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

Stefan Bosse[1,3], Thomas Behrmann[2,3]

University of Bremen, Department Computer Science, Workgroup Robotics, Germany[1], BIMAQ Bremen Institute for Metrology, Automation and Quality Science[2], University of Bremen, ISIS Sensorial Materials Scientific Centre, Germany[3]

Abstract

We propose and demonstrate a design methodology for embedded systems satisfying low power requirements suitable for self-powered sensor and actuator nodes. This design methodology focuses on 1. smart energy management at runtime and 2. application-specific System-On-Chip (SoC) design at design time, contributing to low-power systems on both algorithmic and technology level.

Smart energy management is performed spatially at runtime by a behaviour-based or state-action-driven selection from a set of different (implemented) algorithms classified by their demand of computation power, and temporally by varying data processing rates. It can be shown that power/energy consumption of an application-specific SoC design depends strongly on computation complexity.

Signal and control processing is modelled on abstract level using signal flow diagrams. These signal flow graphs are mapped to Petri Nets to enable direct high-level synthesis of digital SoC circuits using a multi-process architecture with the Communicating-Sequential-Process model on execution level. Power analysis using simulation techniques on gate-level provides input for the algorithmic selection during runtime of the system, leading to a closed-loop design flow. Additionally, the signal-flow approach enables power management by varying the signal flow and data processing rates depending on actual energy consumption, estimated energy deposit, and required Quality-of-Service.

1. Introduction and Overview

Today there is an increasing demand for miniaturized smart sensors embedded in sensorial materials and smart actuators. Each sensor and actuator node provides some kind of sensor, electronics, data processing, and communication. With increasing miniaturization and sensor-actuator density, decentralized network and data processing architectures are preferred, but energy supply is still centralized. Using local energy-harvesting technologies, a **decentralized energy supply** can be provided, too. **Energy harvesting**, for example using solar cells, photo diodes sourced by optical fibers, or thermo-electrical sources actually delivers only low electrical power (due to technology or size constraints).

We propose and demonstrate a design methodology for embedded systems satis-

SPIE Microtechnologies 2011 Conference, 18.4.-20.4.2011, Prague, Session EMT 101
Smart Sensors, Actuators and MEMS, Proc. SPIE 8066, 806619 (2011); doi:10.1117/12.888124, Page 1 of 11

fying low-power requirements suitable for self-powered sensor and actuator nodes. This design methodology focuses on

1. **smart energy management** at runtime using advanced computer science algorithms (artifical intelligence) and
2. application-specific **System-On-Chip** (SoC) design using high-level synthesis at design time. Low-power systems are designed on algorithmic rather than on technological level.

In contrast to various other approaches targeting algorithms and architectures with high computational effort, for example **[5]**, the proposed **smart energy management** is performed spatially at runtime by a selection from a set of different (implemented) algorithms classified by their demand of computation power, and temporally by varying data processing rates. It can be shown that power/energy consumption of an application-specific SoC design strongly depends on computation complexity.
For example, a classical Proportional-Integral-Differential (PID) controller used for feedback position control of an actuator requires basically only the P-part; the I- and D-parts only increase position accuracy and response dynamics which are selectable. Depending on the actual state of the system and the actual and estimated future energy deposit, suitable algorithms can be selected and executed optimizing the Quality-of-Service (QoS) and the trade-off between accuracy and economy.
Signal and control processing is modelled on abstract algorithmic level using signal flow diagrams. These signal flow graphs are mapped to Petri Nets to enable direct high-level synthesis of digital SoC circuits using a multi-process architecture with the Communicating-Sequential-Process model on execution level and the high-level synthesis framework ConPro **[1]**.
Power analysis using simulation techniques on gate-level provides input for the algorithmic selection during runtime of the system leading to a closed-loop design flow. Additionally, the signal-flow approach enables power management by varying the signal flow rate which will be discussed later.

