THE UNIVERSITY *of* EDINBURGH
informatics

Embedded Systems
Lecture 1: Introduction

Björn Franke
University of Edinburgh

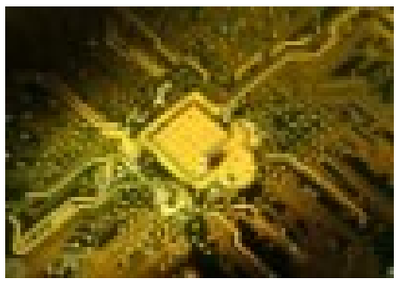
lfcs | Laboratory for Foundations of Computer Science

cisa | Centre for Intelligent Systems and their Applications

```
lemma resPres:
  fixes P :: pi
  and   Q :: pi
  and   x :: name

  assumes PBiSimQ: "P ~ Q"

  shows "<νx>P ~ <νx>Q"
proof -
  let ?X = "{x. ∃P Q. P ~ Q ∧ (∃a. x = (<νa>P, <νa>Q))}"
  from PBiSimQ have "(<νx>P, <νx>Q) ∈ ?X" by blast
  moreover have "⋀P Q a. P ↝[bisim] Q ⟹ <νa>P ↝[(?X ∪ bisim)] <νa>Q"
  proof -
    fix P Q a
    assume PSimQ: "P ↝[bisim] Q"
    moreover have "⋀P Q a. P ~ Q ⟹ (<νa>P, <νa>Q) ∈ ?X ∪ bisim" by blast
    moreover have "bisim ⊆ ?X ∪ bisim" by blast
    moreover have "eqvt bisim" by(rule eqvt)
    moreover have "eqvt (?X ∪ bisim)"
      by(auto simp add: eqvt_def dest: eqvtI)+
    ultimately show "<νa>P ↝[(?X ∪ bisim)] <νa>Q"
      by(rule strongLateSimPres.resPres)
  qed

  ultimately show ?thesis using PBiSimQ
    by(coinduct, blast dest: unfoldE)
qed
```

icsa | Institute for Computing Systems Architecture

epcc



Program

Specify

Design

PhD in Pervasive Parallelism

Verify

Optimize

Apply

http://pervasiveparallelism.inf.ed.ac.uk

intel

Agilent Technologies

Microsoft

ALTRAN

criticalblue
Accelerating Embedded Software

QUALCOMM

IBM

codeplay

ACE

SAMSUNG

ARM

freescale™

SYNOPSYS

wolfson microelectronics

ORACLE

amazon

Lecturers & Teaching Assistant

- Björn Franke
  bfranke@inf.ed.ac.uk

- Michael O'Boyle
  mob@inf.ed.ac.uk

- Stan Manilov
  s.z.manilov@sms.ed.ac.uk

Overview

- Definitions, Motivation
- Examples of Embedded Systems
- Characteristics of Embedded Systems
- Course Overview
- Coursework

Definition of an Embedded System

- “Embedded Systems are information processing systems embedded into a larger product”
(Peter Marwedel, TU Dortmund)

- “Embedded software is software integrated with physical processes. The technical problem is managing time and concurrency in computational systems.”
(Edward Lee, Berkeley)

- “Cyber-Physical (cy-phy) Systems (CPS) are integrations of computation with physical processes”
(Edward Lee, Berkeley)

- *Cyber-physical system* (CPS) =
Embedded System (ES) + *physical environment*

Example of an Embedded System

- Automotive electronics
  - ABS: Anti-lock braking systems
  - ESP: Electronic stability control
  - Airbags
  - Efficient automatic gearboxes
  - Theft prevention with smart keys
  - Blind-angle alert systems
  - In-car entertainment systems
  - ... etc ...

- Multiple networks
- Multiple networked processors

Another Example

- Avionics
  - Flight control systems,
  - anti-collision systems,
  - pilot information systems,
  - power supply system,
  - flap control system,
  - entertainment system,
  - …

Dependability is of outmost importance.

