Embedded Processors

Hongwei Zhang

http://www.cs.wayne.edu/~hzhang/

Ack.: this lecture is prepared in part based on slides of R. Williams
Views of Processors

- **Instruction Set Architecture (ISA)**
  - The view of the machine as seen by the programmer

- **MicroArchitecture (in chip)**
  - Physical realization of the ISA

- Can one ISA be realized using more than one MicroArchitecture?
ISA Variety

- General Purpose Computing Exhibits Limited Variety of ISAs
  - What are the costs of ISA specialization?

- Embedded Computing Exhibits Wide Variety of ISAs
  - What are the benefits of ISA specialization?
Desirable Processor Characteristics

- General Purpose Computing
  - What do you want?

- Embedded Computing
  - What is different?

Generality vs. power, time, and size efficiency
Outline

- Types of processors
- Parallelism
Outline

- Types of processors
- Parallelism
Types of processors

- Microcontroller
- Digital signal processor (DSP)
- Graphics processor
Types of processors

- Microcontroller
- Digital signal processor (DSP)
- Graphics processor
Microcontrollers

- An Embedded Computer System *on a Chip*
  - A CPU
  - Memory (Volatile and Non-Volatile)
  - Timers
  - I/O Devices
- Typically intended for limited energy usage
  - Low power when operating plus sleep modes
- Where might you use a microcontroller?
What is Control?

- Sequencing operations
  - Turning switches on and off
- Adjusting continuously (or at least finely) variable quantities to influence a process
Programmable Logic Controllers

- A microcontroller system for industrial automation
  - Continuous operation
  - Hostile environments

- PLCs are frequently programmed using ladder logic
  - This notation was developed to specify logic constructed with relays and switches

- New development: model-driven development of control software for distributed automation via IEC 61499
  - Function Block (FB): similar to actor model
Ladder Logic Basics

Rung 0
- Input Switch (Normally Open)
- Input Switch (Normally Closed)
- S1
- S2
- Output Coil C1

Rung 1
- Normally Open Switch Controlled by Coil C1
- C1
- C2
- Another Output Coil

Power Rail
- Ground Rail
Ladder Logic Memory

Rung 0

Set

Stored

Reset

Stored

Power Rail

Ground Rail
Example

Rung 0
- Start
- Run
- Stop
- Run
- Motor

Rung 1
- Run

Power Rail

Ground Rail
Ladder Logic & Relays

- Early computers used relays to realize logic functions
  - A relay is a switch where the contact is controlled by coil. When a voltage is applied to the coil, the contact closes, enabling current to flow through the relay.
- Early industrial controllers (i.e., early embedded computers) used racks of relays to implement process control logic
- Modern programmable logic controllers are built from microcontrollers
  - Circuits and enclosures are hardened to survive industrial environments
- Ladder logic interfaces smoothed the transition from real relay racks to PLCs.
Types of processors

- Microcontroller
- Digital signal processor (DSP)
- Graphics processor
Digital signal processor (DSP)

- Common characteristics of signal processing applications
  - Deal with large amounts of data, e.g., samples in time and/or space
  - Typically perform sophisticated mathematical operations on the data, including filtering, system identification, frequency analysis, feature extraction, and machine learning

- DSPs designed to support numerically intensive signal processing applications
Embedded Signal Processing

Frequently be described as finite-impulse-response (FIR) filters
FIR (Finite Impulse Response) Filter Structure

\[
v_o(k) = \sum_{m=0}^{N-1} v_i(k-m) h(m)
\]
Multiply – Accumulate

Fast Integer

Large Integer

Together
Central characteristics of DSPs

- Provide a fast and efficient multiply-accumulate (MAC) instruction
  - Typically including a relatively large accumulator
- Typically use a Harvard memory access architecture (to support multiple simultaneous data and program fetches)
- Addressing modes supporting auto increment, circular buffers (for efficient implementation of delay lines), and bit-reversed addressing (for FFT calculation)
- May support zero-overhead loops (i.e., CPU repeated executing an instruction without instruction fetching etc)
Programming DSPs

- DSPs are difficult to program compared to RISC architectures, primarily because of:
  - complex specialized instructions
  - a pipeline that is exposed to the programmer
  - asymmetric memory architectures

- Until the late 1990s, DSPs were almost always programmed in assembly language.
- Even today, C programs make extensive use of libraries that are hand-coded in assembly language to take advantage of the most esoteric features of the architectures
Types of processors

- Microcontroller
- Digital signal processor (DSP)
- Graphics processor
Graphics processors

