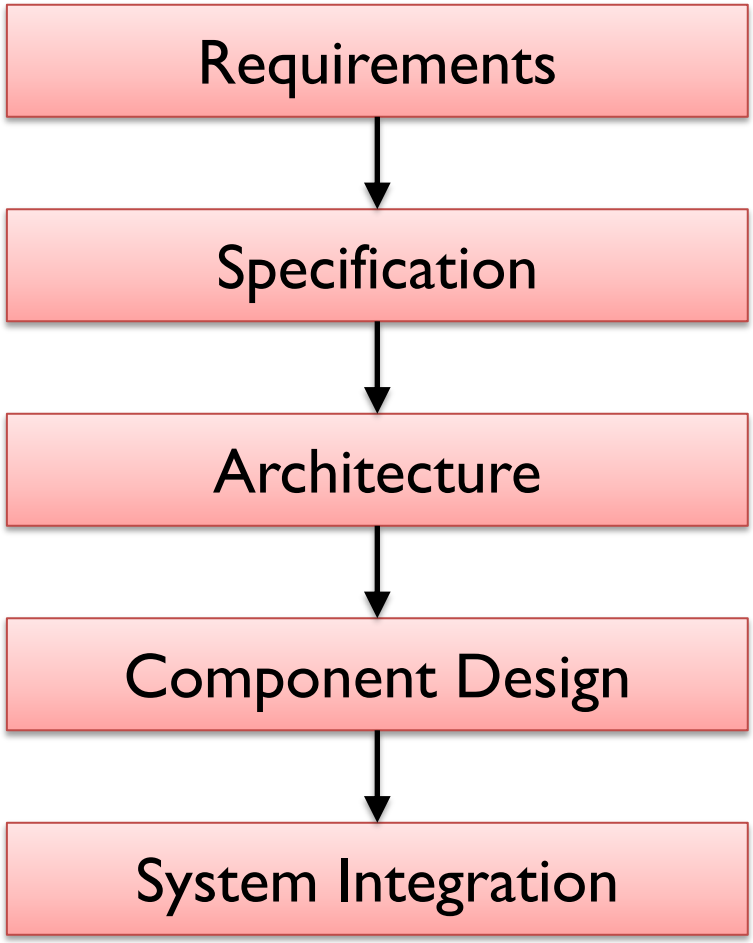
# Embedded Systems: Architecture

Jinkyu Jeong (Jinkyu@skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu

# Designing Embedded Systems



# Basic Architectures

# Control Unit

- Custom logic
- FPGAs (Field-Programmable Gate Arrays)
- Microcontrollers
- Microprocessors
- DSPs (Digital Signal Processors)
- ASIPs (Application Specific Instruction-set Processors)
- Multicore? (symmetric vs. asymmetric)
- Typical word size: 8/16/32-bit

# Why Microprocessors?

- Microprocessors
  - ARM, MIPS, PowerPC, SuperH, Cell, Atom, …
  - Mostly less than 1 GHz
- Microprocessors are often very efficient
  - Can use same logic to perform many different functions
- Microprocessors simplify the design of families of products

# The Performance Paradox

- Microprocessors use much more logic to implement a function than does custom logic
- But microprocessors are often at least as fast:
  - Heavily pipelined
  - Large design teams
  - Aggressive VLSI technology, …

# Power

- Custom logic uses less power, but CPUs have advantages
  - Modern microprocessors offer features to help control power consumption
  - Software design techniques can help reduce power consumption

- Heterogeneous systems
  - Some custom logic for well-defined functions, CPUs + software for everything else

# RAM

- SRAM
  - Easier to integrate on the same chip as processor
- DRAM
  - SDRAM (Synchronous DRAM): SDR/DDR/DDR2/DDR3
  - Mobile SDR/DDR SDRAM
  - RDRAM (Rambus DRAM)
- NVRAM (Non-Volatile RAM)
- Future NVRAMs: PRAM, MRAM, FeRAM
- Cache memory
- SPM (Scratch Pad Memory)

# ROM

- Mask-programmed
  - Connections programmed at fabrication
- OTP (One-time programmable) ROM
  - Connections programmed after manufacture by user
- EPROM (Erasable programmable ROM)
- EEPROM (Electrically erasable programmable ROM)

# Flash Memory

- NOR Flash
- NAND Flash
- e-MMC
- UFS
- Cards (MMC, SD, CF, …)

