ARTIST2 Embedded Systems Design
http://www.artist-embedded.org/FP6/

ARTIST Workshop at DATE'06
W4: "Design Issues in Distributed, Communication-Centric Systems"

Information Society Technologies

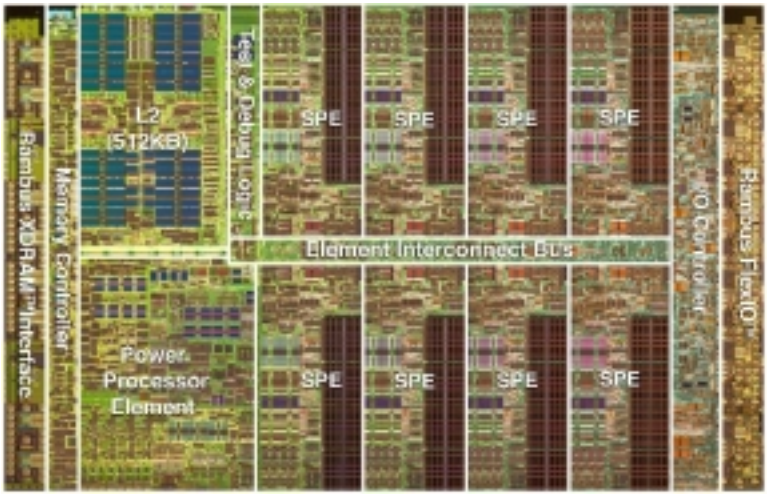
Modelling Networked Embedded Systems: From MPSoC to Sensor Networks

Jan Madsen
Technical University of Denmark

ARTIST2 Embedded Systems Design

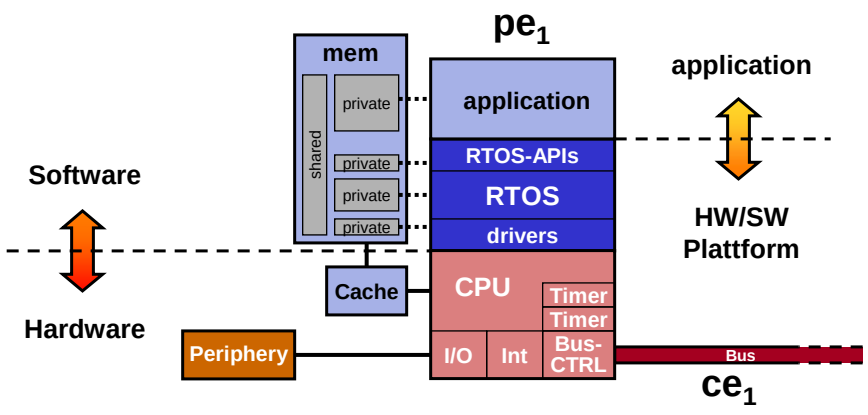
Multiprocessor System-on-Chip

The CELL processor



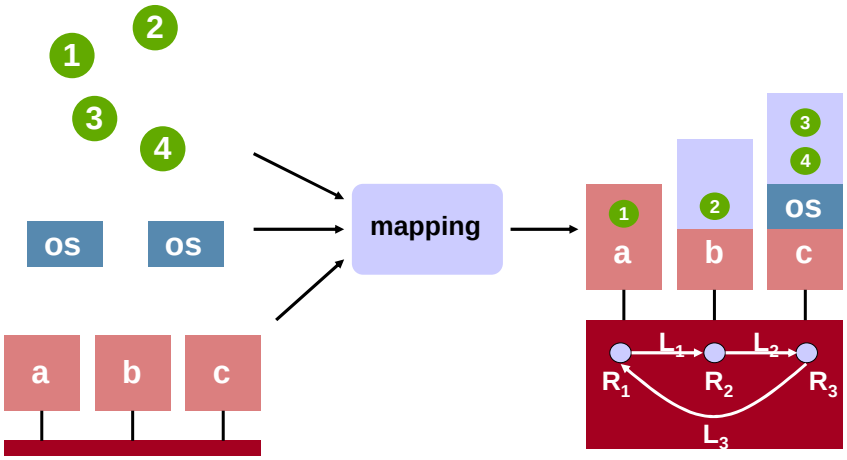
Courtesy of Dac Pam, IBM

Software/Hardware architecture



Courtesy of Rolf Ernst, TU Braunschweigh

Platform-based design



Outline

- The **ARTS** framework
  - basic model
- Modelling multiprocessor System-on-Chip
  - Simple multi media terminal
- Modelling wirekess sensor networks
  - Simple examples
- Summary

ARTS objectives

- System-level modeling framework
- Bridging,
  - Application
  - RTOS
  - Execution platform
    - *Processing elements*
    - *NoC*

