ECE 332 – Embedded Systems Lab

Introduction

Prof. Sandip Kundu

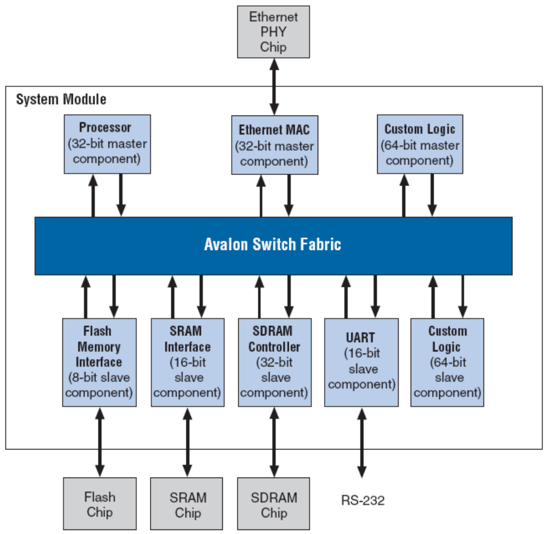
Acknowledgement

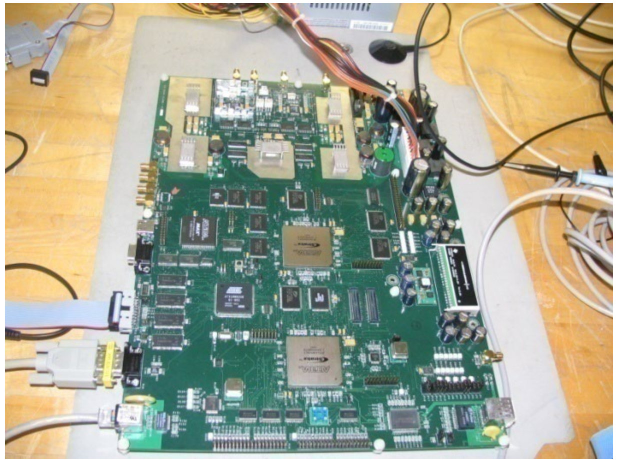
- These slides are a result of cumulative contributions from Prof. Burleson, Koren, Kundu and Moritz.

- Materials have also been adopted from the textbook: *Wolf, Computers as Components, Morgan Kaufman*, 2016

The Big Picture

- What are embedded systems?
- Challenges in embedded computing system design.
- Design methodologies.





- Sophisticated functionality.
- Real-time operation.
- Low manufacturing cost.
- Low power.
- Designed to tight deadlines by small teams.

Definition

- Embedded system: any device that includes a programmable computer but is not itself a general-purpose computer.
- Focuses on application characteristics to optimize:
  - don’t need all the general-purpose bells and whistles.