# Interfacing

- ARM AMBA (Advanced Microcontroller Bus Architecture)
- ISA (Industry Standard Architecture)
- PCI (Peripheral Component Interconnect)

- $I^2C$ (Inter-IC) bus
- USB

# Common Peripheral Devices

- Interrupt controller
- DMA controller
- Timer/Counter
- Real-time clock
- Watchdog timer
- UART (Universal Asynchronous Receiver Transmitter)
- IrDA (Infrared)
- Ethernet (wired/wireless)
- Bluetooth

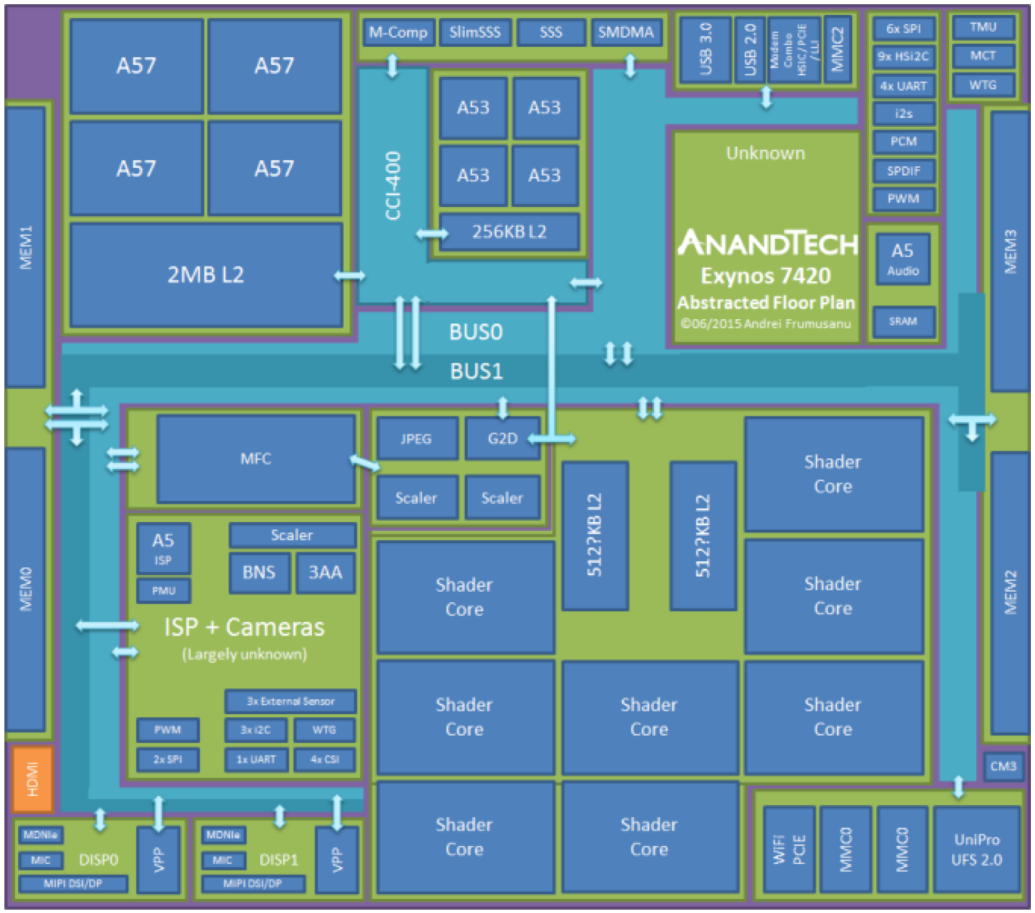
# Recent Trends

- Increasing computation demands
- Increasingly networked
- Increasing need for flexibility

- Getting complex
- Increasingly platform-based
  - Hardware architecture + associated software

# SoC (System-On-a-Chip)

- Samsung Exynos 7420




*Source: http://www.anandtech.com/show/9330/exynos-7420-deep-dive*

# Introduction to the ARM Architecture

# Overview (1)

- Advanced RISC Machine
- The most widely used 32-bit ISA
  - 98% of the more than 1 billion mobile phones sold used at least one ARM processor (2005)
  - 4 billion shipped in 2008
  - 90% of all embedded 32-bit RISC processors (2009)
- ARM architecture has been extended over several versions.

