Smart Energy Management and Low-Power Design of Sensor and Actuator Nodes on Algorithmic Level for Self-Powered Sensorial Materials and Robotics

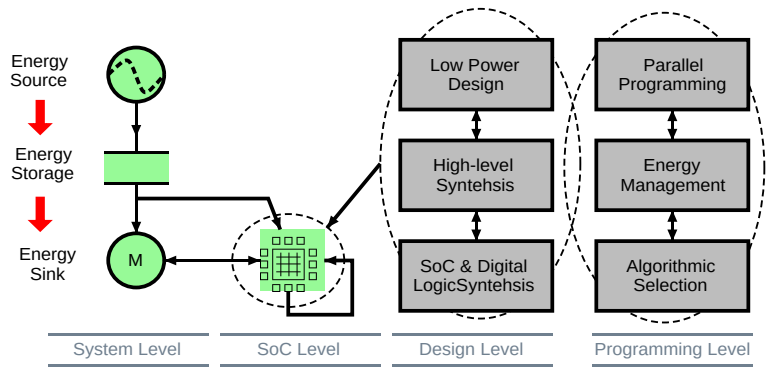
Stefan Bosse
University of Bremen, Department of Computer Science, Workgroup Robotics, Germany(1), ISIS Sensorial Materials Scientific Centre, Germany(2)

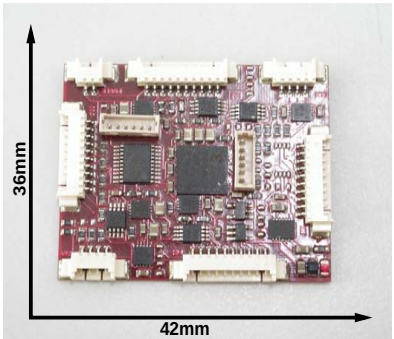
20.4.2011

Overview

Low-Power System Design and Energy Management on Algorithmic Level

1. Self-powered data processing systems for sensorial materials and robotics
2. **Low-power design**: Application-specific System-On-Chip design on algorithmic level using high-level synthesis
3. **Correlation of algorithmic complexity** and energy: Analysis of synthesized microchip activity, estimation of energy demand for computation
4. Technological extensions required for power reduction on microchip level
5. **Energy management** at runtime using **algorithmic selection** and dedicated methods from artificial intelligence and mapping to microchips
6. Prelimenary simulation results





Reference: S. Bosse, T. Behrmann, Smart Energy Management and Low-Power Design of Sensor and Actuator Nodes on Algorithmic Level for Self-Powered Sensorial Materials and Robotics, Proceedings of the SPIE Microtechnologies 2011 Conference, 18.4.-20.4.2011, Prague, Session EMT 101 Smart Sensors, Actuators and MEMS

Self-powered data processing systems used in ...

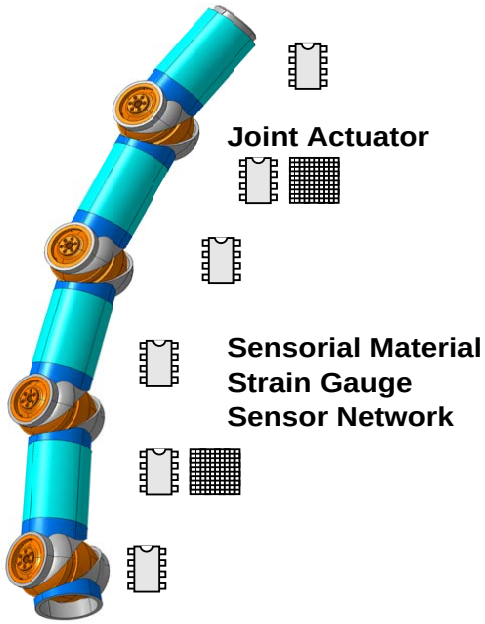
Cyber-Physical-Systems

- Defined by the interaction of the system with its environment
- Tight integration of computation and control with sensing and actuation physical components
- System components: sensors, actuators, data processing, communication ➠ application specific
- CPS must be reliable, adaptable, easy-to-use, and low-power
- **Operation defined on algorithmic level - requires concurrency**

Sensorial Materials

- Network of smart sensor nodes
- Sensor node: sensor, electronics, and data processing
- SM must be reliable, adaptable, highly minaturized, and low-power

