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
Parallel and Distributed Target Platforms

- ACC
- ABS
- ESP
- ASR

engine control

powertrain control

ETH
Swiss Federal Institute of Technology
Computer Engineering and Networks Laboratory
Example: Intel

48 cores

4 cores
More Examples

- Intel Xeon Phi
  (5 Billion transistors,
   22nm technology,
   350mm$^2$ area)

- Oracle Sparc T5
Embedded Multicore Example

Recent development:
- Specialize multicore processors towards real-time processing and low power consumption
- Target domains:

<table>
<thead>
<tr>
<th>Core Generation</th>
<th>Number of Processing Cores</th>
<th>GFLOPS/W</th>
<th>GOPS/W</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andey</td>
<td>256</td>
<td>25</td>
<td>75</td>
</tr>
<tr>
<td>Bostan (2014)</td>
<td>256</td>
<td>50</td>
<td>80</td>
</tr>
<tr>
<td>Coolidge (2015)</td>
<td>64/256/1024</td>
<td>75</td>
<td>115</td>
</tr>
</tbody>
</table>
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,</td>
<td>181M TRs, 13.5M Gate 20.2 Mbit mem</td>
</tr>
<tr>
<td>memory</td>
<td></td>
</tr>
</tbody>
</table>
Multiprocessor systems-on-a-chip (MPSoCs)

G1 Module Diagram
Multiprocessor systems-on-a-chip (MPSoCs)

Samsung Galaxy S6
- Exynos 7420 System on a Chip (SoC)
- 8 ARM Cortex processing cores
  (4 x A57, 4 x A53)
- 30 nanometer: transistor gate width
Multiprocessor systems-on-a-chip (MPSoCs)

Samsung Galaxy S6
- Exynos 7420 System on a Chip (SoC)
- 8 ARM Cortex processing cores (4 x A57, 4 x A53)
- 30 nanometer: transistor gate width

Exynos 5422
- LPDDR3 933MHz DDR 32bit 2-ch, 144GB/s
- 2-ch eMMC 5.0 DDR 100MB/s 200MHz
- 1-ch eMMC 4.5 DDR 200MB/s

Display / Camera
- Single WQXGA 60fps 4-lane eDP
- Single WXGA 60fps: 2-lanes eDP/4-lane MIPI
- HDMI v1.4
- 16MP 30fps ISP, 2-Camera support 14-bit Bayer, 2x 3A, DRC, FD, 3DNR
- 2-ch 4-lane MIPI CSI2: 1.3Gbps 9-Pix

Cortex-A15 Quad
- CPU 0: 2.1GHz, 32KB/32KB
- CPU 1: 2.1GHz, 32KB/32KB

Cortex-A7 Quad
- CPU 2: 1.5GHz, 32KB/32KB
- CPU 3: 1.2GHz, 32KB/32KB

Memory 1/F
- 1GB LPDDR3 1200ps Codec
- VPE Codec
- Mali-T650 series
- JPEG HW codec

Multimedia
- Low Power Multi-layer AXI / AHB Bus
- Low Power Multi-layer AXI / AHB Bus
- Crypto Engine

Peripheral
- 5x UART
- 3x SPI
- 1x TSI
- 2x I2S/PCM
- 1x I2C/PDF
- 7x HS-I2C & 4x I2C

Systems
- Dynamic addressing
- PLLs
- CPU cache coherence
- 24ch DMA
- Memory Interleaving
- PWM/MCT/Timers
- DVS control for Low Power

Computer Engineering
and Networks Laboratory
Zero Power Systems and Sensors
Zero Power Systems and Sensors

IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 48, NO. 1, JANUARY 2013
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 (see next chapter)
    - 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

1. General-purpose processors
2. Application-specific instruction set processors (ASIPs)
   - Microcontroller
   - DSPs (digital signal processors)
3. Programmable hardware
   - FPGA (field-programmable gate arrays)
4. 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
  - Architecture

**Structure**
- Function
- Object Code

**Behavior**
- SW
  - Function
  - Object Code

- HW
  - Architecture
  - RTL

**Levels of Abstractions**
- Gate-level models
- Switch-level models
- Circuit-level models
- Device-level models
- Layout models
Contents

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

1. Specification
2. System Synthesis
3. SW-Compilation
4. Instruction Set
5. HW-Synthesis
6. Estimation
8. Machine Code
10. Net lists
Fixed Processor Architecture

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

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

Specification

System Synthesis

Estimation

SW-Compilation

Instruction Set

HW-Synthesis

Intellectual Prop. Code

Machine Code

Net lists

Intellectual Prop. Block
The System Design Problem

Relating the problem level with the implementation level

Heterogeneous, problem-level description

Heterogeneous, implementation-level description

Modeling

Synthesis
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 showing HW/SW mapping and scheduling](image)
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.
Review: Course Synopsis

- **Specification and Models of Computation (Section 2)**
  - State-Charts
  - Kahn Process Networks

- **System Design**
  - Mapping (Section 3)
  - Partitioning (Section 4)
  - Multi-Criteria Optimization (Section 5)
  - Design Space Exploration (Section 7)

- **Estimation**
  - Simulation-based Methods (Section 6)
  - Performance Estimation (Section 8)
  - Worst-Case Execution Time Analysis (Section 9)
  - Performance Analysis of Distributed Systems (Section 10)
  - Thermal-aware Design (Section 11)