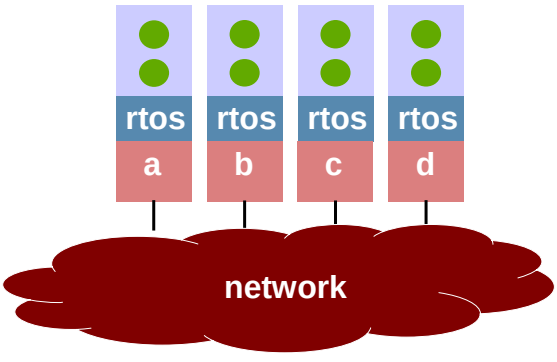
  } ***Cross-layer optimization***

- Supporting
  - System-level analysis
  - Early design space exploration

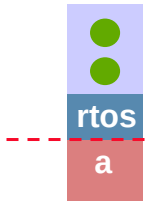
System-level analysis

- Consequences of
  - **RTOS** selection
    scheduling, synchronization and resource allocation policies
  - **Processor** selection
    General purpose, semicustom or dedicated hardware
  - **Network** selection
    topologies and communication protocols.
  - **Task mapping** to processors
    software or hardware
- On
  - Performance
  - Area
  - Memory profile
  - Power
  - …

Basic ARTS model



Uni-processor …



Framework to experiment with different RTOS strategies

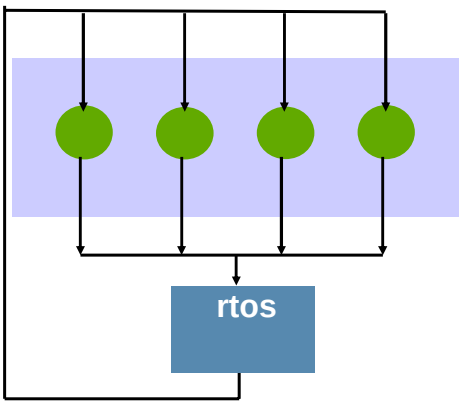
Focus on analysis of **timing** and resource sharing

Abstract software model, i.e. no behavior/functionality

Easy to create tasks and implement RTOS models

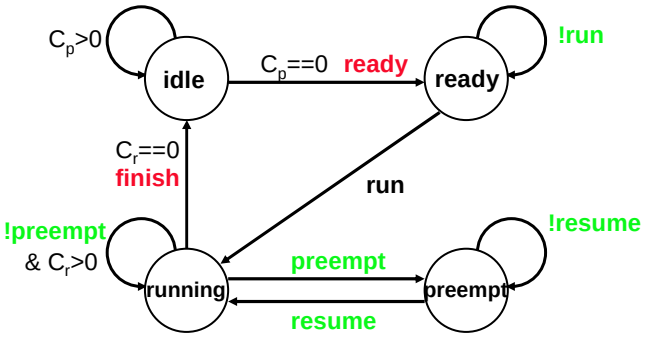
Based on **SystemC**

System model

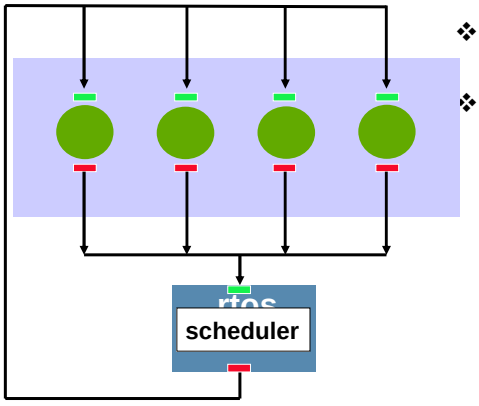


- Task messages:
  - `ready`
  - `finished`
- RTOS commands:
  - `run`
  - `preempt`
  - `Resume`

Task model

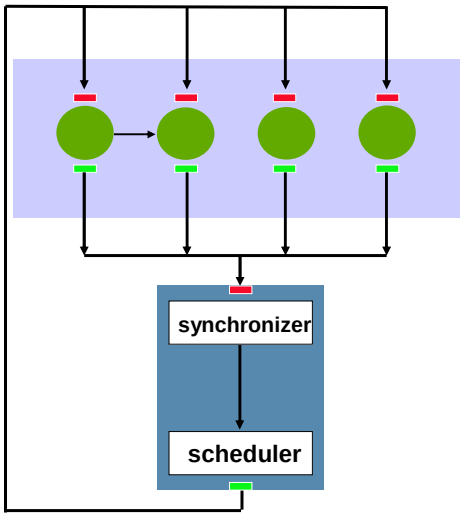


Scheduling model

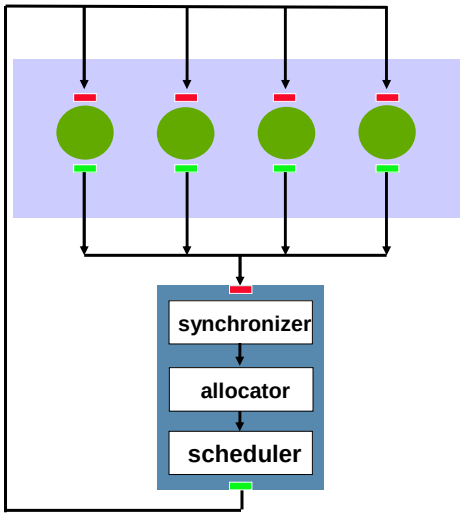


- Aim: Simple way to describe the scheduling algorithm
- Scheduler should only handle one message at a time
- Schedulers:
  - RM
  - DM
  - EDF
  - static