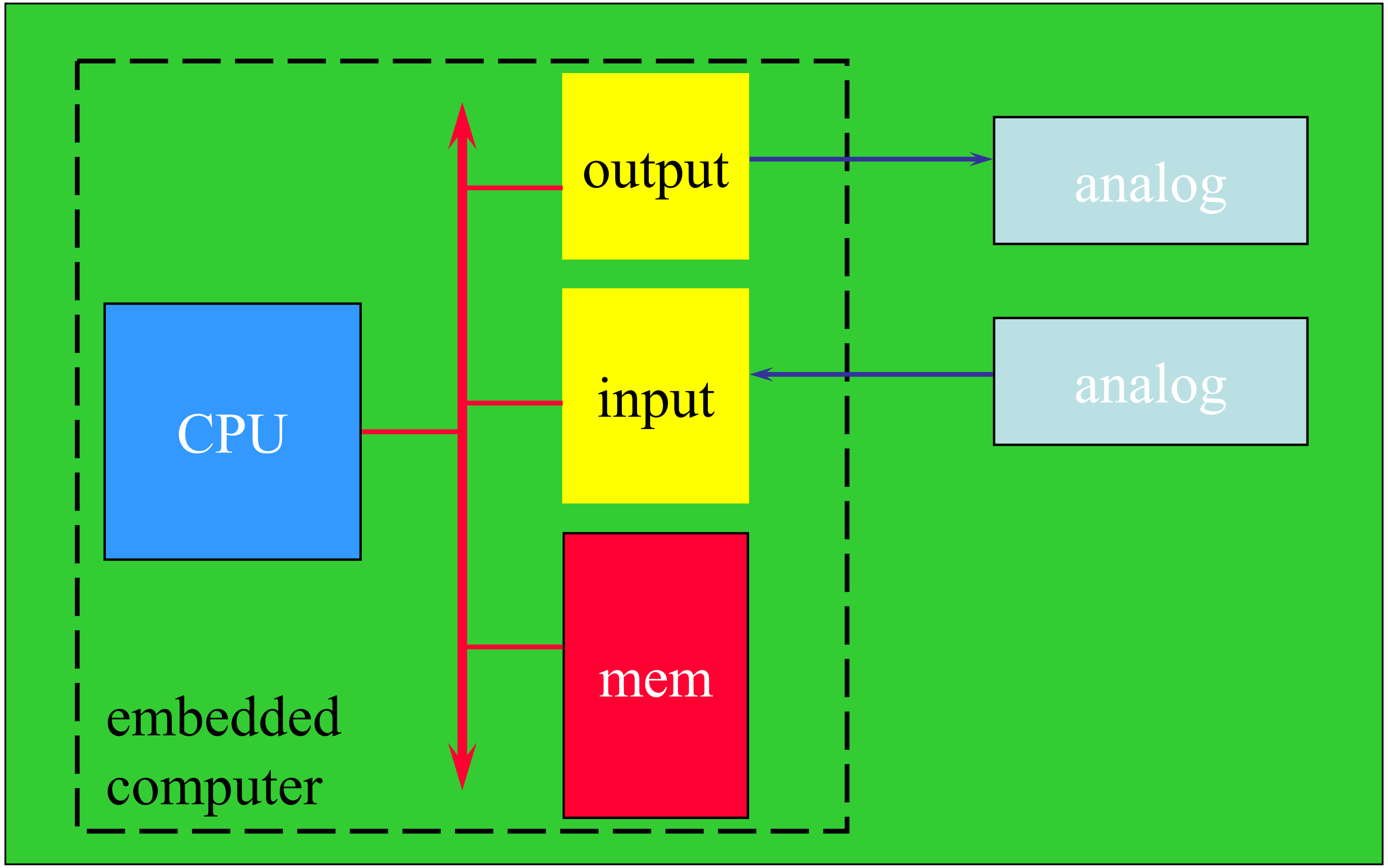
Examples

- Cell phones, Webcams, Navigation Systems
- Routers, Blade servers, Wireless PC cards
- Automobiles, Car Alarms, Keyless Entry Systems
- Building security, card swiping systems
- Embedded Medical Devices - Pacemakers

Characteristics of embedded systems

- Sophisticated functionality.
- Real-time operation.
- Low manufacturing cost.
- Low power.
- Reliable and secure.
- Designed to tight deadlines by small teams.

Embedding a computer: a very simplified view...



Non-functional requirements

- Many embedded systems are mass-market items that must have low manufacturing costs.
  - Limited memory, microprocessor power, etc.
- Power consumption is critical in battery-powered devices.
  - Excessive power consumption increases system cost even in wall-powered devices.
- Must finish operations by deadlines.
  - Hard real time: missing deadline causes failure.
  - Soft real time: missing deadline results in degraded performance.

Microprocessors in Embedded Systems

Microprocessor alternatives for embedded systems

- Ordinary microprocessor: CPU plus on-chip cache units.
- Microcontroller: includes I/O devices, on-board memory.
- Digital signal processor (DSP): microprocessor optimized for digital signal processing.
- Hard core vs. soft core.
- Typical embedded word sizes: 8-bit, 16-bit, 32-bit.