Figure 1. ModuACT robot arm manipulator with network of sensorial materials and actuator joints



ConPro: Programming Language & Highlevel-Synthesis

Synthesis of parallel application-specific System-On-Chip designs on programming level targeting FPGA and ASIC technologies

Programming Model & Language

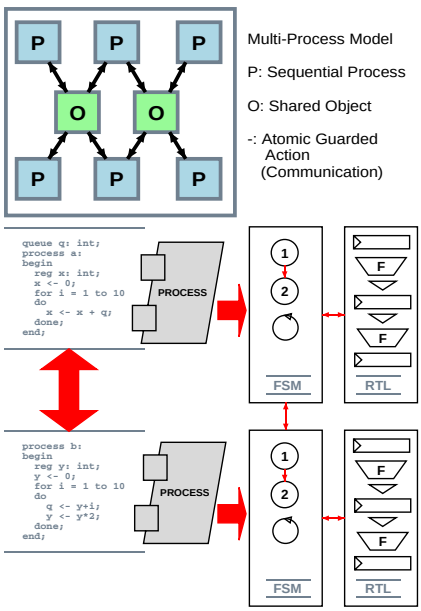
- Communicating Sequential Processes
- Imperative Language with explicitly modelled parallelism
- Guarded shared objects (atomic guarded access)
- Object oriantated access of hardware blocks
- Concurrency on control- and data-path level
- Synchronization and Interprocess-Communication ➠ directly implementable in hardware

Execution Model

- Process: strict sequential
- Mapped to Finite-State-Machine &

Register-Transfer Logic

Figure 2. Multi-Process Model [mod. CSP/Hoare]



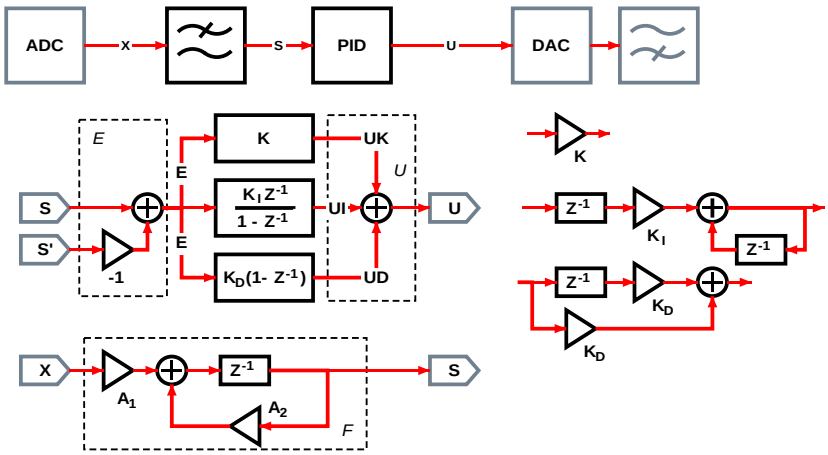
Modelling of Data Processing Systems

Modelling of parallel data processing systems using signal flow diagrams is an entry level for high-level synthesis on programming level

Signal Flow Diagrams

- Data processing is performed by using operational blocks on a sequential flow of discrete data sets
- Operational blocks: low level (simple arithmetic and relational operations), high-level (complex computation blocks), storage (delay)

Example 1. A PID controller modelled with signal flow diagrams



Mapping of Signal Flow Graphs to Petri-Nets

The signal flow graph is mapped to an intermediate representation to derive communication architecture, initial setup, and to explore concurrency

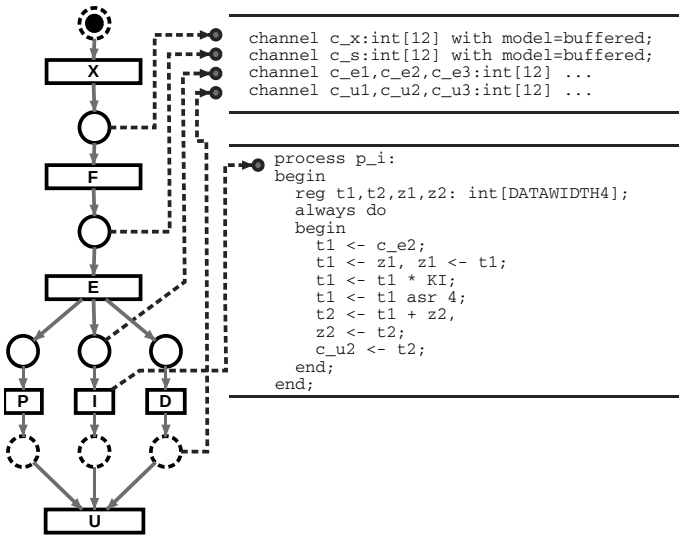
Petri Nets

- Functional blocks are mapped to transitions
- States represent data which is exchanged between functional blocks
- The partitioning of functional blocks to transitions can be performed on differenet complexity levels
- Forked transitions provide mapping to concurrent data processing

