

Smart Energy Management and Low-Power Design of Embedded Systems on Algorithmic Level for Self-Powered Sensorial Materials and Robotics

Stefan Bosse^(1,3), Thomas Behrmann^(2,3), Frank Kirchner^(1,3)

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

A new design methodology for low-power embedded systems is presented which is based on advanced algorithms from computer science. System-On-Chip architecture, power analysis, and advanced system modelling methodologies are used.

Keywords

Energy Management, Low-Power, System-On-Chip, Sensorial Materials, Microsystems, High-Level Synthesis, Signal Flow Diagrams, Petri-Nets

1. Introduction and Overview

Today there is an increasing demand for miniaturized smart sensors embedded in sensorial materials and smart actuators in microsystem applications. 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 thermo-electrical sources, actually delivers only low electrical power, requiring 1. smart energy management on the consumer side controlling the energy consumption and 2. low-power design.

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 technological 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 consump-

tion of an application-specific SoC design depends strongly on computation complexity. For example a 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 dynamic 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 level using signal flow diagrams (Matlab & Simulink toolbox [2]). 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 is obtained from the Silicium Compiler and Analyzer SiCA providing 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 of low-power embedded data-processing and control systems should be demonstrated using a concrete example. The system is modelled using signal flow diagrams.

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

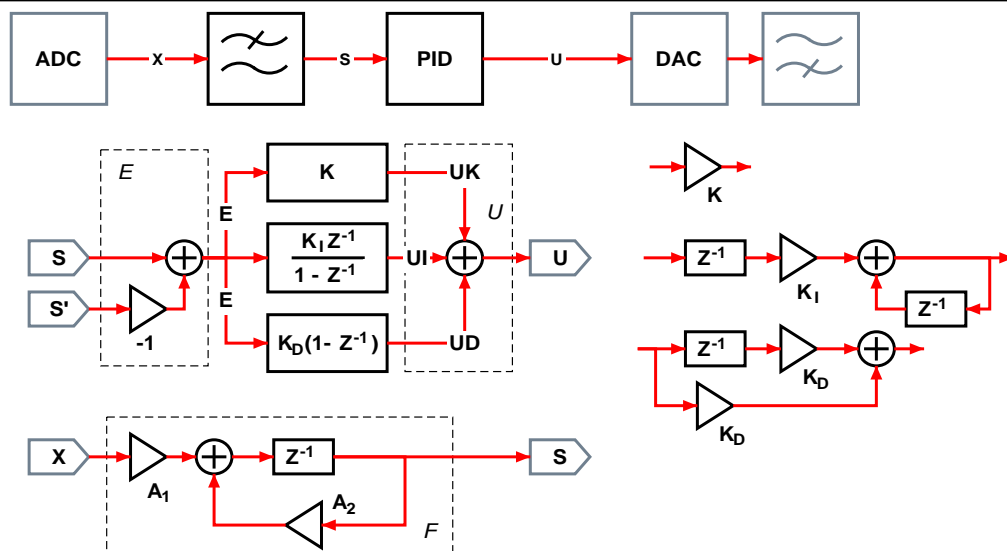
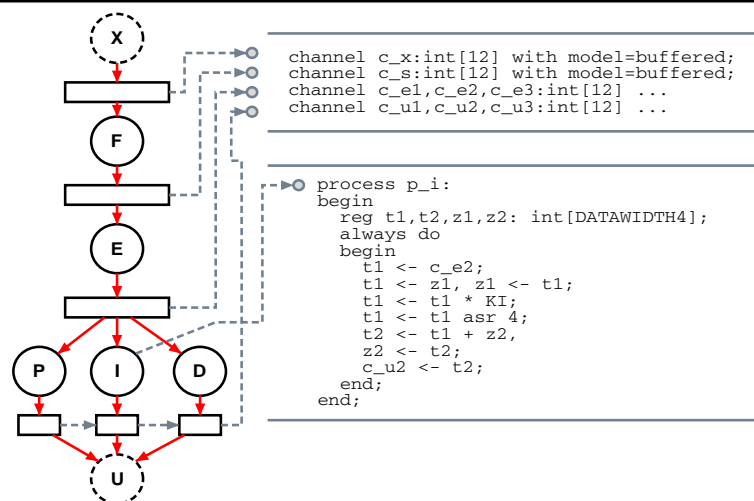


Figure 1 shows a composition of a complete feedback controlled system consisting of sensor signal acquisition (ADC), filtering, and error controller with a proportional, integral, and differential sub-controller [4], and finally a signal

generator (DAC).