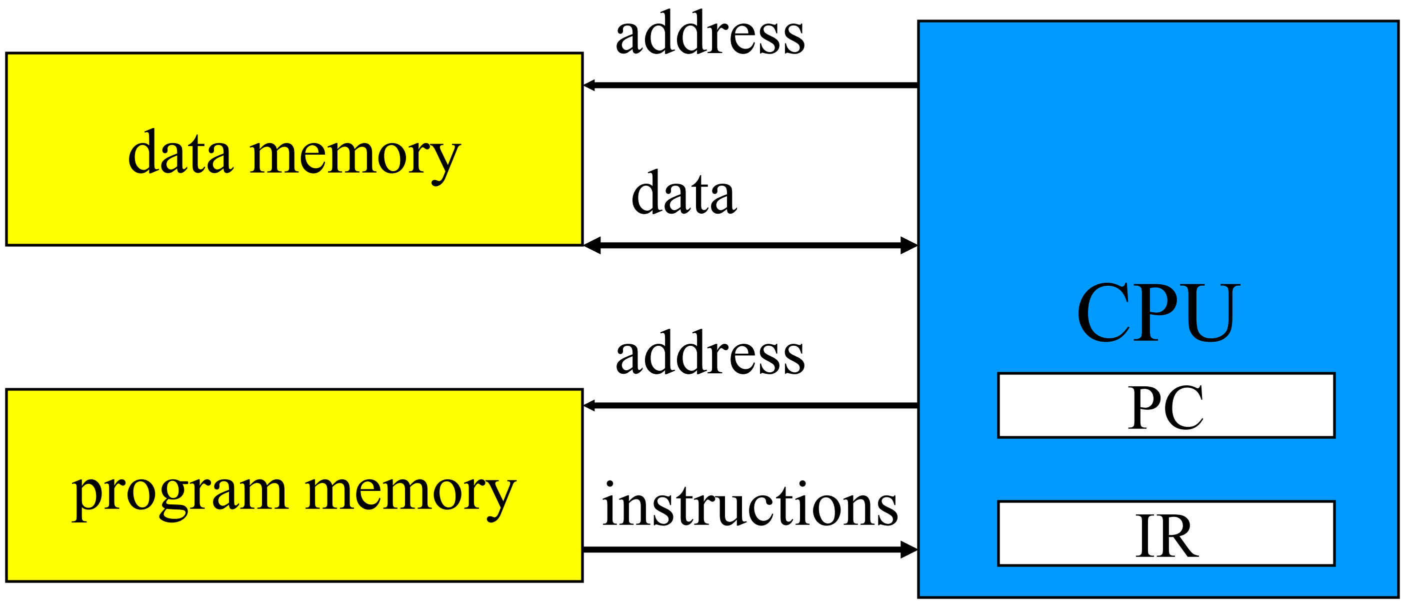
Embedded microprocessors

- ARM, MIPS, Power PC, Freescale, 8051, X86
- Various purposes
  - Networks – MIPS, PowerPC
  - Mobile phone – ARM dominated
  - Industrial – Freescale Coldfire
  - Security – 8051 based, Infineon, BlueRISC
  - High performance – X86, Intel Epic, other VLIW and superscalars, multi-core ARM, Tilera tiled processor

Von Neumann CPU Architecture

- Memory holds data and instructions.
- Central processing unit (CPU) fetches instructions from memory.
  - Separation between CPU and memory distinguishes programmable computer.
- CPU registers:
  - program counter (PC)
  - general-purpose registers

Harvard architecture

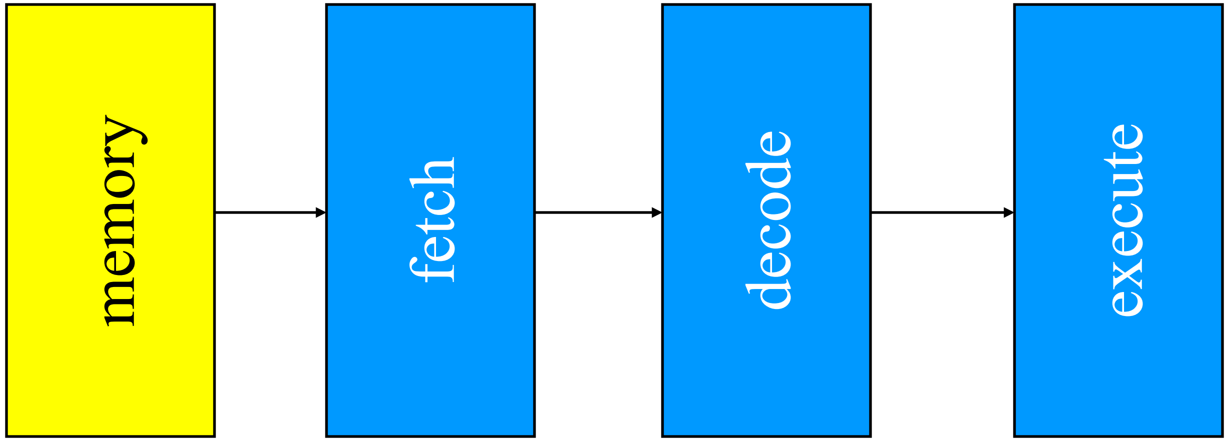


RISC vs. CISC

- Complex instruction set computer (CISC):
  - many addressing modes
  - most operations can access memory
  - variable length instructions
- Reduced instruction set computer (RISC):
  - only load/store can access memory
  - fixed-length instructions
- Instruction set architectures – characteristics:
  - Fixed vs. variable length.
  - Addressing modes.
  - Number of operands.
  - Types of operands.

