Embedded Systems: Architecture

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

- Requirements
- Specification
- Architecture
- Component Design
- System Integration
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

- **iPhone 3GS (600MHz)**
- **iPhone 4 (clock speed not disclosed, 800MHz?)**
- **iPad (1GHz)**
- **Samsung Galaxy S (1GHz)**
- **Samsung Galaxy Tab 7” (1GHz)**
- **iPhone 4S (dual-core, 800MHz)**
- **iPad 2 (dual-core, 1GHz)**
- **New iPad (dual-core, 1GHz)**
- **Galaxy S2 (dual-core, 1.2GHz)**
- **Galaxy S3 (quad-core, 1.4GHz)**
- **Galaxy S4 (quad-core 1.6GHz + quad-core 1.2GHz Cortex-A7)**

Performance, Functionality vs. Capability
Embedded Processors

Classic ARM Processors
- ARM7TDMI-S
- ARM946-S
- ARM968-S

Embedded Cortex Processors
- Cortex-M0
- Cortex-M1
- Cortex-M3
- Cortex-M4
- Cortex-R4
- Cortex-R5
- Cortex-R7

Performance, Functionality vs. Capability
Processors Lineup

- Classic ARM Processors
  - ARMv4T
  - ARMv5TJ
  - ARMv6
  - ARM11MP
  - ARM1176JZ
  - ARM1136J
  - ARM1156T2
  - ARM926
  - ARM968
  - ARM946

- Application Cortex Processors
  - Cortex-A15
  - Cortex-A9
  - Cortex-A8
  - Cortex-A5
  - Cortex-R7
  - Cortex-R5
  - Cortex-R4
  - SC300
  - Cortex-M3
  - Cortex-M1
  - Cortex-M4
  - Cortex-M0

- Embedded Cortex Processors
  - ARMv7M/ME
  - ARMv6M
  - Thumb
  - Thumb-2
  - Thumb-2 Mixed ISA
  - VFPv2
  - VFPv3
  - Jazelle
  - Jazelle
  - TrustZone
  - TrustZone
  - SIMD
  - NEON
  - Virtualization

ICE3028: Embedded Systems Design, Fall 2018, Jinkyu Jeong (jinkyu@skku.edu)
ARM 7TDMI-S

- 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