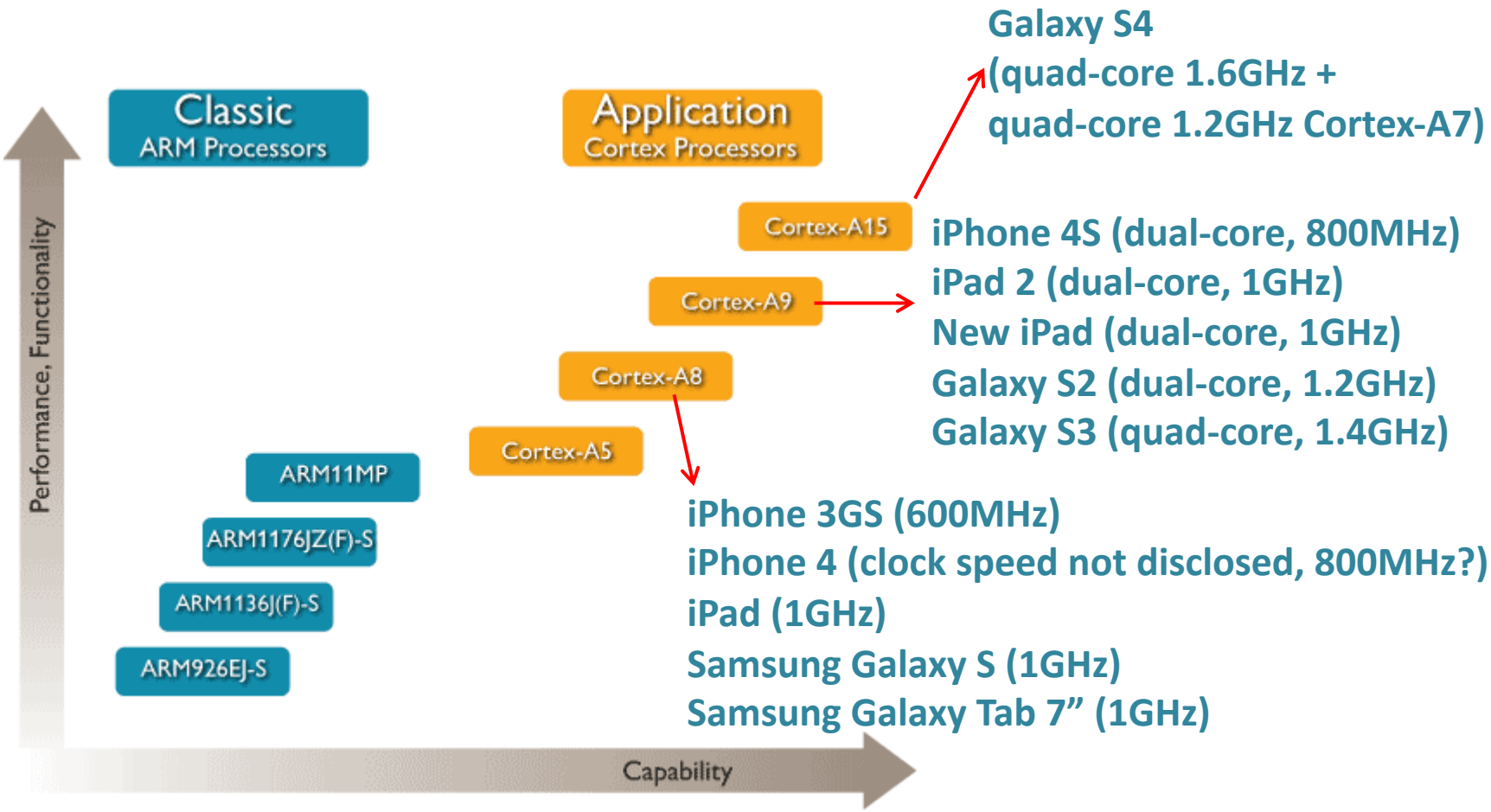
# Overview (2)

- ARM processors developed by licensees
  - DEC StrongARM
  - Marvell (formerly Intel) Xscale
  - Nintendo
  - TI OMAP
  - Qualcomm Snapdragon
  - Samsung Hummingbird/Exynos
  - Apple A4/A5/A5X/A6/A7…
  - Nvidia Tegra
  - Applied Micro X-Gene