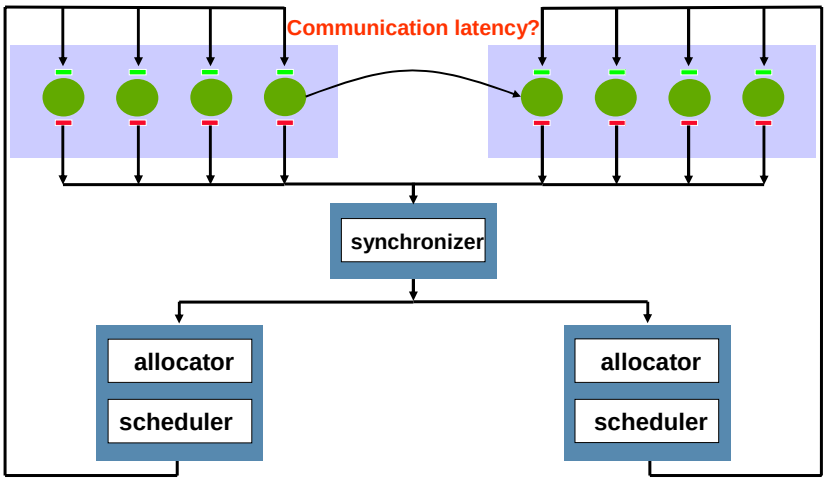
Data dependencies



Resource sharing

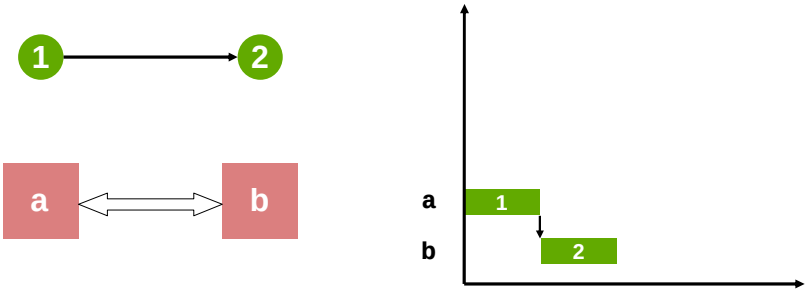


Multi-processors



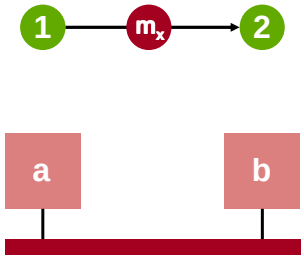
Communication modelling

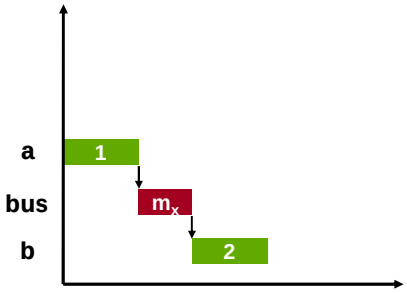
Fully interconnected or point-to-point
Allows concurrent communication