This initial specification is used to derive 1. a multi-process programming model, and 2. a hardware model for a System-On-Chip design on Register-Transfer level. Furthermore, the signal flow diagram provides input for energy optimization at synthesis and runtime. The signal flow diagram is first transformed into a S/T Petri Net representation which is shown in figure 2. The Petri Net is used 1. to derive the communication architecture, and 2. to determine an initial configuration for the communication network. Transitions of the net are mapped to buffered communication channels and states are mapped to sequential processes using the ConPro programming language [2].

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.



At runtime different state variables are analyzed and controlled by a constraint-net approach. The set of variables consists of: Energy, error, runtime, processing rate, and processing level with a definition set [0,1,2,3] assigning quality factors to the state variables. The constraints are defined in definition 1.

Definition 1. Constraint relations satisfying quality of service and minimizing power consumption to be fulfilled at runtime.

```

Runtime ≤ 3 - Error,
Runtime = Energy | Runtime = Energy - 1 | Runtime = Energy + 1
Energy ≥ Rate & Rate > 0 & Energy > 0
Error ≤ 3 - Rate & (Rate > 0 | Error = 3)
Error ≤ Level
Energy ≥ Level

```

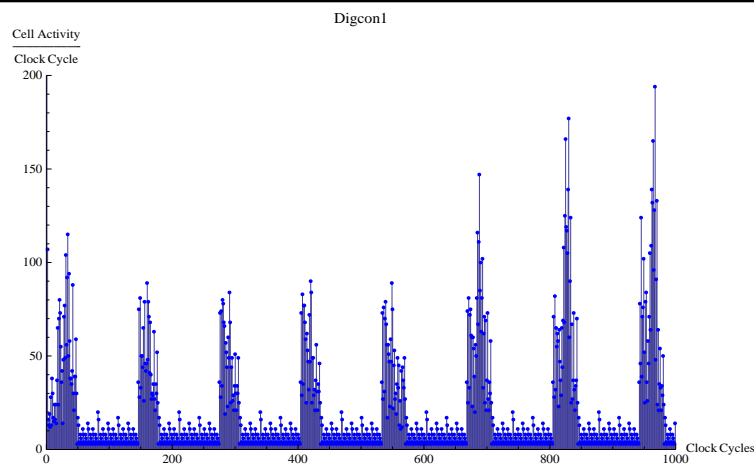
A set of solutions are computed at design time using a chronological backtracking algorithm and monte carlo simulation. These solutions are stored in a look-up table of the hardware design. During runtime best matching energy optimized solutions depending on the actual state can be selected. Data processing rate and level can be chosen for an actual energy and runtime estimation.

3. Experimental Results and Conclusions

The derived multi-process programming model was synthesized to RTL using the ConPro compiler. Gate-level synthesis was performed with a standard cell library (sxlib) and Leonardo Spectrum. The resulting net-list was analyzed with an event-driven simulator (modified asimut which is part of SiCA), calculating the overall cell activity for each event time (with sub clock-period resolution), and finally cumulating for one clock cycle. Power dissipation can only be estimated from this value if clock-gated registers are assumed [3]. Results of the control system are shown in figure 3. The cell activity of the circuit has strong peaks around the computation of a new output value U. Every 140 clock cycles a new input value X is generated. The first five computations are only performed with the P-part of the controller, after the fifth calculation the I and D computational blocks are switched on. This results in an increase of activity of about 50% (about 1900 versus 3000 cell switches for one output value computation). The full computation of an output value requires a power of 150 μ W/MHz for a TSMC 0.18 μ m 1.8V ASIC process.

The power dissipation of the controller can be changed during runtime 1. by selecting different computation levels, and 2. by varying the data processing rate.

Figure 3. 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.



Bibliography

- [1] S. Bosse, *Hardware Synthesis of Complex System-on-Chip-Designs for Embedded Systems Using a Behavioural Programming and Multi-Process Model*, Proceedings of the 55th IWK - Internationales Wissenschaftliches Kolloquium, Session C4, Ilmenau, 13 - 17 Sept. 2010
- [2] T. Behrmann, C. Zschippig, M. Lemmel, S. Bosse, *Toolbox for Energy Analysis and Simulation of self-powered Sensor Nodes*, Proceedings of the 55th IWK - Internationales Wissenschaftliches Kolloquium, Session A3, Ilmenau, 13 - 17 Sept. 2010
- [3] Y. Xia and A.E.A. Almaini., *Differential CMOS edge-triggered flip-flop with clock-gating*, ELECTRONICS LETTERS, Vol. 3rd January 2002 Vol. 38 No. 1J
- [4] R. Isermann, *Digital Control Systems*, Springer, 1989