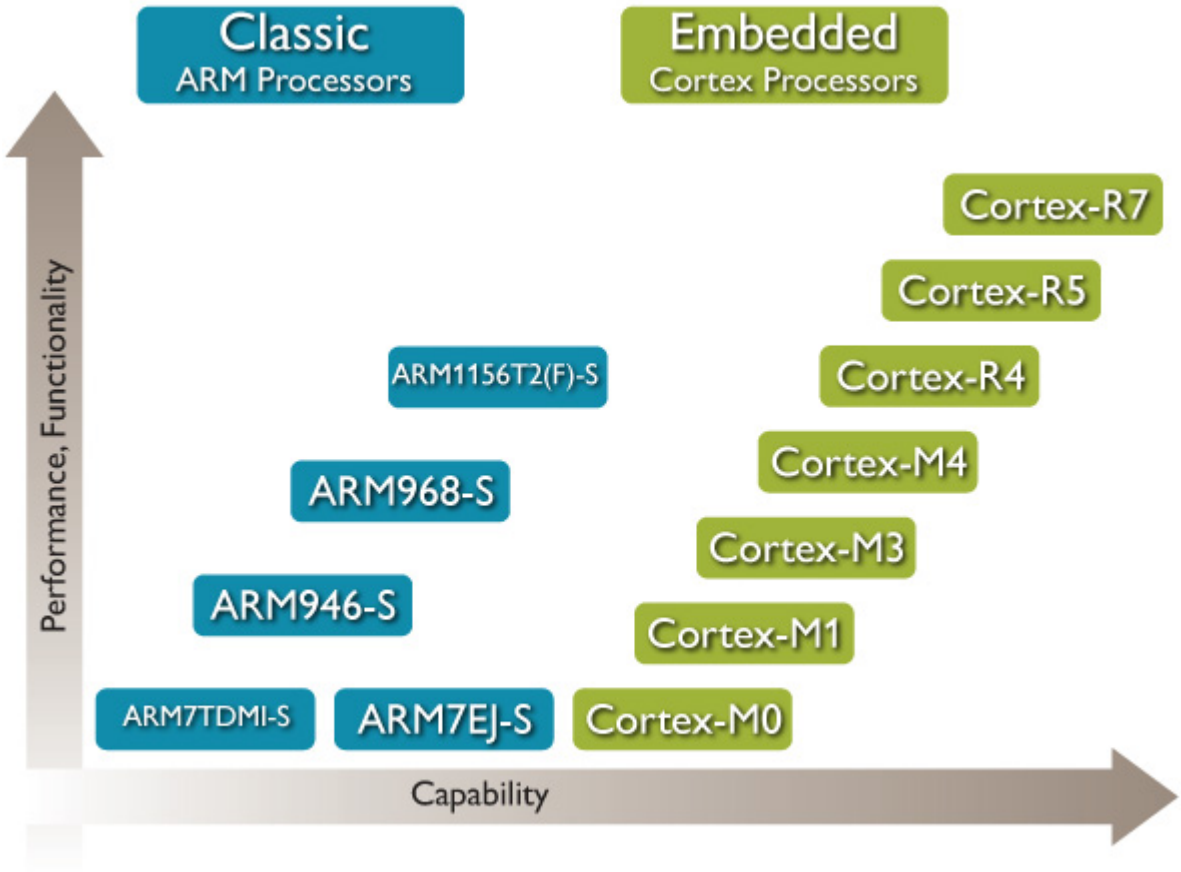
# ARM Design Goals

- High performance
- Small code size
- Low power consumption
- Small silicon area