Communication modelling

Bus-based
Shared resource

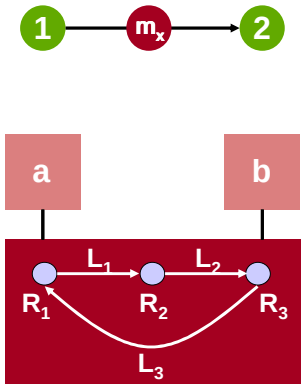


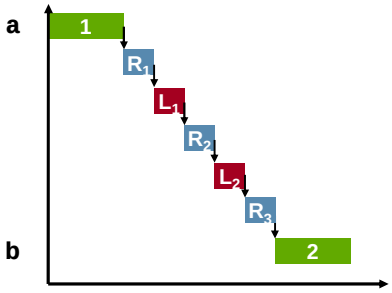


ARTIST2 Embedded Systems Design

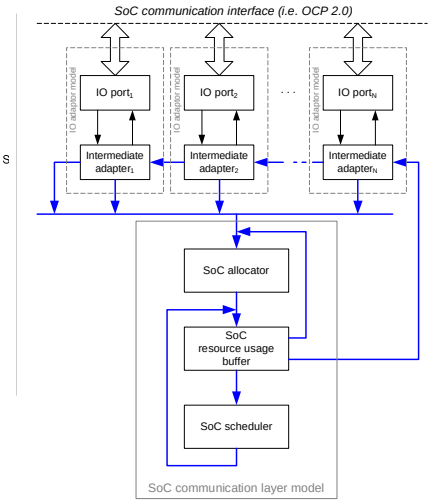
Communication modelling

Torus network
Concurrent, shared and multi-hop





ARTS Framework

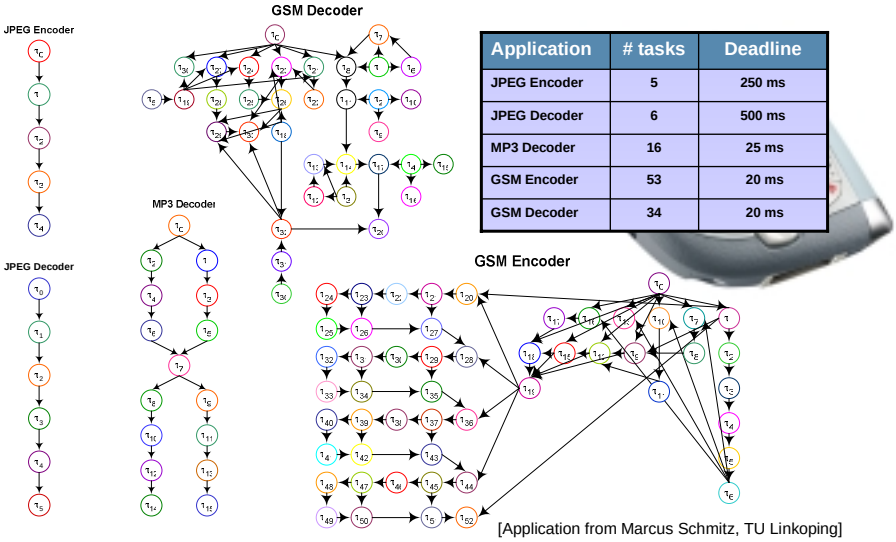


- **ARTS** Simulation framework based on SystemC
- **ARTS** PE module:
  - Application
  - OS
  - IO ports (OCP 2.0 interface)
  - IO device drivers
- **ARTS** Communication module:
  - Network topology and protocol
  - Network adapters
  - IO ports (OCP 2.0 interface)
- Applications of **ARTS**:
  - MPSoC (NoC exploration)
  - Wireless sensor networks
  - Automotive systems (TT vs. ET)
  - Dynamic reconfiguration

Outline

- The **ARTS** framework
  - basic model
- Modelling multiprocessor System-on-Chip
  - Simple multi media terminal
- Modelling wirekess sensor networks
  - Simple examples
- Summary

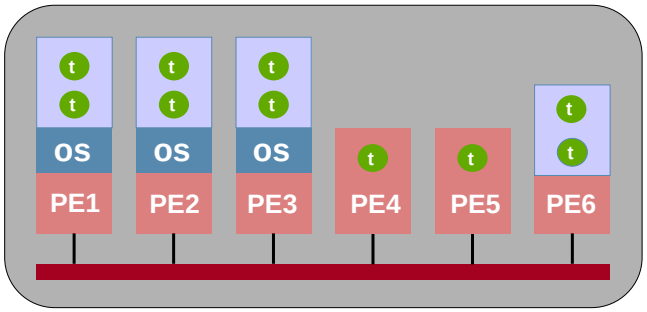
Design of a hand-held multimedia terminal



Application	# tasks	Deadline
JPEG Encoder	5	250 ms
JPEG Decoder	6	500 ms
MP3 Decoder	16	25 ms
GSM Encoder	53	20 ms
GSM Decoder	34	20 ms

[Application from Marcus Schmitz, TU Linkoping]

Case study: Execution platform



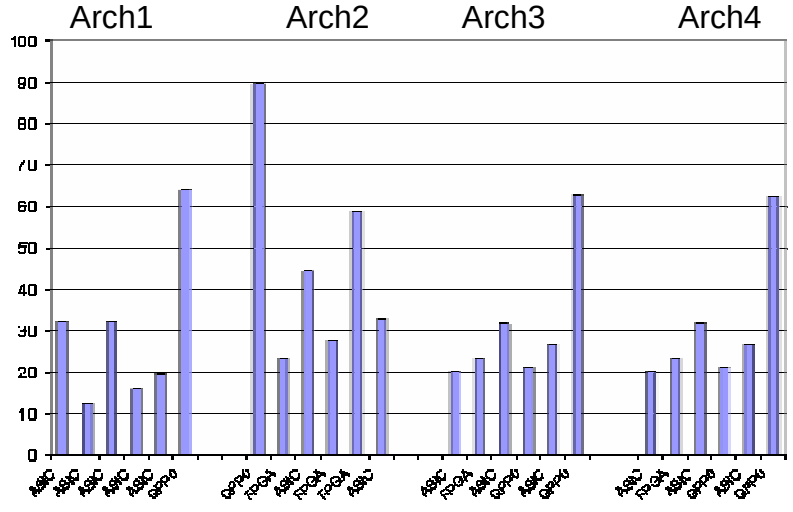
PE	GPP0	GPP1	GPP2	FPGA	ASIC
Frequency (MHz)	25	10	6.6	2.5	2.5

OS	RM (Rate Monotonic)	EDF (Early Deadline First)