Motivation for Studying Embedded Systems

- Trend in Information Processing Systems towards
  - Ubiquitous computing, Pervasive computing, Ambient intelligence
  - Post-PC era
- Requires holistic approach involving embedded software, embedded hardware and physical environment
- Additional constraints and challenges:
  Power/Energy, Cost, Dependability, Real-Time Processing, ...
- Underrepresented in teaching

Importance of Embedded Systems

- $6bn embedded processors market in 2012,
  12-15% growth in the next two years

- 49.7% of Americans own smartphones
  [www.itfacts.biz, March 31, 2012]

- Average car has about 15 microprocessors in it. S-class has 63 microprocessors; a 1999 BMW 7-series has 65 [Microprocessor Report 2009]

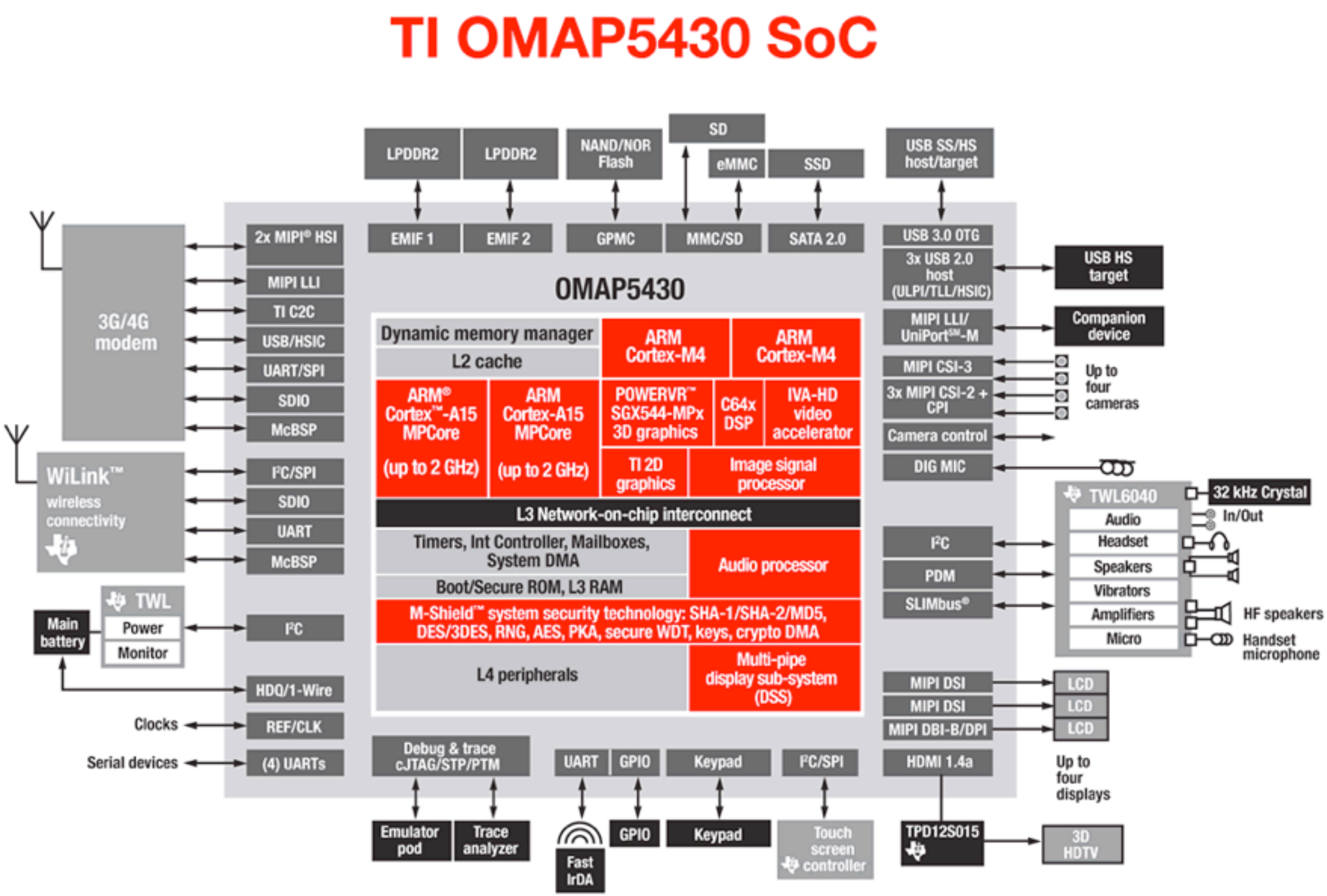
- Average middle-class household has about 40 to 50 microprocessors in it [Microprocessor Report 2009]