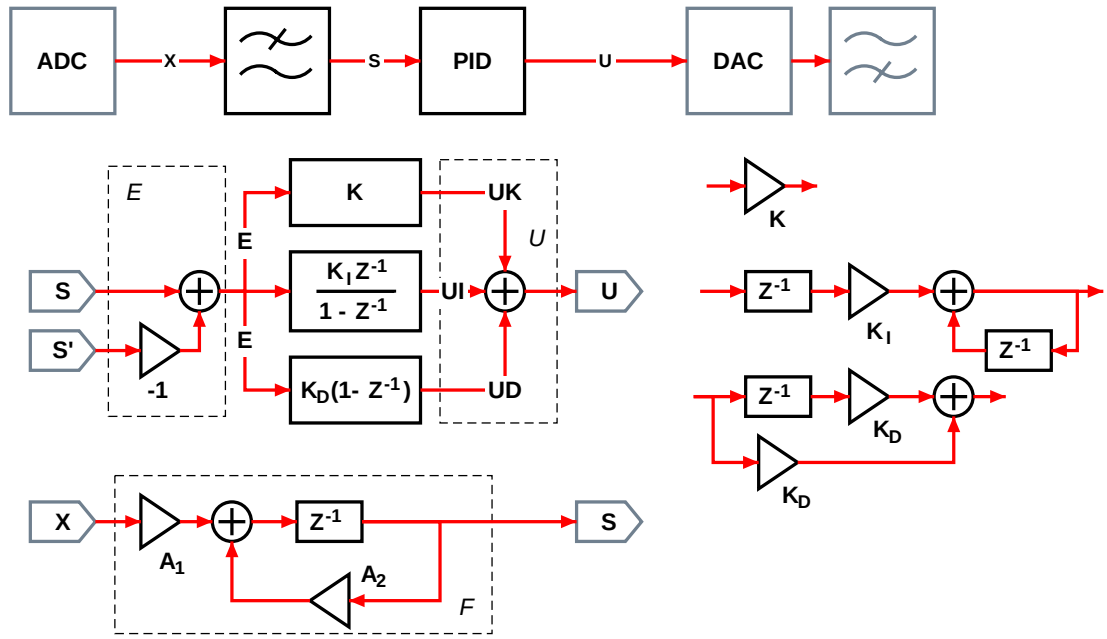
2. Design Flow

The design flow for low-power embedded data processing and control systems should be demonstrated using a concrete example. The system is modelled on an abstract level using signal flow diagrams **[3]**.

Figure **1** shows a composition of a complete feedback-controlled system consisting of sensor signal acquisition (ADC), filtering, an error controller with a proportional, integral, and differential sub-controller **[4]**, and finally a signal generator (DAC) driving an actuator. The controller is used to control the position of an actuator. The control error is defined by the difference of the acquired position signal X(S) and the setable position parameter S'.
This initial specification is used to derive 1. a multi-process programming model, and 2. a hardware model for a SoC design on Register-Transfer level. Furthermore,

the signal flow diagram provides input for energy optimization at synthesis and run-time.

Figure 1. Composition and modelling of a digital control system with signal flow diagrams



The signal flow diagram is first transformed into a S/T Petri Net representation which is shown in figure **2**. Functional blocks are mapped to transitions, and states represent data which is exchanged between those functional blocks. The partitioning of functional blocks to transitions of the net can be performed at different composition and complexity levels. The signal flow diagram from figure **1** was partitioned using complex blocks (merging low-level blocks like multipliers and adders) to reduce communication complexity (and data processing latency).
Sensor data (X) is acquired periodically and passed to the data processing system. A token of the net is equal to a data set of one computation processed by the functional blocks. The functional blocks P,I, and D are placed in concurrent paths of the net.
The Petri Net is then used 1. to derive the communication architecture, and 2. to determine an initial configuration for the communication network. Functional blocks with a feedback path require the injection of initial tokens in the appropiate states (not required in the example).
States of the net are mapped to buffered communication channels and transitions are mapped to concurrently executing processes - each with sequential instruction processing - using the ConPro programming language **[1]**, shown in figure **2**, too.
Forked states indicate concurrency in the Petri Net flow. Exploring concurrency in signal flow diagrams using Petri Nets reduces latency for the computation of one data set. Also pipelining can decrease latency of a data set stream significantly, de-

rived again from the Petri Net representation.

Figure 2. Mapping of the signal flow diagram to a Petri Net and mapping of Petri Net to communication channels and sequential processes using the ConPro programming language.

