Hardware-Software Codesign

1. Introduction

Lothar Thiele
Contents

- What is an Embedded System?
- Levels of Abstraction in Electronic System Design
- Typical Design Flow of Hardware-Software Systems
Embedded Systems

Embedded systems (ES) = information processing systems embedded into a larger product

Examples:

Main reason for buying is not information processing
Embedded Systems

external process

human interface

eMBEDDED SYSTEM

sensors, actuators

human interface

external process

sensors, actuators

human interface
Parallel and Distributed Target Platforms

ACC
ABS
ESP
ASR

engine control
powertrain control
Example: Cell Processor

Cell Processor (IBM) combines
- general-purpose architecture core with
- coprocessing elements which greatly accelerate multimedia and vector processing applications, as well as many other forms of dedicated computation.
Example: Intel

48 cores

4 cores
More Examples

Intel Xeon Phi

Oracle Sparc T5
Example ST2012/STHORM
Multiprocessor systems-on-a-chip (MPSoCs)

### SH-MobileG1: Chip Overview

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Die size</td>
<td>11.15mm x 11.15mm</td>
</tr>
<tr>
<td>Process</td>
<td>90nm LP 8M(7Cu+1Al) CMOS dual-Vth</td>
</tr>
<tr>
<td>Supply voltage</td>
<td>1.2V(internal), 1.8/2.5/3.3V(I/O)</td>
</tr>
<tr>
<td># of TRs, gate, memory</td>
<td>181M TRs, 13.5M Gate, 20.2 Mbit mem</td>
</tr>
</tbody>
</table>

- CPG: Communication Processor Group
- DDR: Double Data Rate
- Sound: Sound Processing
- AP-Misc: Application Miscellaneous
- AP-SYS: Application System
- CPU: Central Processing Unit
- W-CDMA: Wideband Code Division Multiple Access
- BB-Misc: Baseband Miscellaneous
- BB-CPU: Baseband CPU
- 3D G: 3D Graphics
- MPEG: Moving Pictures Expert Group
- JPEG: Joint Photographic Experts Group
- Camera: Image Processing
- LCD: Liquid Crystal Display
- Media: Multimedia
- RAM: Random Access Memory
- APL-RT: Application Platform for Real-Time
- SRAM: Static Random Access Memory
- GSM: Global System for Mobile Communications

Multiprocessor systems-on-a-chip (MPSoCs)
Multiprocessor systems-on-a-chip (MPSoCs)

Samsung Galaxy Note II
- Eynos 4412 System on a Chip (SoC)
- ARM Cortex-A9 processing core
- 32 nanometer: transistor gate width
- Four processing cores
Zero Power Systems and Sensors
Zero Power Systems and Sensors
Comparison

- Embedded Systems
  - Few applications that are known at design-time.
  - Not programmable by end user.
  - Fixed run-time requirements (additional computing power not useful).
  - Criteria:
    - cost
    - power consumption
    - predictability
    - meeting time bounds
    - ...

- General Purpose Computing
  - Broad class of applications.
  - Programmable by end user.
  - Faster is better.
  - Criteria:
    - cost
    - average speed
Design Challenges

**Challenges in the design of embedded systems**

- **increasing application complexity** even in standard and large volume products
  - large systems with legacy functions
  - mixture of event driven and data flow tasks
  - examples: multimedia, automotive, mobile communication

- **increasing target system complexity**
  - mixture of different technologies, processor types, and design styles
  - large systems-on-a-chip combining components from different sources, distributed system implementations

- **numerous constraints and design objectives**
  - examples: cost, power consumption, timing constraints, temperature
Implementation Alternatives

- General-purpose processors
- Application-specific instruction set processors (ASIPs)
  - Microcontroller
  - DSPs (digital signal processors)
- Programmable hardware
  - FPGA (field-programmable gate arrays)
- Application-specific integrated circuits (ASICs)
Contents

- What is an Embedded System?
- *Levels of Abstraction in Electronic System Design*
- Typical Design Flow of Hardware-Software Systems
Abstraction, Models and Synthesis

Model
- Formal description of selected properties of a system or subsystem
- A model consists of data and associated methods

Classification of models
- Degree of abstraction, granularity
  - hardware: system, architecture, logic, transistor,
  - software: module, block/class, function/method, ...
- View
  - behavior, structural, physical

Synthesis
- Linking adjacent levels of abstraction (refinement)
- Stepwise adding of structural information
Levels of Abstractions

- **System**
  - **Process/Module**
    - Functional
    - Object Code
  - **Architecture**
    - Gate-level models
    - Switch-level models
    - Circuit-level models
    - Device-level models
    - Layout models
  - **HW**
  - **RTL**
    - Behavior
    - Structure

**Swiss Federal Institute of Technology**

**Computer Engineering and Networks Laboratory**
Contents

- What is an Embedded System?
- Levels of Abstraction in Electronic System Design
- Typical Design Flow of Hardware-Software Systems
System Design

**Specification**

**System Synthesis**

- **SW-Compilation**
  - Intellectual Prop. Code
  - Machine Code

- **Instruction Set**
  - Instruction Set
  - Net lists

- **HW-Synthesis**
  - Intellectual Prop. Block
  - Net lists

**Estimation**
Fixed Processor Architecture

- Specification
  - System Synthesis
    - SW-Compilation
      - Intellectual Prop. Code
        - Machine Code
    - Instruction Set
    - HW-Synthesis
      - Intellectual Prop. Block
        - Net lists
  - Estimation
Application-Specific Instruction Set Processor

1 - Specification
2 - System Synthesis
3 - Estimation
4 - SW-Compilation
5 - Instruction Set
6 - HW-Synthesis
7 - Intellectual Prop. Code
8 - Machine Code
9 - Net lists
10 - Intellectual Prop. Block
System-Level Design

**System-level design** is a complex synthesis task:
- software synthesis and code generation
- hardware synthesis
- interface and communication synthesis
- hardware/software partitioning and component selection
- hardware/software scheduling

**Major Components:**
- application specification
- design space exploration and system optimization
- estimation
The Mapping Problem
HW/SW Mapping and Scheduling

- **Hardware/software mapping**
  - Partitioning of system function to programmable components (software), hard-wired or parameterized components (hardware) or application specific instruction set processors.

- **Similarity** to scheduling and load distribution problem in real-time operating systems
  - time constraints, context switch and context switch overhead, process synchronization and communication

- **Differences** to real-time operating systems
  - larger design space with very different solutions
  - high optimization requirements (motivation for hardware design)
  - underlying hardware is not fixed
HW/SW Mapping and Scheduling

- Similarity to allocation (or load distribution) problem in high-level synthesis (or real-time operating systems)

Diagram:
- HW components
- SW (processors)
- P1, P2, P3, P4
Estimation

The principle of synthesis based on abstraction only makes sense if there are powerful estimation methods available:

- Estimate properties of the next layer(s) of abstraction.
- Design decisions are based on these estimated properties: If the estimation is not correct (or not accurate enough), the design will be sub-optimal or even not working correctly.