- …, the market for remote home health monitoring is expected to generate $225 mln revenue in 2011, up from less than $70 mln in 2006, according to Parks Associates. [www.itfacts.biz, Sep. 4th, 2007]

Embedded Hardware

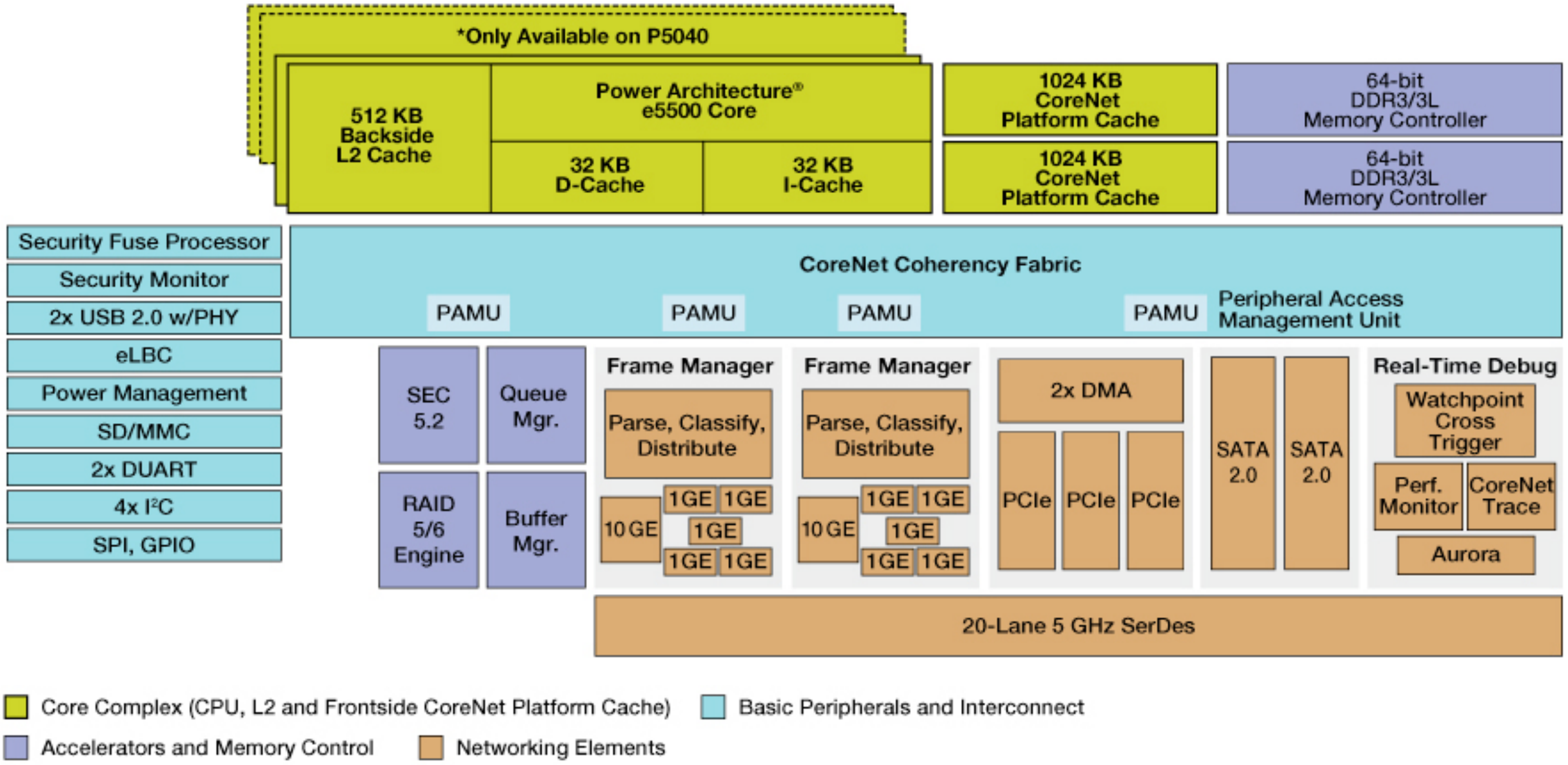
- **Domain/application-specific:** Optimised for one fixed domain/application
- **Energy-efficiency** often more important than raw performance, especially for battery operated devices
- **Power constraints**: Cooling, power supply, ...
- **Cost**: Low cost for large volume device vs Non-recurring engineering cost
- **Programmability:** ASIC (no flexibility), ASIP, CPU, FPGA (lots of flexibility)
- **Design Complexity:** Composed of individual building blocks (IP blocks)