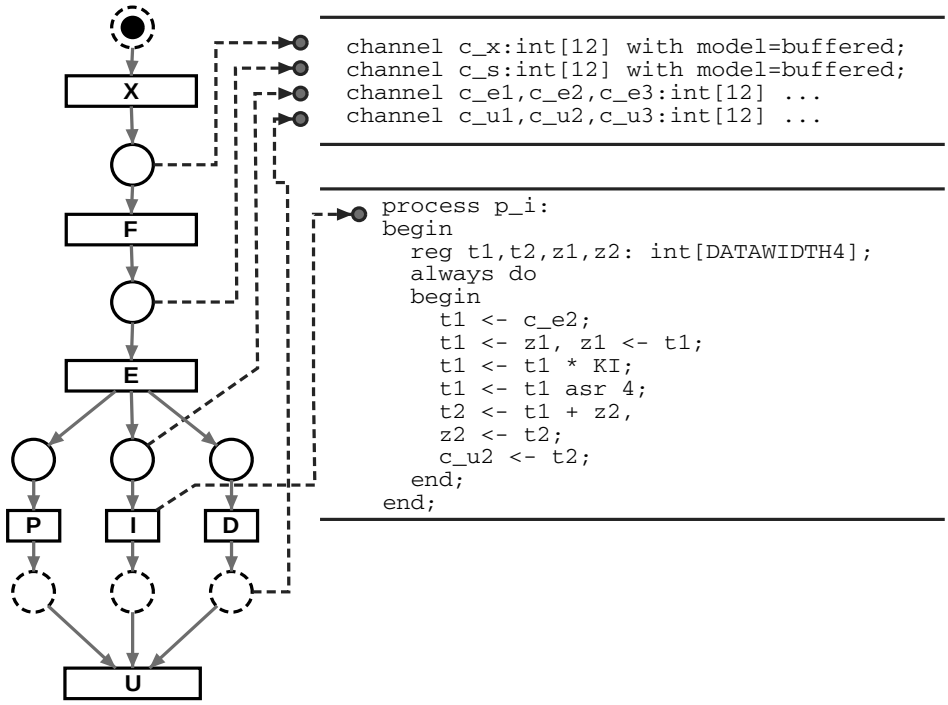
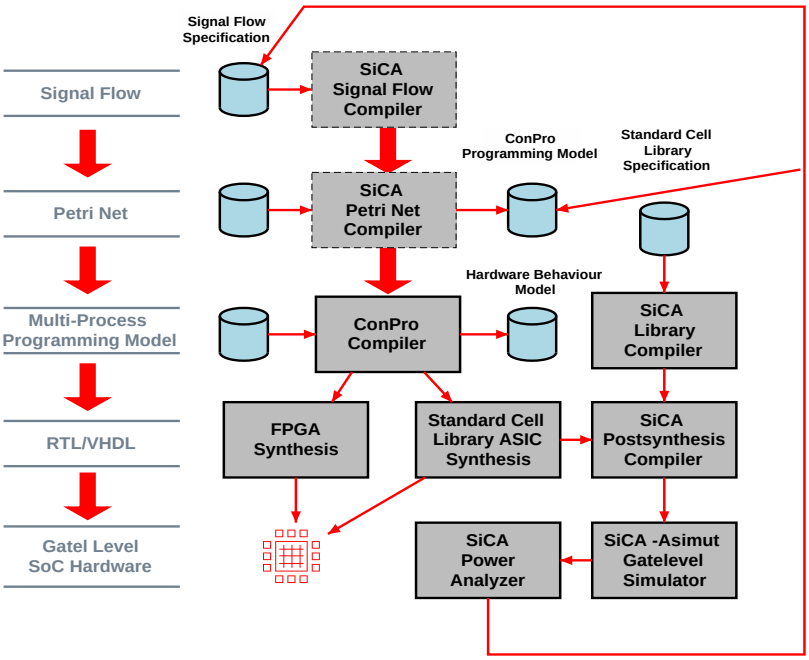


Figure 3. Overall design flow for low power embedded systems using the SiCA and ConPro Synthesis and Analysis framework (Dotted boxes are actually under development). The SiCA design flow uses a graph-based virtual database for advanced data management.