Programming Model

- Transitions are mapped to concurrently executing processes
- States are mapped to communication channels

Figure 3. Mapping of Petri-Net to multi-process programming model & ConPro language

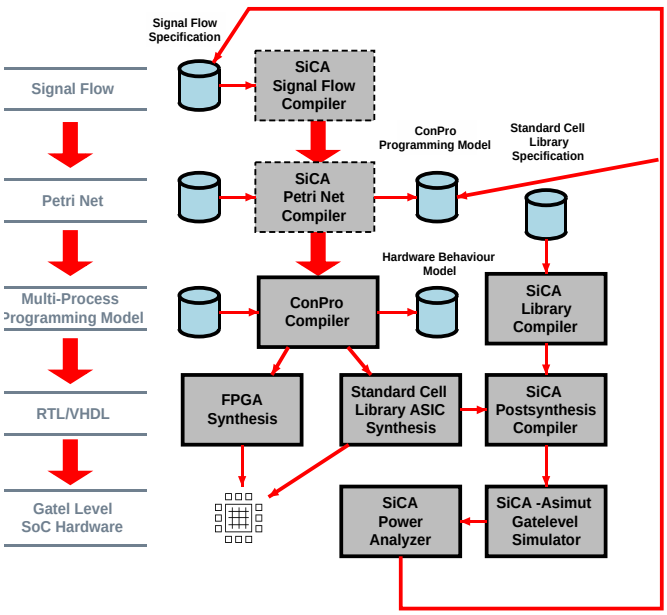


Design Flow

A closed-loop design flow enables the design of optimized low-power embedded systems

- Specification of data processing system using signal flow graphs
- Mapping of signal flow graphs to Petri Nets
- Synthesis of Petrie Net IR to processes and communication using ConPro programming modle and language
- Synthesis of embededded SoC design on RT level
- Gate-level synthesis of SoC using standard cell library
- Event driven gate-level simulation provides input for power analysis (activity traces)
- Activity and power analysis of SoC provides input for Smart Energy Management

Figure 4. Closed-loop design flow using a graph-based virtual database environment

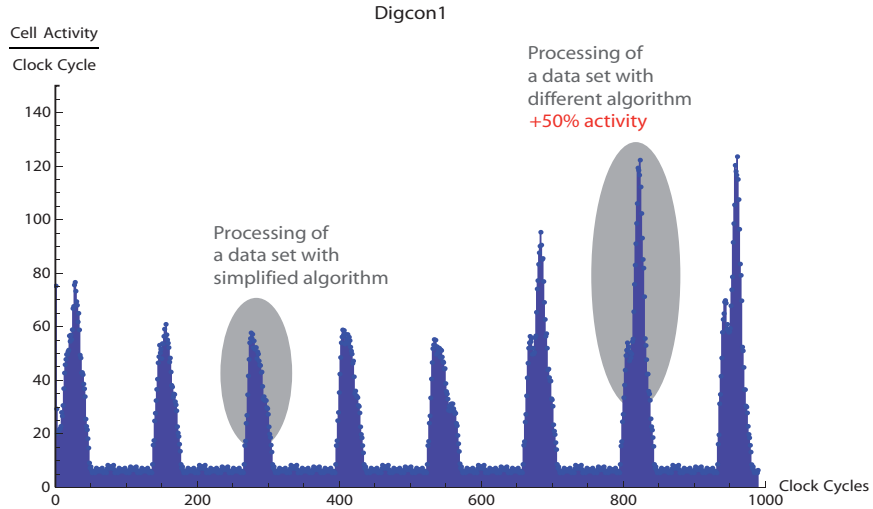


Activity Analysis by Gate-Level Simulation

Activity analysis using advanced simulation methods enables correlation with algorithm complexity

- Cell activity is defined by number of cells changing output signals per time unit
- Strong correlation of activity with data processing and algorithm complexity

Figure 5. Analysis of cell activity of a microchip implementing example PID controller. Two different complex algorithms results in different chip activity.