Pipelining

- Execute several instructions simultaneously but at different stages.
- Pipeline hazards.
- Simple three-stage pipe:



Why use microprocessors?

- Alternatives: random logic on a field-programmable gate arrays (FPGAs), custom logic, etc.
- 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;
  - caches and virtual memory
  - aggressive VLSI technology.

Power

- Custom logic is a clear winner for low power devices.
- Modern microprocessors offer features to help control power consumption.
  - Software design techniques can help reduce power consumption.
  - Compiler-architecture interaction
  - Circuit-level optimizations (CAD tool support)

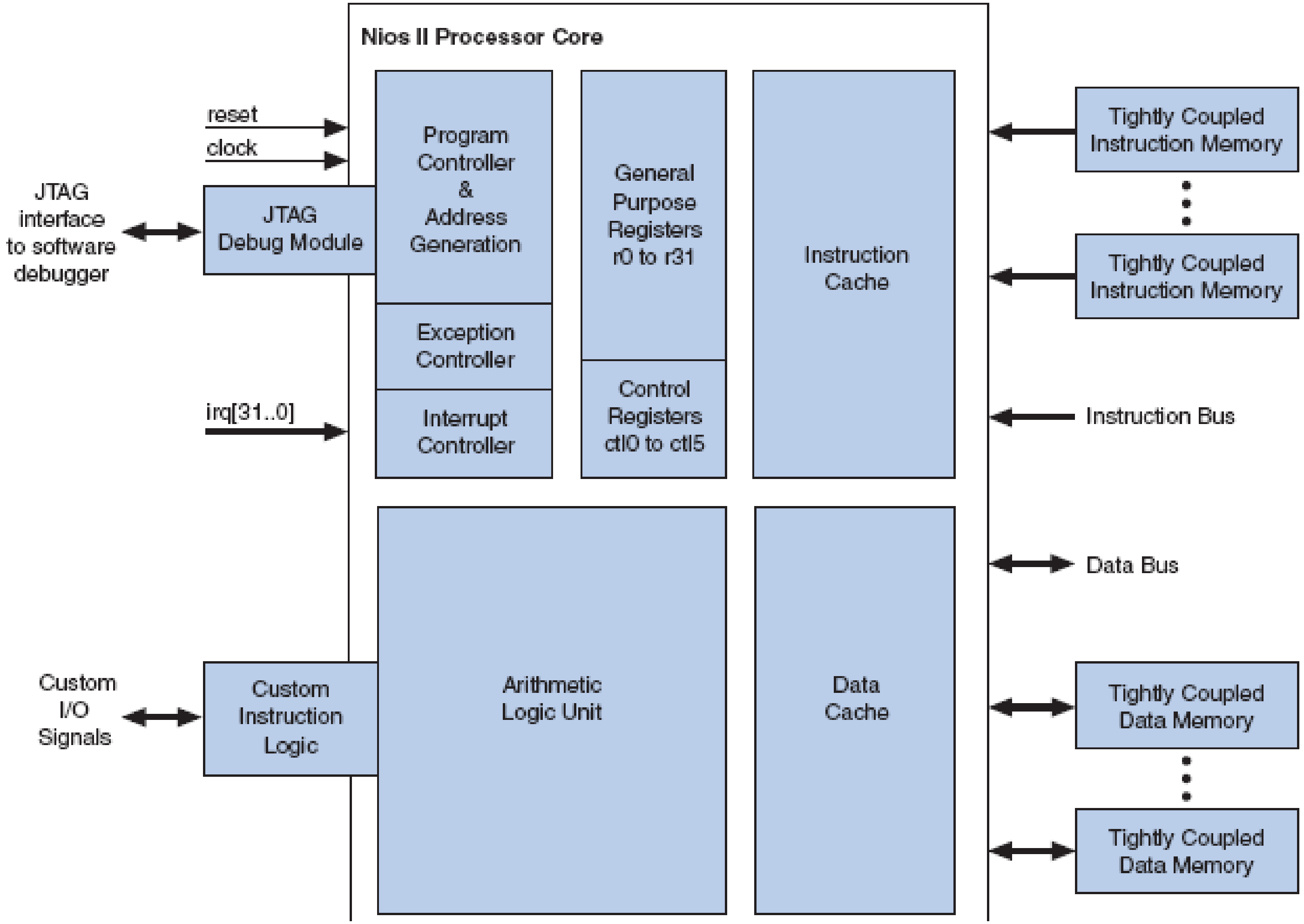
Soft Core Processors

- Are soft, i.e. specified through field programming just like programmable logic
  - Shipped as hardware description files, which can be mapped onto FPGA. e.g: Nios 2.
  - Offer flexibility as microprocessor parameters can be tuned to the application.
  - You can test and validate many designs quickly.
  - Fast time to market.

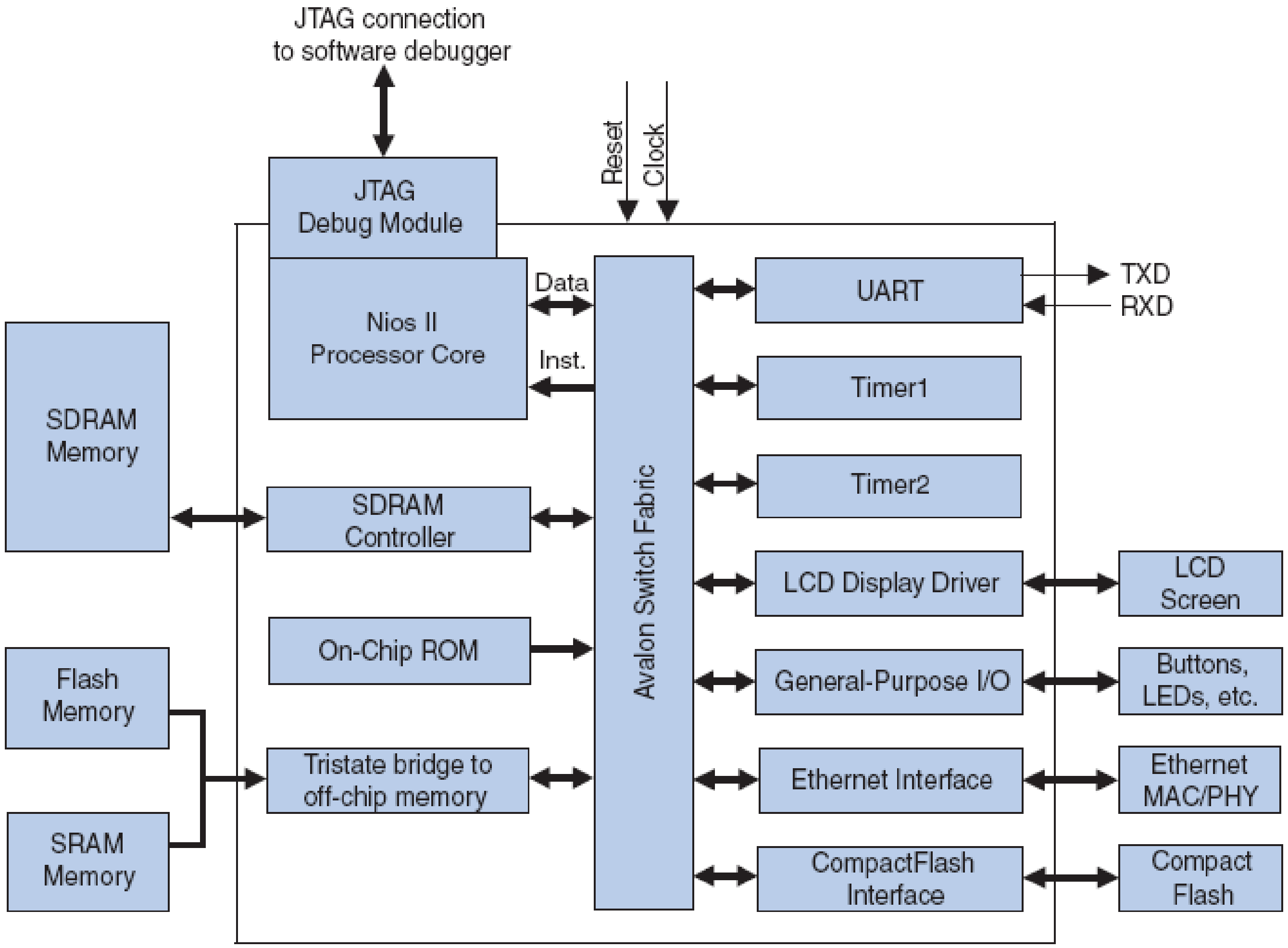
What is Nios 2?

