10.1117/2.1201106.003694

Smart energy management and low-power embedded system design

Stefan Bosse and Frank Kirchner

Self-powered embedded systems with a large number of active sensor and microsystem actuator nodes fulfill the low-power requirements of miniaturized sensors.

There is growing demand for miniaturized smart sensors that are embedded in sensorial materials and smart actuators. In these applications, each sensor and actuator node provides a sensor, electronics, data processing, and communication. Increasing miniaturization and sensor-actuator density are making it challenging to design an appropriate network, data processing architecture, and energy supply. Decentralized network and data processing architectures are preferred, but energy supply is still typically centralized. A decentralized energy supply with local energy-harvesting technologies, such as solar cells or thermo-electrical sources, could only be used to deliver low electrical power due to technology or size constraints.

Embedded systems satisfying low-power requirements, suitable for self-powered sensor and actuator nodes, are still largely in development. Our work proposes and demonstrates such a design. It focuses on smart energy management at runtime using advanced computer science algorithms (artificial intelligence) and an application-specific system-on-chip (SoC) design using high-level synthesis. Low-power systems are designed on an algorithmic, as opposed to technological, level.

Self-powered systems must function with a limited amount of energy during runtime. The energy charge that will be available in the future is uncertain, so it should be effectively handled with smart energy management. Methods from artificial intelligence (AI) can be used to manage energy at runtime with dynamic parameter adaption and algorithm selection. However, AI methods differ in complexity, so only a few are suitable for embedding in microchips.

In an application-specific SoC, smart energy management can be implemented at runtime by using an algorithm based on power demand, as well as by varying the data-processing rates. The power consumption of this type of SoC depends strongly on computational complexity. For example, a classical proportional-integral-differential (PID) controller used for feedback-position control of an actuator requires essentially only the proportional (P) component. The integral (I) and differential (D) components 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, specific algorithms can optimize the quality-of-service as well as the trade-off between accuracy and economy.

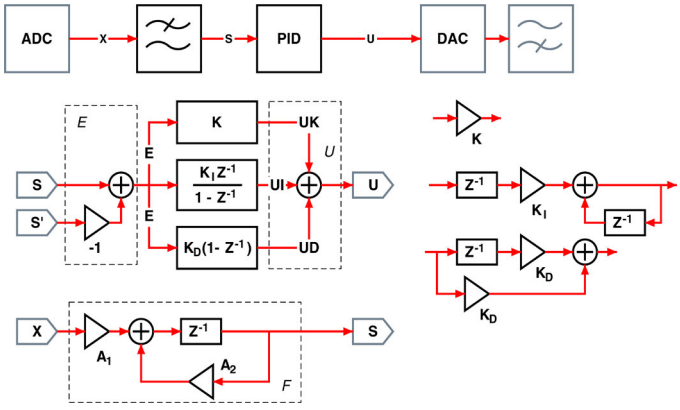

Figure 1. *Composition and modeling of a digital control system with signal flow diagrams. ADC: Sensor signal acquisition. PID: Proportional-integral-differential controller. DAC: Signal generator. S: The actual (low-pass filtered) input signal for the PID controller, representing a position value coming from an analog-to-digital converter. S′: The desired position value of the actuator. E: Position error (S-S′). F: Low-pass filter implementation (the block between the original measured signal X and the filtered signal S). U: Calculated output signal (driving the actuator). The other symbols represent different data processing elements of the digital control system.*

Continued on next page

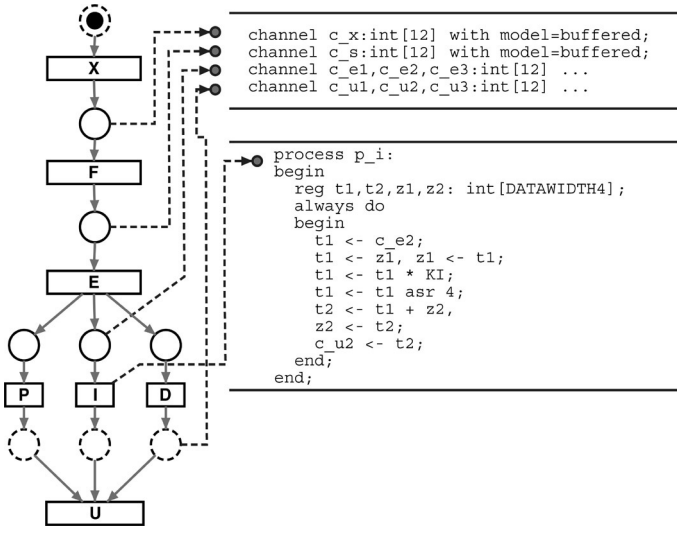

Figure 2. *Mapping of the signal flow diagram to a Petri Net, and mapping of the Petri Net to communication channels and sequential processes, using the ConPro programming language. P: Proportional controller. I: Integral controller. D: Differential controller.*

We used signal flow diagrams to model signal and control processing. By mapping these signal flow graphs to Petri Nets, we were able to direct high-level synthesis of digital SoC circuits through a multi-process architecture that used the communicating-sequential-process model on the execution level and a high-level synthesis framework (ConPro).[1] We performed power analysis using simulation techniques at the gate level to provide input for algorithm selection, which resulted in a closed-loop design flow. An additional advantage of our approach is that it enabled power management by varying the signal flow rate.