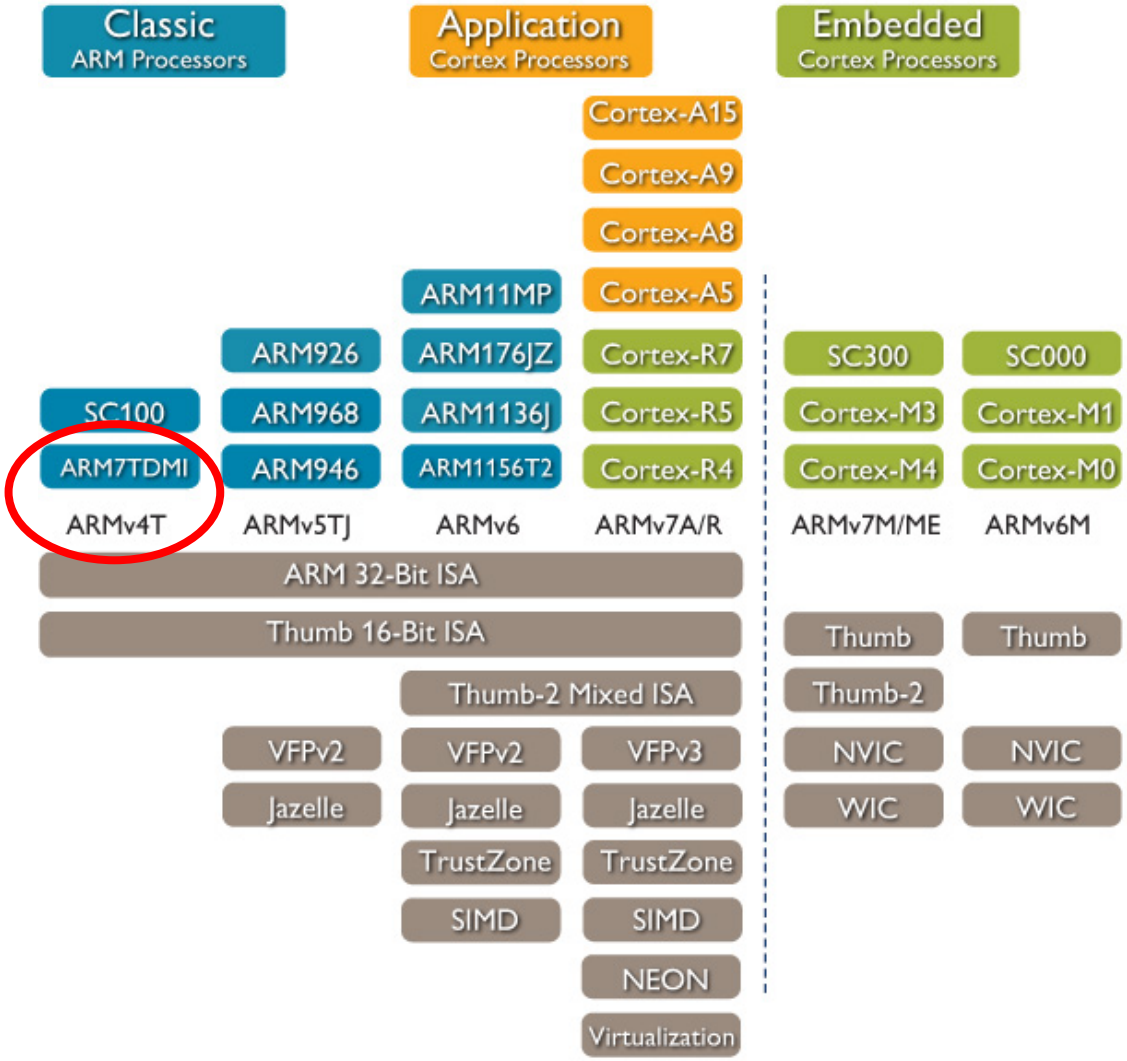
# Application Processors



# Embedded Processors



# Processors Lineup



# ARM 7TDMI-S

Synthesizable core

Thumb Debug Multiplier ICE

- Implements the ARMv4T architecture
- Applications
  - iPod from Apple
  - Nintendo DS & Game Boy Advance
  - Most of Nokia's mobile phones
  - Lego Mindstorms NXT
  - iriver portable digital audio players
  - Roomba 500 series from iRobot