- A 32-bit soft core processor from Altera
- Is a RISC, Harvard Architecture: Simple instructions, separate data and instruction memories.
- Has 32 levels of interrupts.
- Uses the Avalon Bus interface
- Programs compiled using GNU C/C++ toolchain.

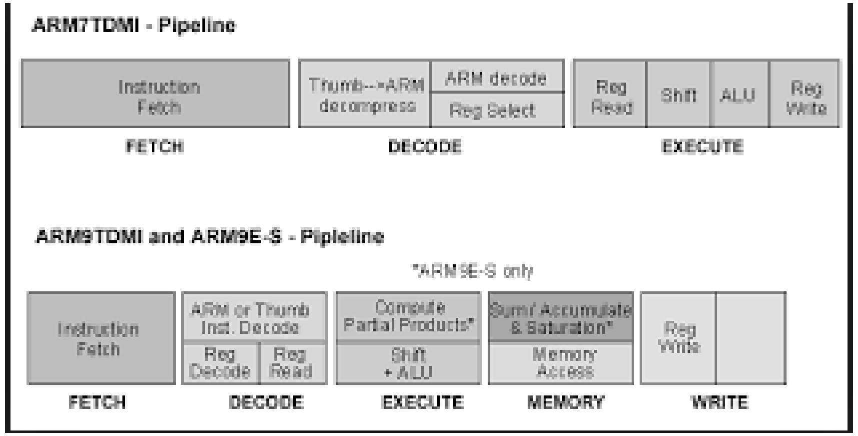
Nios 2 Architecture:



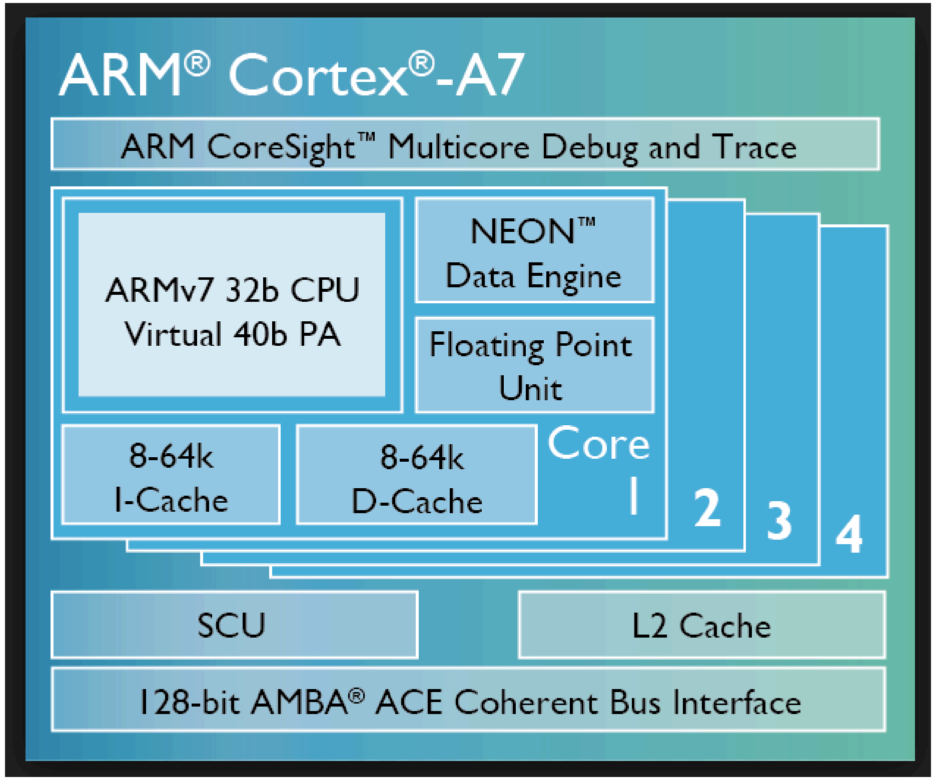
SOPC system having NIOS:



Pipelined CPUs – like ARM, Multi-Core SoCs



ARM7-9 Pipelines



ARM multi-core with ARMv7

Application Processors

ARM CPUs Currently

Application

RT

Low power

Security