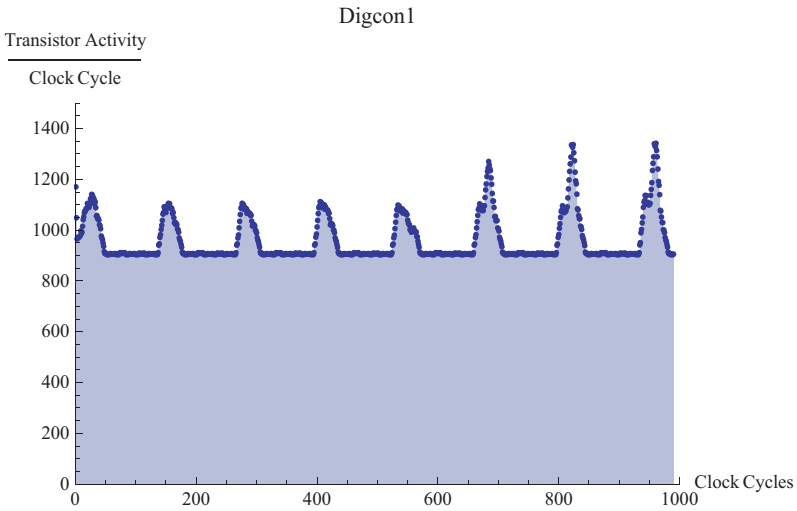


Transistor Switch Analysis by Gate-Level Simulation

Cell activity analysis leads to complexity classification of algorithms, but transistor activity is required for estimation of power consumption.

- In contrast to cell activity dominated by algorithm complexity, transistor switching is dominated by continious clock activity of registers (regardless of storage activity)

Figure 6. Analysis of transistor activity of a microchip implementing example PID controller.

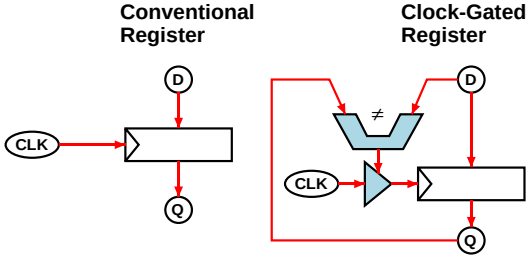


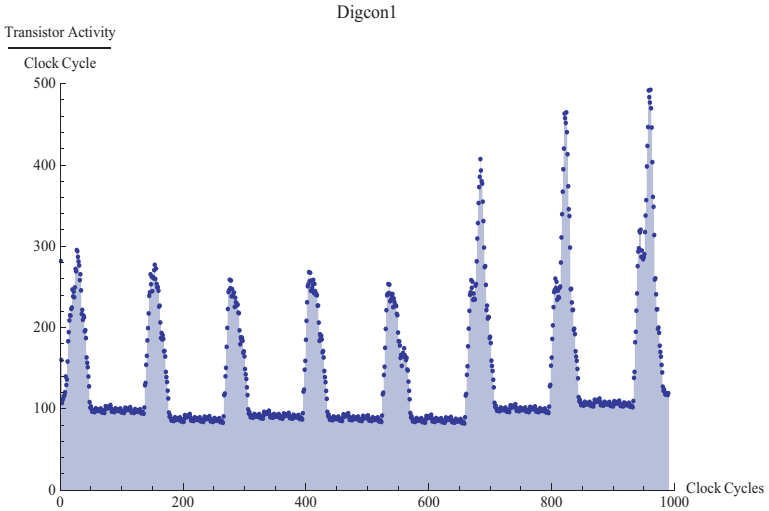
Technology Modification: Clock-Gated Registers

Clock gating of registers reduces transistor switching and power consumption significantly.

- Again there is a strong correlation between data processing activity and algorithm complexity

Figure 7. Analysis of transistor activity (right) of a microchip implementing example PID controller with clock-gated registers (left).