Example


TI OMAP5430 SoC
LPDDR2
LPDDR2
NAND/NOR Flash
SD
eMMC
SSD
USB SS/HS host/target
2x MIPI® HSI
EMIF 1
EMIF 2
GPMC
MMC/SD
SATA 2.0
USB 3.0 OTG
OMAP5430
3x USB 2.0 host (ULPI/TLL/HSIC)
USB HS target
MIPI LLI
TI C2C
USB/HSIC
UART/SPI
SDIO
McBSP
3G/4G modem
Dynamic memory manager
L2 cache
ARM Cortex-M4
ARM Cortex-M4
ARM® Cortex™-A15 MPCore (up to 2 GHz)
ARM Cortex-A15 MPCore (up to 2 GHz)
POWERVR™ SGX544-MPx 3D graphics
C64x DSP
IVA-HD video accelerator
TI 2D graphics
Image signal processor
L3 Network-on-chip interconnect
Timers, Int Controller, Mailboxes, System DMA
Audio processor
Boot/Secure ROM, L3 RAM
M-Shield™ system security technology: SHA-1/SHA-2/MD5, DES/3DES, RNG, AES, PKA, secure WDT, keys, crypto DMA
L4 peripherals
Multi-pipe display sub-system (DSS)
MIPI LLI/ UniPort™-M
Companion device
MIPI CSI-3
3x MIPI CSI-2 + CPI
Up to four cameras
Camera control
DIG MIC
WiLink™ wireless connectivity
I²C/SPI
SDIO
UART
McBSP
TWL6040
32 kHz Crystal
In/Out
Audio
Headset
Speakers
Vibrators
Amplifiers
Micro
HF speakers
Handset microphone
I²C
PDM
SLIMbus®
TWL
Power
Monitor
Main battery
I²C
HDQ/1-Wire
Clocks
REF/CLK
Serial devices
(4) UARTs
MIPI DSI
MIPI DSI
MIPI DBI-B/DPI
LCD
LCD
LCD
Up to four displays
HDMI 1.4a
Debug & trace cJTAG/STP/PTM
UART
GPIO
Keypad
I²C/SPI
Emulator pod
Trace analyzer
Fast IrDA
GPIO
Keypad
Touch screen controller
TPD12S015
3D HDTV

Another Example

QorIQ P5040/P5021 Processors



Embedded Software

- **Real-time:** Timing constraints set by physical environment
- **Reactive:** Response to physical environment
- **Concurrency:** Physical environment is not sequential
- **Dependability:** Impact on physical environment, safety-critical
- **Reliability:** Fixing bugs in the field may be costly/impossible
- **Efficiency:** Manual optimisation required
- **(Lack of) Abstraction:** Exposure of underlying hardware to the programmer