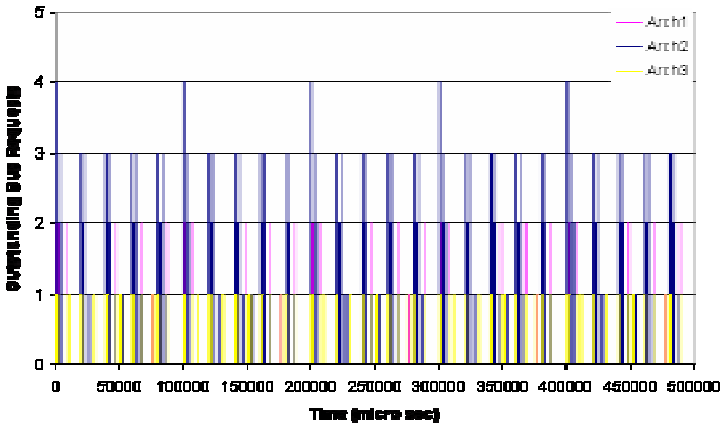
Case study: Architectures

	PE1	PE2	PE3	PE4	PE5	PE6
Arch1						
IP Type	ASIC	ASIC	ASIC	ASIC	ASIC	GPP0
OS	-	-	-	-	-	RM
Tasks	21	15	15	18	13	32
Arch2						
IP Type	GPP0	FPGA	ASIC	FPGA	FPGA	ASIC
OS	RM	RM	-	RM	RM	-
Tasks	18	15	21	18	18	24
Arch3						
IP Type	ASIC	FPGA	ASIC	GPP0	ASIC	GPP0
OS	-	RM	-	RM	-	RM
Tasks	15	15	15	18	19	32
Arch4						
IP Type	ASIC	FPGA	ASIC	GPP0	ASIC	GPP0
OS	-	EDF	-	EDF	-	EDF
Tasks	15	15	15	18	19	32