Figure 1 shows a complete feedback-controlled system that consists of sensor signal acquisition, filtering, and an error controller (with a PID sub-controller),[2] and a signal generator driving an actuator. We used this initial specification to derive a multi-process programming model and a hardware model for an SoC design at a register-transfer level. The signal-flow diagram provided input for energy optimization at synthesis and runtime. We used the Petri Net (see Figure 2) to derive the communication architecture and to determine an initial configuration for the communication network. (States of the net are mapped to buffered communication channels, and transitions are mapped to sequential processes using the ConPro programming language.[1] A token of the net is equal to a data set for one computation.)

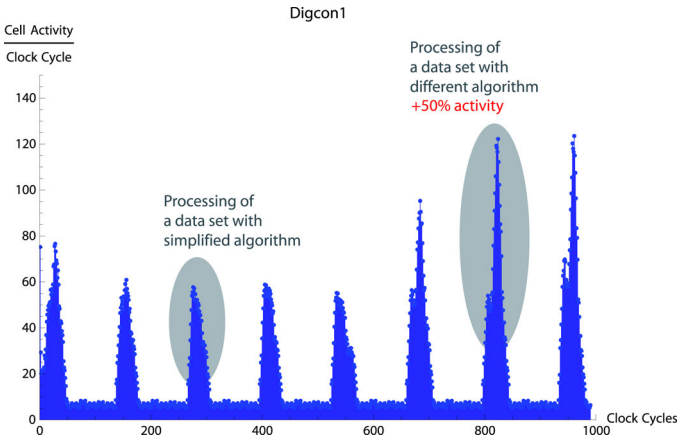

Figure 3. *Averaged system-on-chip cell activity correlates strongly with computation and signal/data flow. After obtaining the fifth result value U, the I and D computational blocks are switched on.*

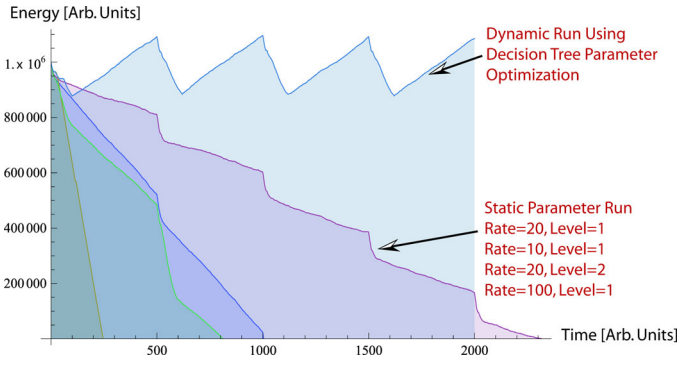

Figure 4. *System simulation with different runtime behaviors using a decision tree, which can be retrieved by machine learning methods. Parameters: Data processing rate* $= \{1, 5, 10, 20, 100\}$. *Algorithmic level* $= \{P : 1, PID : 2\}$.

We used this multi-process programming model to synthesize a digital logic SoC. For simulation, we performed gate-level synthesis with a standard cell library. The resulting net-list was analyzed with an event-driven simulator that calculated the overall cell activity for each time unit (see Figure 3). We estimated the power dissipation by using an assumption of clock-gated registers.[3]

Our analysis found that the cell activity of the circuit had strong peaks when computing a new output value (U). Approximately every 140 clock cycles, a new input value (X) was generated. The first five computations were performed with the P-part of the controller enabled, and after the fifth calculation

Continued on next page

the I and D computational blocks were added. This resulted in an activity increase of approximately 50%.

Figure 4 shows simulation results of a complex sensor-actuator system implementing the PID controller. We used a decision tree method for parameter adaption. The system was charged with a stochastic energy source that was discharged by computation and actuator activity. With static parameter settings, this system always ran out of energy, regardless of parameter settings. However, using the smart decision tree method, we found that the system could hold a stable power level.

Miniaturized embedded smart sensors come with network, data, and energy challenges. We have demonstrated a methodology for designing embedded systems with low-power requirements based on smart energy management. Our future work will investigate suitable machine learning methods to retrieve decision trees.[4]

Author Information

Stefan Bosse
Department of Mathematics and Computer Science, Robotics
University of Bremen
Bremen, Germany

Stefan Bosse received a diploma in physics from the University of Bremen in 1998. He received his PhD in physics in 2002. In 2004 he joined the Department of Computer Science and the robotics workgroup. Since 2008 he has conducted a project in the Integrated Solutions in Sensorial Structure Engineering (ISIS) Sensorial Materials Scientific Center.

Frank Kirchner
Department of Mathematics and Computer Science
University of Bremen
Bremen, Germany

References

1. S. Bosse, *Hardware synthesis of complex system-on-chip-designs for embedded systems using a behavioural programming and multi-process model*, **Proc. 55th IWK - Internationales Wissenschaftliches Kolloquium**. In press.
2. R. Isermann, **Digital Control Systems**, Springer, 1989.
3. Y. Xia and A. E. A. Almaini, *Differential CMOS edge-triggered flip-flop with clock-gating*, **Electron. Lett. 38**, pp. 9–11, 2002. doi:10.1049/el:20020038
4. S. Bosse and Thomas Behrmann, *Smart energy management and low-power design of sensor and actuator nodes on algorithmic level for self-powered sensorial materials and robotics*, **Proc. SPIE 8066,** pp. 806619, 2011. doi:10.1117/12.888124