Developing Power-Aware Strategies for the Blackfin Processor

Steven VanderSanden
David Kaeli
Northeastern University
Boston, MA

svanders@ece.neu.edu
kaeli@ece.neu.edu

Giuseppe Olivadoti
Richard Gentile
Analog Devices
Norwood, MA

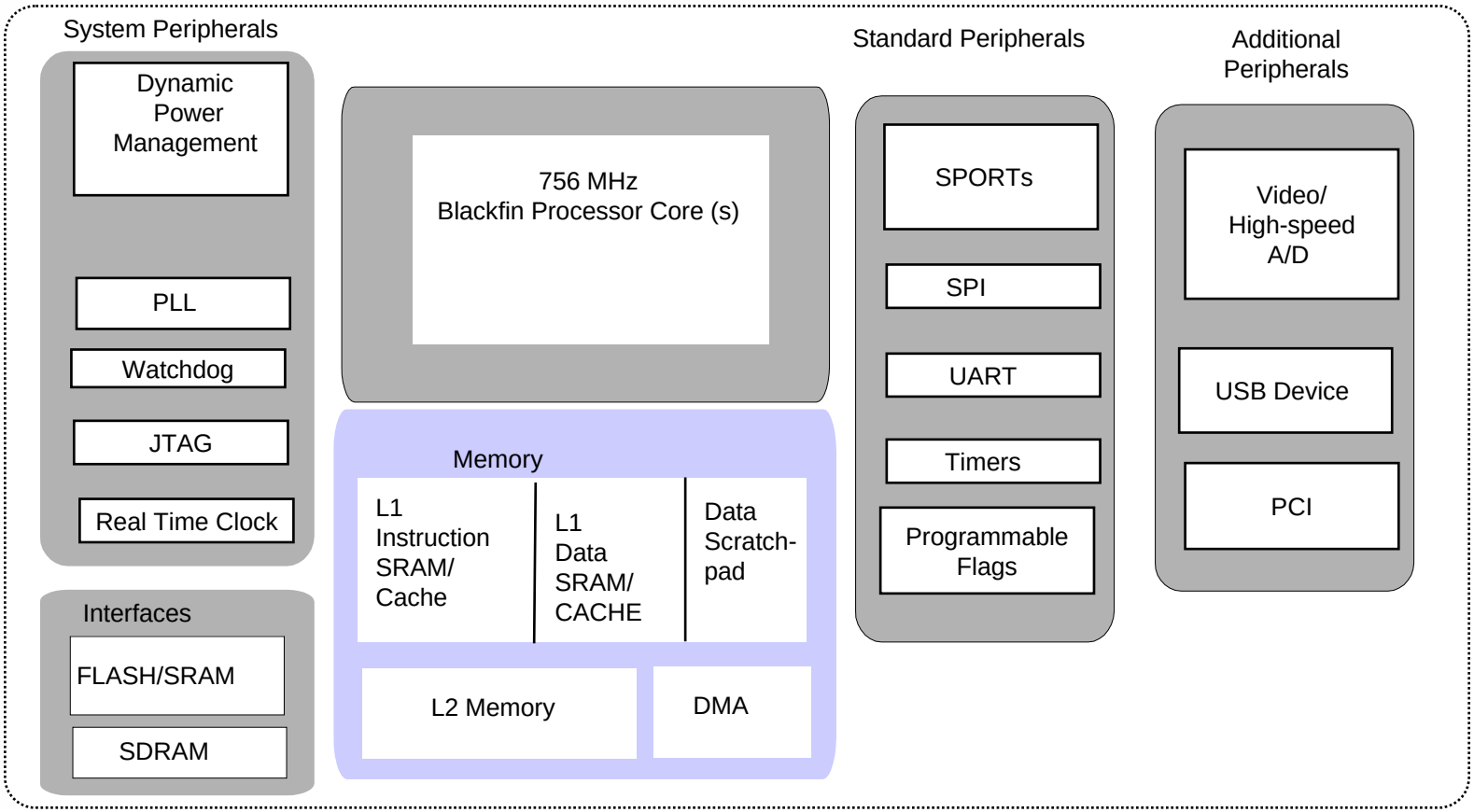
giuseppe.olivadoti@analog.com
richard.gentile@analog.com

The Need for Accurate Power Estimation

- Power management is particularly critical for portable embedded systems
- Power estimates will drive future core design decisions and impact battery design
- Present power estimation techniques utilize abstract architectural models
  - Good for predicting relative performance, but lack precision
  - Difficult to adapt across different core models
- Our work develops an instruction-level model
  - Profiles power/energy instruction-by-instruction
  - Utilizes statistical methods for estimating full program power
  - Methodology is portable to any embedded processor design
- This project is targeted to improve the power analysis capabilities of the ADI Blackfin family of processors and systems

Blackfin Processor

- **Built around Micro Signal Architecture, developed jointly with Intel Corp.**
- **Integrates DSP with features more typically found in an MCU**
- **Full suite of power management capabilities**



A DSP with a RISC instruction set and an MMU, an event controller and a wide range of peripherals

Our Approach

- Instruction-level power modeling
  - Computes energy budget by characterizing single instruction and inter-instruction power usage, combined with instruction execution time
  - Total energy = base energy cost + inter-instruction effects
- Profiling is used to construct a power/energy table for both base costs and inter-instruction effects
- We consider variability introduced by:
  - Operand types and operand values
  - Memory system configuration
  - Instruction selection
  - Measurement environment
- We then build energy estimates, working with individual basic blocks, and weight blocks based on the dynamic call graph traversal during program execution
- We are able to accurately estimate full program behavior (including memory access) within 6% of measured values