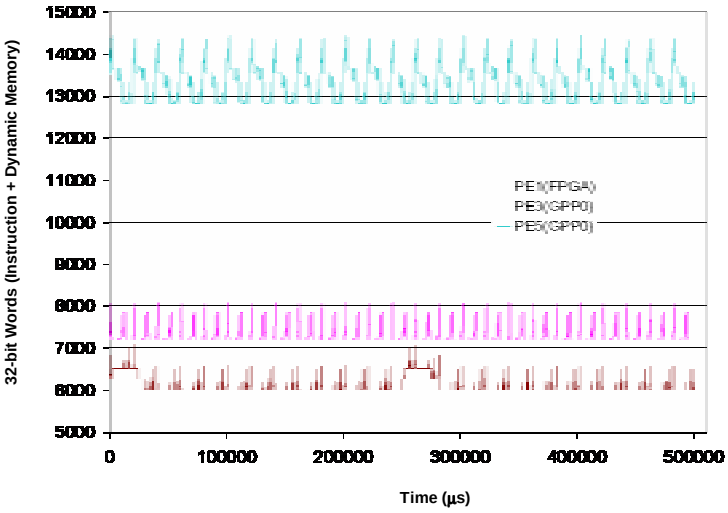
Processor Utilization



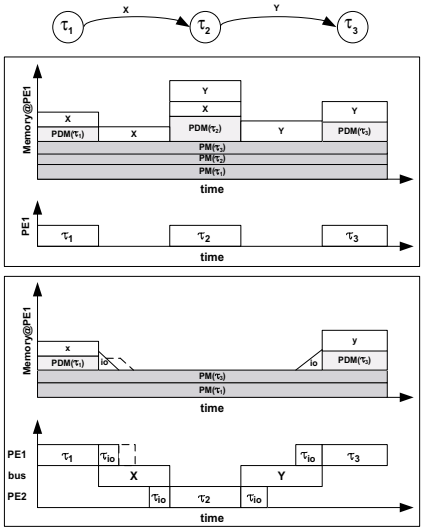
Bus Contention



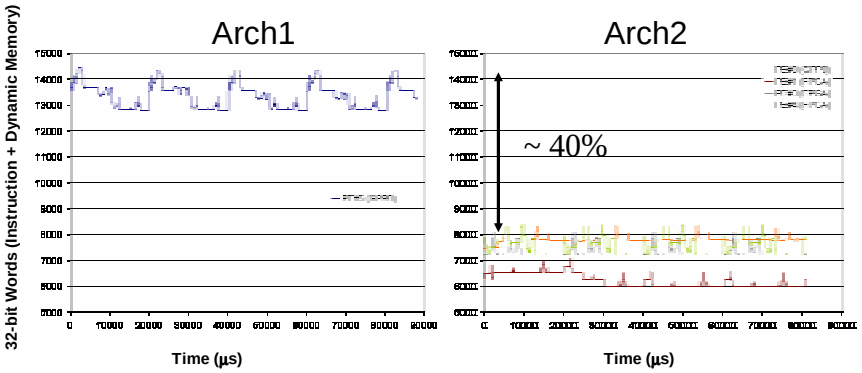
Memory Profile of Arch3



Simple memory model



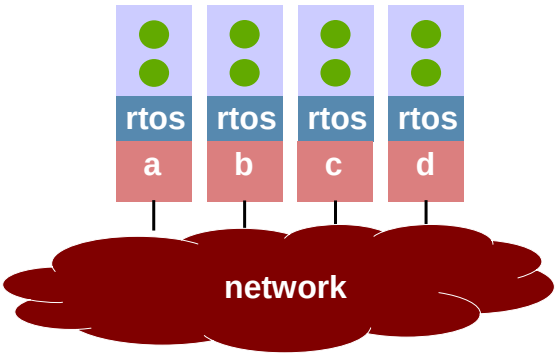
Memory Profile



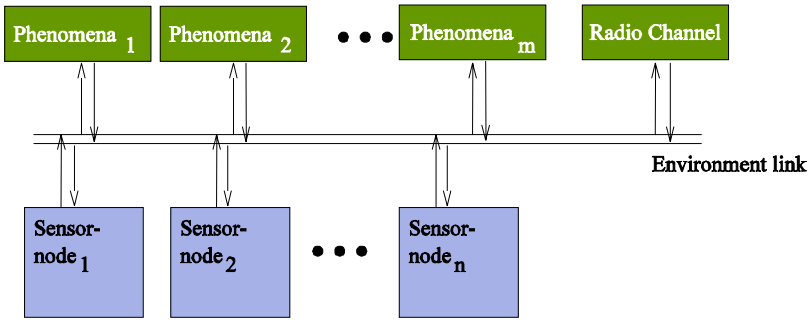
Outline

- The **ARTS** framework
  - basic model
- Modelling multiprocessor System-on-Chip
  - Simple multi media terminal
- Modelling wirekess sensor networks
  - Simple examples
- Summary

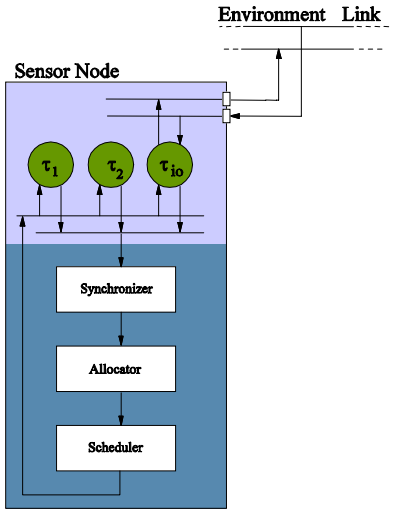
Modelling wireless sensor networks



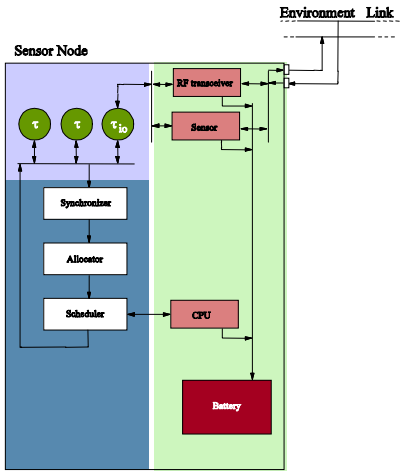
Sensor network model



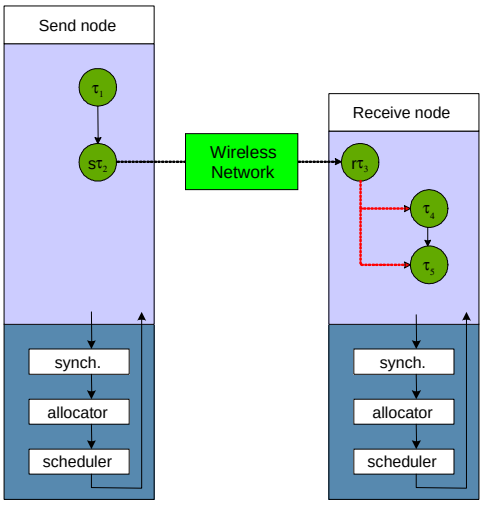
Sensor node model



Energy modeling

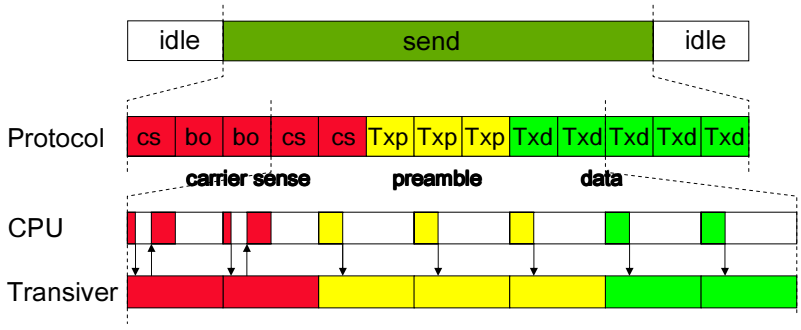


Communication example

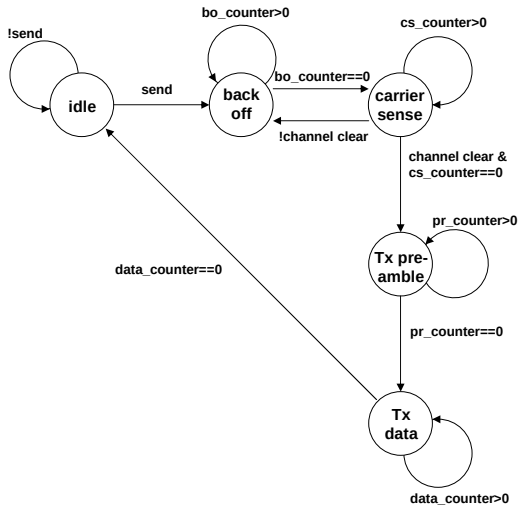


Modeling radio communication

Modeling the CSMA protocol
Sender:

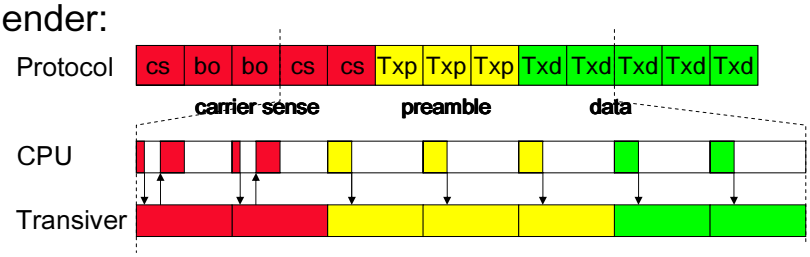


CSMA Protocol for sending

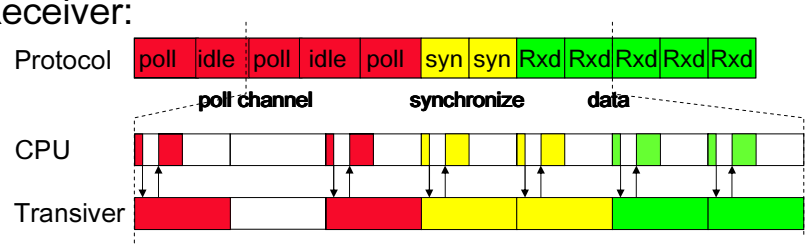


Modeling radio communication

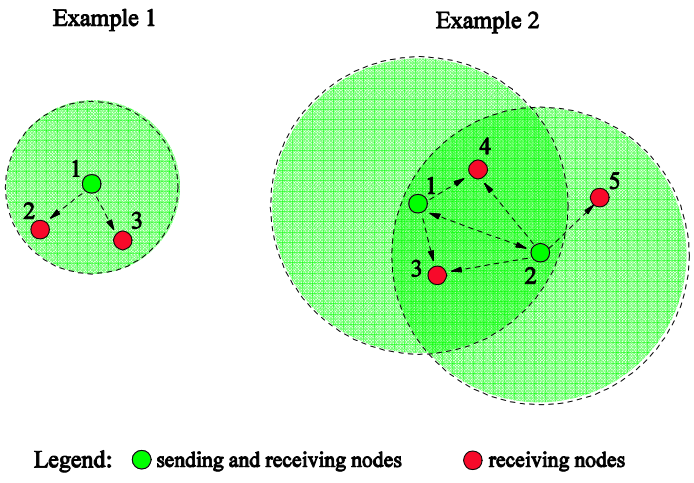
Sender:



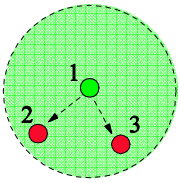
Receiver:



Sensor network example



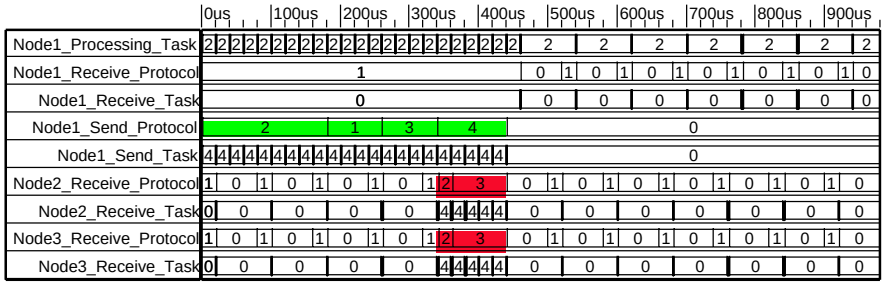
Example 1: Simple broadcast



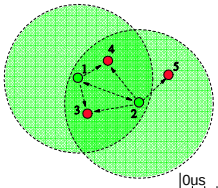
Sending task
0 = idle
1 = back-Off
2 = carrier sensing
3 = transmit preamble
4 = transmit data

Receiving task
0 = idle
1 = polling
2 = synchronize
3 = receive data

Application task
0 = idle
1 = ready
2 = running
3 = preempted
4 = self-preempted



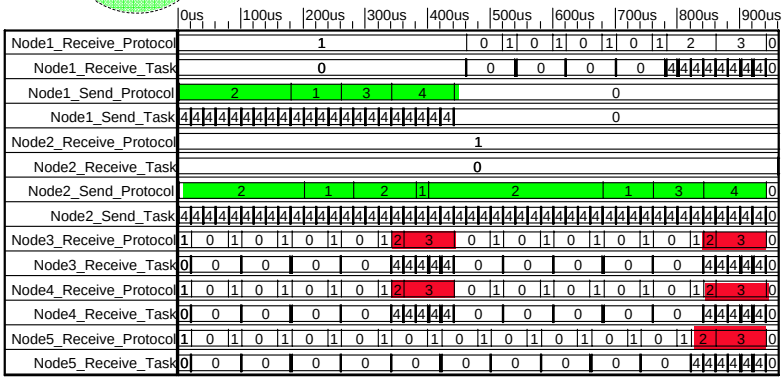
Example 2: Radio interference



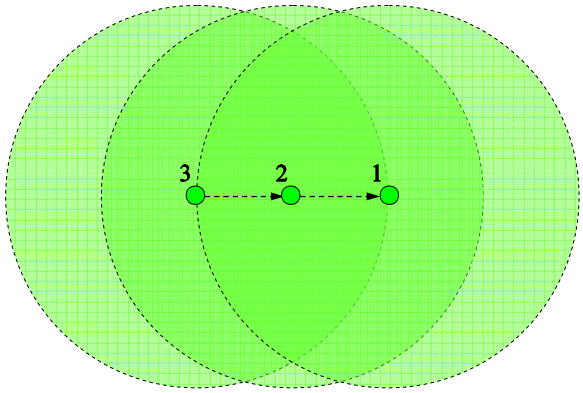
Sending task
0 = idle
1 = back-Off
2 = carrier sensing
3 = transmit preamble
4 = transmit data