SPIE Microtechnologies 2011 Conference, 18.4.-20.4.2011, Prague, Session EMT 101
Smart Sensors, Actuators and MEMS, Proc. SPIE 8066, 806619 (2011); doi:10.1117/12.888124, Page 4 of 11

The complete design flow is shown in figure **3**. It is a closed-loop design flow with feedback from power analysis. Results from power analysis are used to make modifications and optimizations on algorithmic level (signal flow and programming model level). The SiCA design flow consists of several analyzer and compiler modules, and uses a graph-based virtual database for advanced data management and inter-module data exchange.

3. Energy Analysis

The derived multi-process programming model was synthesized to a digital logic SoC using high-level synthesis. For simulation, gate-level synthesis was performed with a standard logic cell technology library. The resulting net-list was analyzed with an event-driven simulator, calculating the overall cell activity for each time unit, defined by terms of cell output changes. Synthesis and analysis were performed using the Concurrent Programming (ConPro) compiler **[1]** and the Silicium-Compiler-and-Analyzer framework (SiCA).
The SoC circuit activity correlates strongly with a computation of a new data set (with sensor data input sampled periodically) and the computation complexity, shown in figure **4**.
The logic cell activity of the circuit has strong peaks around the computation of a new output value U. About every 140 clock cycles a new input value X is generated, triggering the calculation of a new output value U.
The first five data sets are computated with an enabled P-part of the controller only. After the fifth computation, the I and D parts were enabled, too. This results in an increase of circuit activity of about 50%.
But power dissipation cannot be estimated directly from this cell activity. Logic cells consist of a network of (paired) transistors. Power dissipation of a CMOS circuit depends proportionally from the transistor switching activity. Simulation results for the controller are shown in figure **5** (using SiCA, too). There is only weak correlation between data processing activity (and computation complexity) and power dissipation due to clocking activity of registers.
Power dissipation can only be estimated from the above circuit cell activity if clock-gated registers are assumed **[2]**. The principle-architecture of gated registers is shown in figure **6**. The clock gating prevents switching activity of the register cell if there is no change of input data. Fine-grained inherent clock gating is a requirement for the proposed low-power design method and enables a strong correlation between computation activity (and hence algorithmic complexity) with the power dissipation of the data processing system.
Results of such a modified control system with clock-gated registers are shown in figure **7**. There is again a significant increase of transistor switching activity of about

30% if the two different computation levels (P, PID) are compared.

Figure 4. Averaged SoC cell activity correlates strongly with computation and signal/data flow. After obtaining the fifth result value U, the I and D computational blocks are switched on.

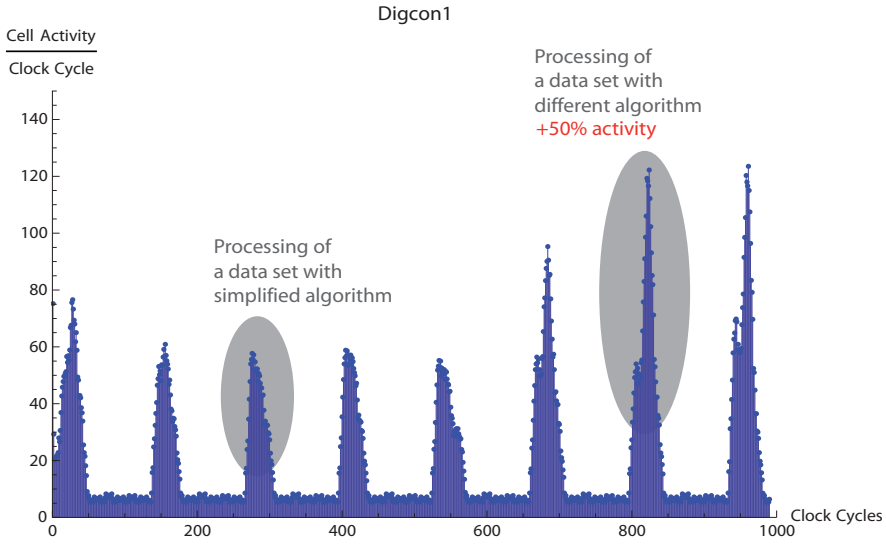


Figure 5. Averaged SoC transistor switching activity of the circuit retrieved from simulation. Power dissipation is proportional to transistor activity.