- A graphics processing unit (GPU) is a specialized processor designed especially to perform calculations required in *graphics rendering*
- Such processors date back to the 1970s

- GPUs have evolved towards more general programming *models*, and hence have started to appear in other compute-intensive applications, such as instrumentation
- GPUs are typically quite *power hungry*, and therefore today are not a good match for energy-constrained embedded applications
Outline

- Types of processors
- Parallelism
Parallelism vs. Concurrency

- Embedded computing applications typically do more than one thing “at a time”

- Tasks are said to be “concurrent” if they conceptually execute simultaneously

- Tasks are said to be “parallel” if they physically execute simultaneously
Imperative Languages and Sequentiality

- An imperative language expresses a computation as a sequence of operations.
- Every correct execution of a program in an imperative language must behave as if the instructions were executed in exactly the specified sequence.
  - This is called sequential consistency.
- It may be possible to improve processor performance by performing some (data-flow-independent) operations in parallel, but the processor must maintain sequential consistency.
Embedded Performance “Improvement”

- Various current architectures seek to improve performance by finding and exploiting potentials for parallel execution
  - Frequently improves processing throughput
  - Does not always improve processing latency
  - Frequently makes processing time less predictable

- Many embedded applications rely on results being produced at predictable regular rates
  - Embedded results must be available at the right time (neither early nor late)
Parallelism

- Temporal Parallelism – Pipelining

- Spatial Parallelism –
  - CISC (complex instruction set computer) instructions
  - Subword parallelism
  - Superscalar
  - VLIW (very large instruction word)
  - Multicore
Pipelining

**hardware resources:**
- instruction memory
- register bank read 1
- register bank read 2
- ALU
- data memory
- register bank write

Sequential consistency vs. data-induced dependency between instructions
CISC instructions

- DSPs are typically CISC machines, and include instructions specifically supporting FIR filtering, and often other algorithms such as FFTs (fast Fourier transforms) and Viterbi decoding
  - *Parallel instruction execution* on homogeneous/heterogeneous data flows
- Disadvantages
  - it is extremely challenging (perhaps impossible) for a compiler to make optimal use of such an instruction set. As a consequence, DSP processors are commonly used with *code libraries* written and optimized in assembly language
  - CISC instruction sets can have subtle timing issues that can interfere with achieving hard real-time scheduling
Subword parallelism

- A wide ALU is divided into narrower slices enabling simultaneous arithmetic or logical operations on smaller words.

- At each time instant, same arithmetic or logical operations but different data flows
  - SIMD: single instruction, multiple data
Superscalar

- Hardware can *simultaneously dispatch multiple instructions to distinct hardware units* when it detects that such simultaneous dispatch will not change the behavior of the program.

- Superscalar processors have a significant disadvantage for embedded systems:
  - execution times may be extremely difficult to predict, and in the context of multitasking (interrupts and threads), may not even be repeatable.
  - execution times may be very sensitive to the exact timing of interrupts, i.e., small variations in such timing may have big effects on the execution times of programs.
VLIW (very large instruction word)

- Often preferred over superscalar in embedded systems
- VLIW processors have multiple function units to support parallel execution, similar to superscalar arch.
- A VLIW instruction set combines multiple independent operations into a single instruction
  - Like superscalar architectures, these multiple operations are executed simultaneously on distinct hardware
  - Unlike superscalar, however, the order and simultaneity of the execution is fixed in the program rather than being decided on-the-fly. It is up to the programmer (working at assembly language level) or the compiler to ensure that the simultaneous operations are indeed independent
- In exchange for additional complexity in programming, execution times become repeatable and (often) predictable
Multicore architecture

- A multicore machine is a combination of several processors on a single chip
- Heterogeneous multicore machines combine a variety of processor types on a single chip (vs. multiple instances of the same processor type)
  - Real-time and safety-critical tasks can have a dedicated processor; this is the reason for the heterogeneous architectures used for cell phones, since the radio and speech processing functions are hard real-time functions with considerable computational load
- Interference tends to be problematic with general-purpose multicore architectures, e.g., multi-level caches shared across cores
A unique multicore architecture

- Use of one or more *soft cores* together with custom hardware on a field-programmable gate array (FPGA)
  - FPGAs are chips whose hardware function is programmable using hardware design tools
  - Soft cores are processors implemented on FPGAs

- Advantage of soft cores is that they *can be tightly coupled to custom hardware* more easily than off-the-shelf processors
Summary

- Types of processors
- Parallelism