Receiving task
0 = idle
1 = polling
2 = synchronize
3 = receive data

Application task
0 = idle
1 = ready
2 = running
3 = preempted
4 = self-preempted

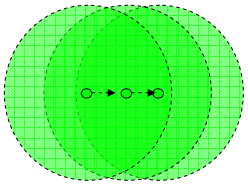


Example 3: Network routing



Legend: sending and receiving nodes receiving nodes

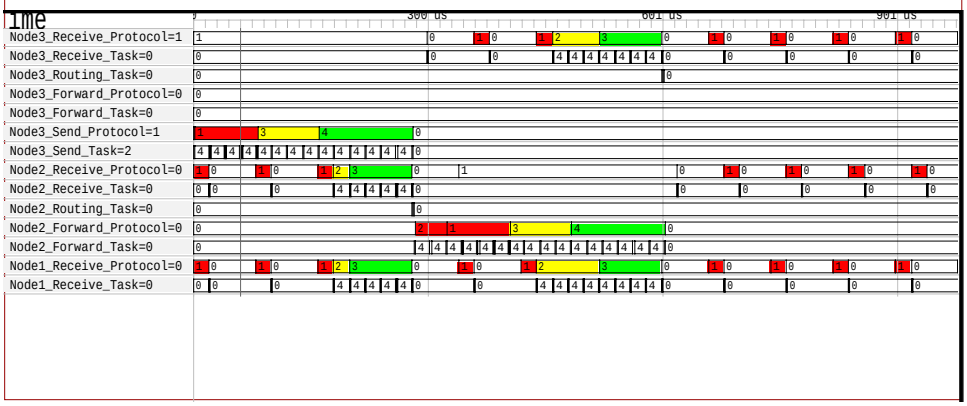
Example 3: Routing



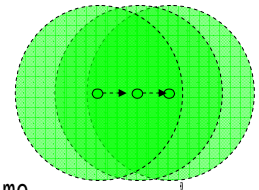
Sending task
0 = idle
1 = back-Off
2 = carrier sensing
3 = transmit preamble
4 = transmit data

Receiving task
0 = idle
1 = polling
2 = synchronize
3 = receive data

Application task
0 = idle
1 = ready
2 = running
3 = preempted
4 = self-preempted



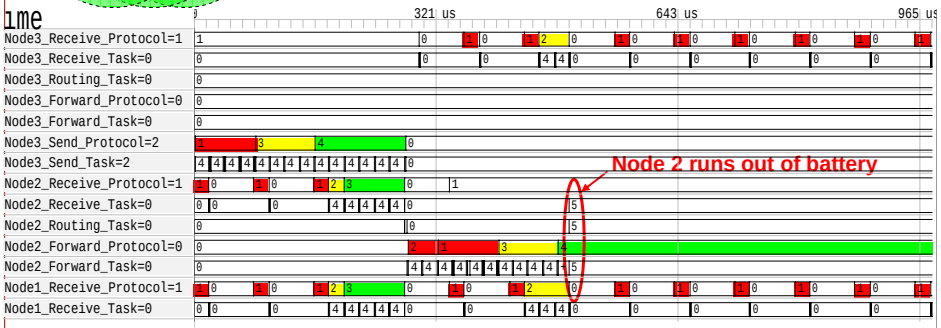
Example 3: Battery shortage



Sending task
0 = idle
1 = back-Off
2 = carrier sensing
3 = transmit preamble
4 = transmit data

Receiving task
0 = idle
1 = polling
2 = synchronize
3 = receive data

Application task
0 = idle
1 = ready
2 = running
3 = preempted
4 = self-preempted



Summary

- System-level modeling framework
- Bridging,
  - Application
  - RTOS
  - Execution platform
    - *Processing elements*
    - *NoC*
- Supporting
  - System-level analysis
  - Early design space exploration
- Applications
  - MPSoC
  - Wireless Sensor Networks

Summary

- Work in progress
  - Mixed-level simulation together with **MPARM** from University of Bologna (Luca Benini)
  - Linking to formal modelling based on **UppAal** from University of Aalborg (Kim G. Larsen)
  - Mixed TT and ET communication models and hierarchical schedulers together with Technical University of Linkoping (Petru Eles)
- To be used in the **Hogthrob project**
  - Thursdays Keynote on Wireless Sensor Networks

Acknowledgements

- ARTS
  - Shankar Mahadevan (PhD student)
  - Kashif Virk (PhD student)
  - Michael Storgaard (MSc. student)
  - Knud Hansen (MSc. student)
  - Mercury Gonzalez (MSc. student)