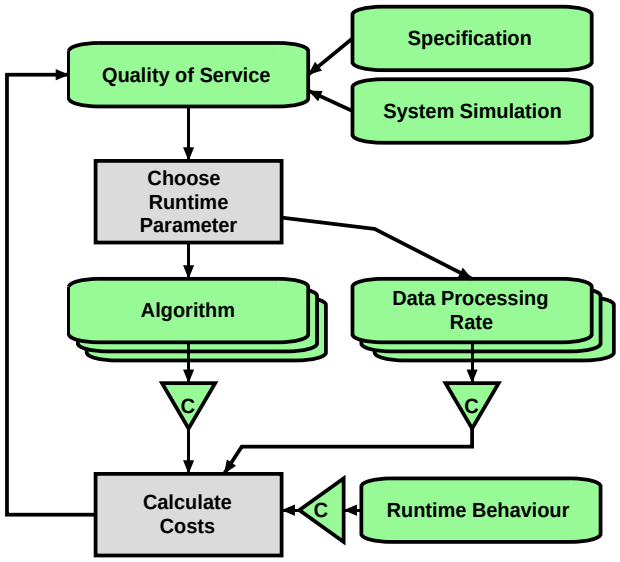


Smart Energy Management

Previous results from activity and energy analysis show strong correlation between algorithm complexity and power consumption of the data processing system.

- Self-powered systems are charged from “stochastic” energy sources with uncertain future
- Energy management using runtime parameter adaption can reduce the power consumption on a longer time scale
- If there is a set of algorithms *A* with different complexity and Quality-of-Service (QoS), a best suited algorithm can be selected at runtime.
- Each parameter is assigned a cost factor C_n (derived from analysis), runtime behaviour get cost values C_r
- Energy management is performed by parameter selector with cost-feedback control

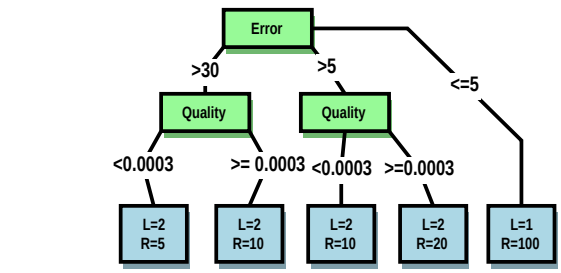
Figure 8. Selection of different algorithms with feedback-of-cost analysis at runtime

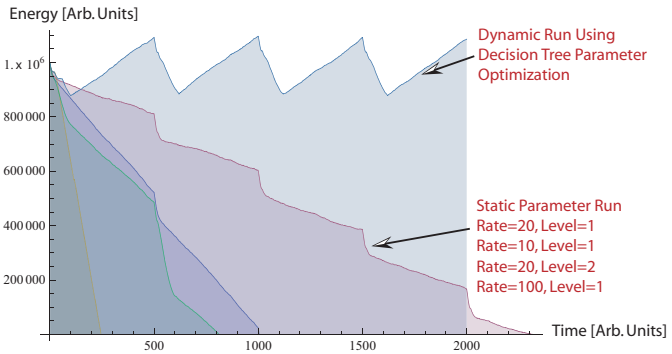


Smart power management at runtime can be performed using advanced methods from artificial intelligence.

- Runtime parameter selection with descision trees are suitable for runtime management
- Machine Learning can be used to retrieve descision trees.
- **Example:** Simulation of system consisting of PID controller example and a simple actuator model
- The system is charged from a stochastic energy source and discharged by data processing and actuator activity
- **Runtime parameters**: 1. set of data processing rates `R={1,2,5,10,50, 100 ms}`, 2. two different algorithms for position error control: `L={P:1, PID:2}`
- **Result:** with dynamic parameter optimization system reaches stable state

Figure 9. Descision tree used at runtime





Summary

Low-Power Design ...

- using high-level synthesis from programming level provides application-specific implementation of algorithms in digital logic.
- Application-specific SoC designs on RT level contributes to resource minimization and enhance system efficiency significantly.
- Signal flow graphs used for system modelling can be used to derive parallelized data processing based on token passing with variable data processing rates.
- Algorithm complexity and data processing rates correlate strongly with microchip activity and power consumption - derived from gate-level simulation -
- But technology extension required: clock-gated registers

Energy Management ...

- implemented in the microchip on algorithmic level can improve system behaviour and reduces energy consumption at runtime.
- If data processing can be implemented with a set of algorithms with different complexity and quality-of-service, dynamic algorithmic selection can be used to optimize system behaviour.
- Additionally data processing rate optimization is used in energy management.
- Machine learning methods can be used to derive descisions trees for dynamic parameter optimization during runtime.