Preliminary Course Overview

1. Introduction
2. Interfacing with the Environment
3. Coursework Session
4. Models of Computation 1 & 2
5. Imperative Programming Languages
6. Embedded Hardware
7. Power/Energy/Faults
8. Scheduling Theory
9. Real-Time Operating Systems
10. Guest Lecture
11. Worst-Case Execution Time
12. Mapping & Scheduling for Multi-Core 1
13. Mapping & Scheduling for Multi-Core 2
14. Mapping & Scheduling for Multi-Core 3
15. HW & SW Optimisations 1
16. HW & SW Optimisations 2
17. Dynamic Voltage Scaling/Dynamic Frequency Scaling
18. Revision

Coursework Overview

- Two parts of individual coursework
- Accompanied by lab sessions (and demonstrator support)
- Coursework 25% of total course mark
- 50/50 split of marks
- Networked home alarm system
- Freescale Kinetis K70 Tower Module (ARM Cortex-M4)

Textbook and Course Website

- Recommended textbook:
  Peter Marwedel
  “Embedded System Design”
  2nd Edition, Springer Verlag, 2011
  ISBN 13 978 94 007 0256 1

- Other textbooks:
  Alan Shaw, Real-Time Systems and Software, John Wiley & Sons
  Alan Burns & Andy Wellings, Real-Time Systems & Programming Languages,
  Addison Wesley.

- Course website:
  www.inf.ed.ac.uk/teaching/courses/es

Summary

- Examples of Embedded Systems
- Embedded Hardware and Software
- Course Overview

Preview

- Next Lecture: Interfacing with the Environment
- Input (Sensors), Output (Actors)
- Analog-Digital Conversion, Digital-Analog Conversion

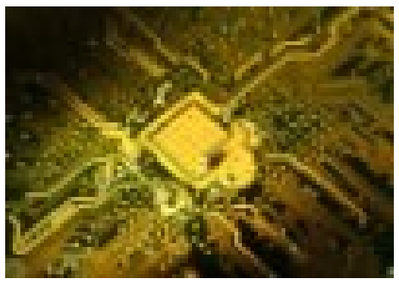
lfcs | Laboratory for Foundations of Computer Science

cisa | Centre for Intelligent Systems and their Applications

```
lemma resPres:
  fixes P :: pi
  and   Q :: pi
  and   x :: name

  assumes PBiSimQ: "P ~ Q"

  shows "<νx>P ~ <νx>Q"
proof -
  let ?X = "{x. ∃P Q. P ~ Q ∧ (∃a. x = (<νa>P, <νa>Q))}"
  from PBiSimQ have "(<νx>P, <νx>Q) ∈ ?X" by blast
  moreover have "⋀P Q a. P ↝[bisim] Q ⟹ <νa>P ↝[(?X ∪ bisim)] <νa>Q"
  proof -
    fix P Q a
    assume PSimQ: "P ↝[bisim] Q"
    moreover have "⋀P Q a. P ~ Q ⟹ (<νa>P, <νa>Q) ∈ ?X ∪ bisim" by blast
    moreover have "bisim ⊆ ?X ∪ bisim" by blast
    moreover have "eqvt bisim" by(rule eqvt)
    moreover have "eqvt (?X ∪ bisim)"
      by(auto simp add: eqvt_def dest: eqvtI)+
    ultimately show "<νa>P ↝[(?X ∪ bisim)] <νa>Q"
      by(rule strongLateSimPres.resPres)
  qed

  ultimately show ?thesis using PBiSimQ
    by(coinduct, blast dest: unfoldE)
qed
```

icsa | Institute for Computing Systems Architecture

epcc



Program

Specify

Design

PhD in Pervasive Parallelism

Verify

Optimize

Apply

http://pervasiveparallelism.inf.ed.ac.uk

intel

Agilent Technologies

Microsoft

ALTRAN

criticalblue
Accelerating Embedded Software

QUALCOMM

IBM

codeplay

ACE

SAMSUNG

ARM

freescale™

SYNOPSYS

wolfson microelectronics

ORACLE

amazon