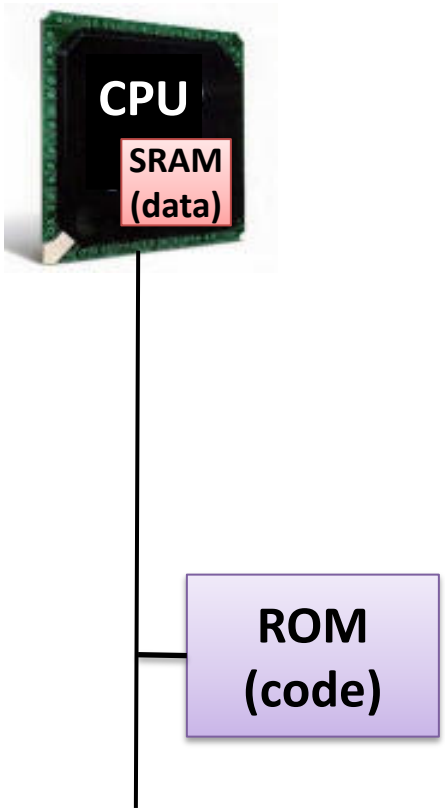
# Thumb Instruction Sets

- 16-bit instruction set
- A subset of the most commonly used 32-bit ARM instructions
- Significantly improved code density at a cost of some reduction in performance
  - Typically 65% of the ARM code size
  - 160% performance of the ARM code when running from a 16-bit memory system
- Transparently decompressed to full 32-bit ARM instructions in real time, without performance loss
  - ARM instructions only for performance critical segments

# Memory Architectures

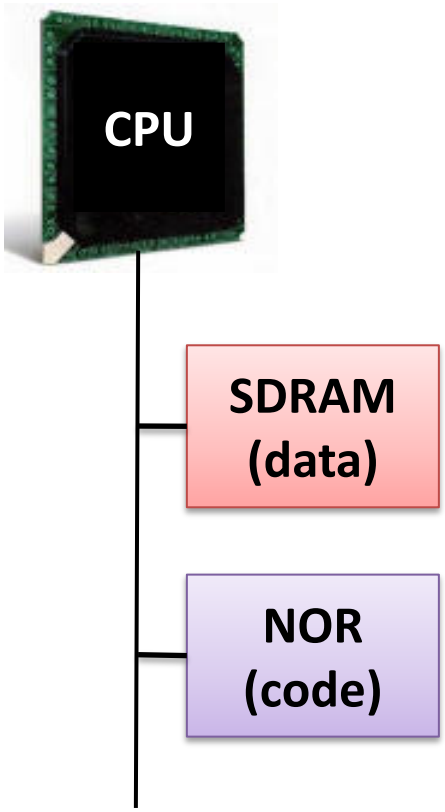
# Minimalist Approach

- Pros
  - Simple, easy to design
- Cons
  - Limited ROM size
  - Limited SRAM size
  - No firmware upgrades



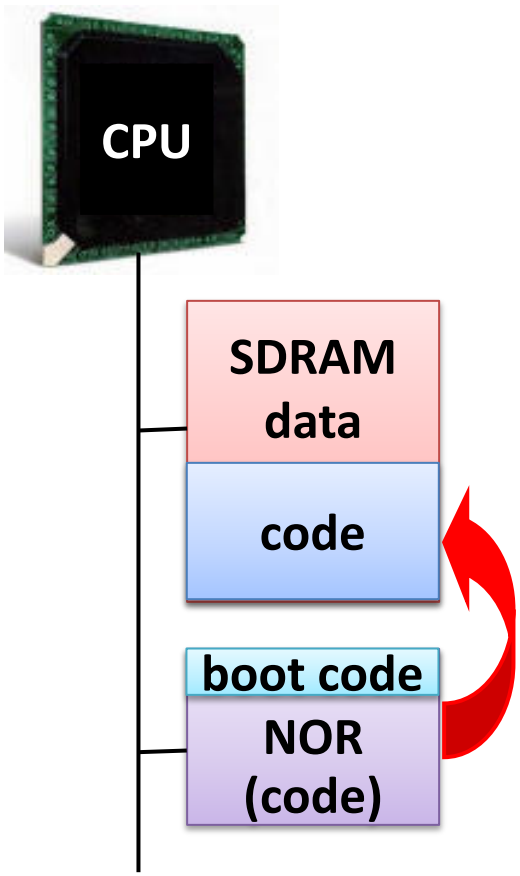
# NOR XIP

- Pros
  - Simple, easy to design
  - Execute-In-Place (XIP)
  - Predictable read latency
  - Code + Storage in NOR
  - Firmware upgrades
- Cons
  - Slow read speed
  - Much slower write speed
  - The high cost of NOR



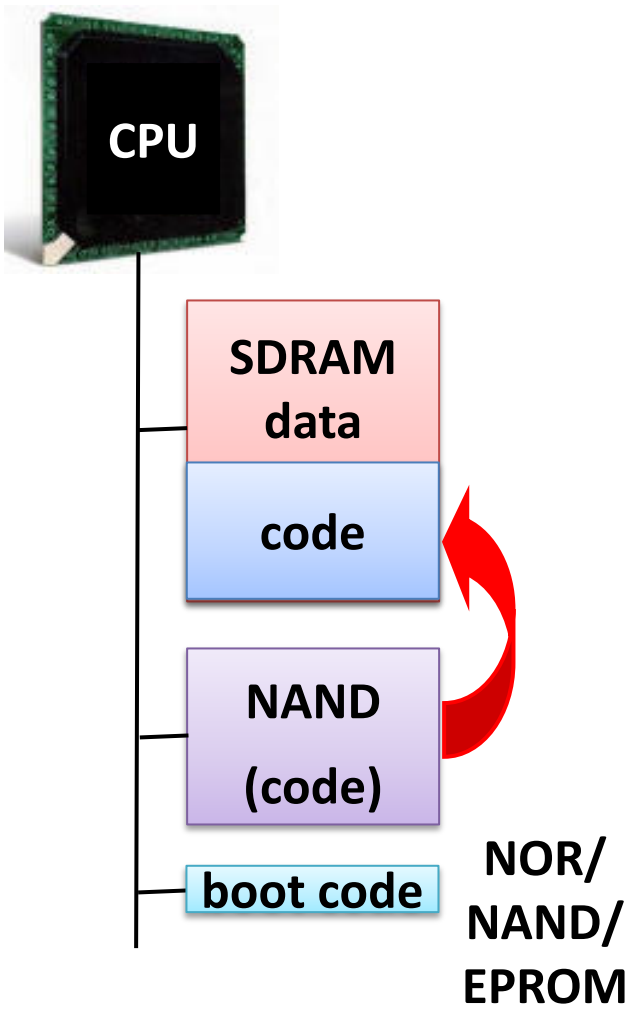
# NOR Shadowing

- Pros
  - Faster read and write
  - Easy boot-up
  - Use a relatively pricey NOR only to boot up the system
  - Code can be compressed
- Cons
  - Larger DRAM needed
  - Require more design time
  - Not energy efficient



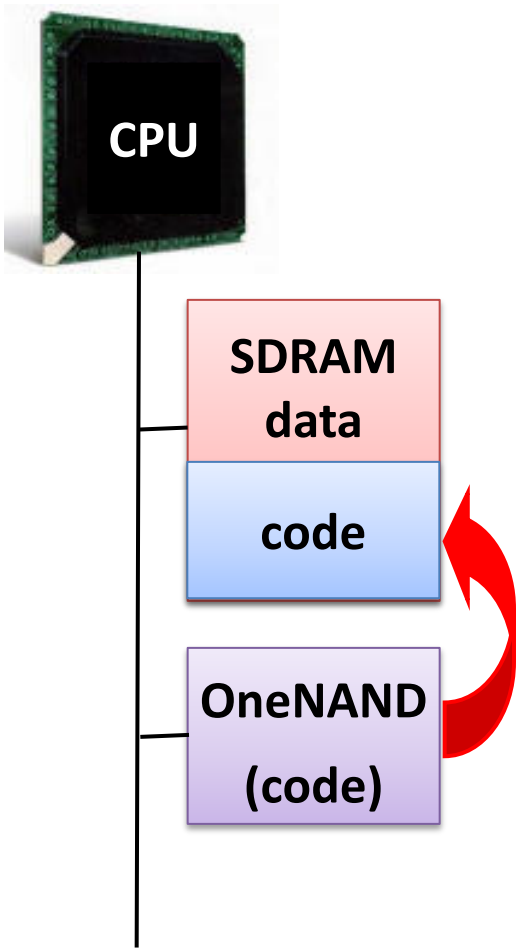
# NAND Shadowing

- Pros
  - Faster read and write
  - Cost effective
  - NAND for both code and data storage
- Cons
  - Require a special boot mechanism
  - Extensive ECC for NAND
  - Larger DRAM needed
  - Require more design time
  - Not energy efficient



# Hybrid NAND Shadowing

- Pros
  - Much faster read and write speed
  - ECC embedded
  - Cost effective
  - NAND for both code and data storage
- Cons
  - Larger DRAM needed
  - Not energy efficient



# NAND Demand Paging

- Pros
  - Less DRAM required
  - Low cost
  - Energy efficient
  - NAND for both code and data storage
- Cons
  - Require MMU-enabled CPU
  - Unpredictable read latency
  - Complex to design and test