Energy Management in Embedded Systems – Towards a Taxonomy

Umesh Balaji Kothandapani Ramesh
Mälardalen University
School of Innovation, Design and Engineering,
Software Engineering Division, Västerås, Sweden
saiumeshbalaji@gmail.com

Severine Sentilles, Ivica Crnkovic
Mälardalen University
School of Innovation, Design and Engineering
Software Engineering Division, Västerås, Sweden
{severine.sentilles, ivica.crnkovic}@mdh.se

Abstract—**Energy is an important constraint in embedded systems, and there exists a huge expertise in this domain about monitoring, managing and optimizing energy consumption in the computer systems. The aim of this paper is to present the energy management addressed in the research literature. Based on a systematic review, the paper presents a taxonomy of energy consumption and management in embedded systems.**

Keywords – embedded systems, energy consumption, systematic review

I. INTRODUCTION

Embedded systems (ES) are computer systems that are integrated in another systems. Their purpose is to control these systems and to provide information important for the system functions. ES are highly interactive with their environment, are performing often in real-time, and are continuously available. ES make the vast majority of computer systems (99,99% of computer nodes belong to the category of embedded systems [1]). In addition to their functional specifics, ES are characterized by specific extra-functional properties and constraints: dependability, real-time and resources. In respect to the resources, ES are classified as resource-constrained and resource non-constrained embedded systems. They range from very small systems (for example controlling small sensors) to very large distributed complex systems (for example distributed telecommunication systems). In addition to their functional specifics, ES are characterized by specific extra-functional properties and constraints: dependability, real-time and resources. In respect to the resources, ES are classified as resource-constrained and resource non-constrained embedded systems. Energy and power is one of the most important constraints, and often related to other properties. The importance of energy is less related to its large consumption or large savings (looking from a "green" energy perspective), but to the system dependability – loosing access to energy leads to degradation or to a non-functioning system, which can have severe consequences. This is in particular true for dependable (especially safety-critical) systems. In addition, a dramatic increase of ES with wireless communication capability leads to an enormously increasing use of batteries which consequently increases the waste dangerous for nature. During many years of ES development, many energy-aware solutions have been provided – from energy consumption optimization on the code level, through software scheduling, and to software and system architecture optimization. This knowledge can be successfully applied in design of other type of systems, for example those that are huge consumers of energy.

In this paper we provide taxonomy of energy concerns in ES development. The taxonomy is derived from the systematic literature review and the categorization of the topics is based on findings from this literature. To obtain information from the research papers we adopted a Systematic Literature Review (SLR) [2]. The starting point of the SLR is the formulation of the research questions. Our primary research interest was extra-functional properties in ES design (research questions: "1. Which extra-functional properties mostly are considered in ES development?"; and 2. "What is the relationship between the identified extra-functional properties?"). [1] We did the search ACM, IEEE Xplore, and SpringerLink databases. The total numbers of research papers obtained were 1138 from the prescribed databases. Out of the 1138 research papers, 76 papers were directly related to power/energy properties of embedded system. This clearly shows energy/power as one of the most important extra-functional property. By further reading those papers and using forward references, we selected additional 19 papers, collecting in total 95 papers. The complete reference list is presented in a technical report [3]. Based on these papers we have derived a taxonomy for power/energy properties of embedded system is constructed in this paper. The taxonomy is based on topics addressed in the selected studies. The process of building the taxonomy was iterative, by identifying categories for each study, and then by grouping the categories in a hierarchical structure.

II. THE ENERGY TAXONOMY

The term energy and power are used alternatively in the studies although they are different physical values. Power is rate of consumption of energy. $P = I * V_{CC}$, where P → average power, I → average current, V_{CC} → supply voltage. The energy required for a computer execution is defined in a relation to instruction execution, i.e. $E = k*N * _{T}$, where E is energy, k is a factor (for simplicity reason k is set to 1 for relative measurements), N is number of clock cycles, $_{T}$ is clock period. In reality there are other sources of energy consumption and this is reflected in several papers discussing the energy dissipation.

The taxonomy for power/energy property of embedded system is given in Figure 1.

[1] Details about this SLR can be found in http://www.idt.mdh.se/utbildning/exjobb/files/TR1216.pdf

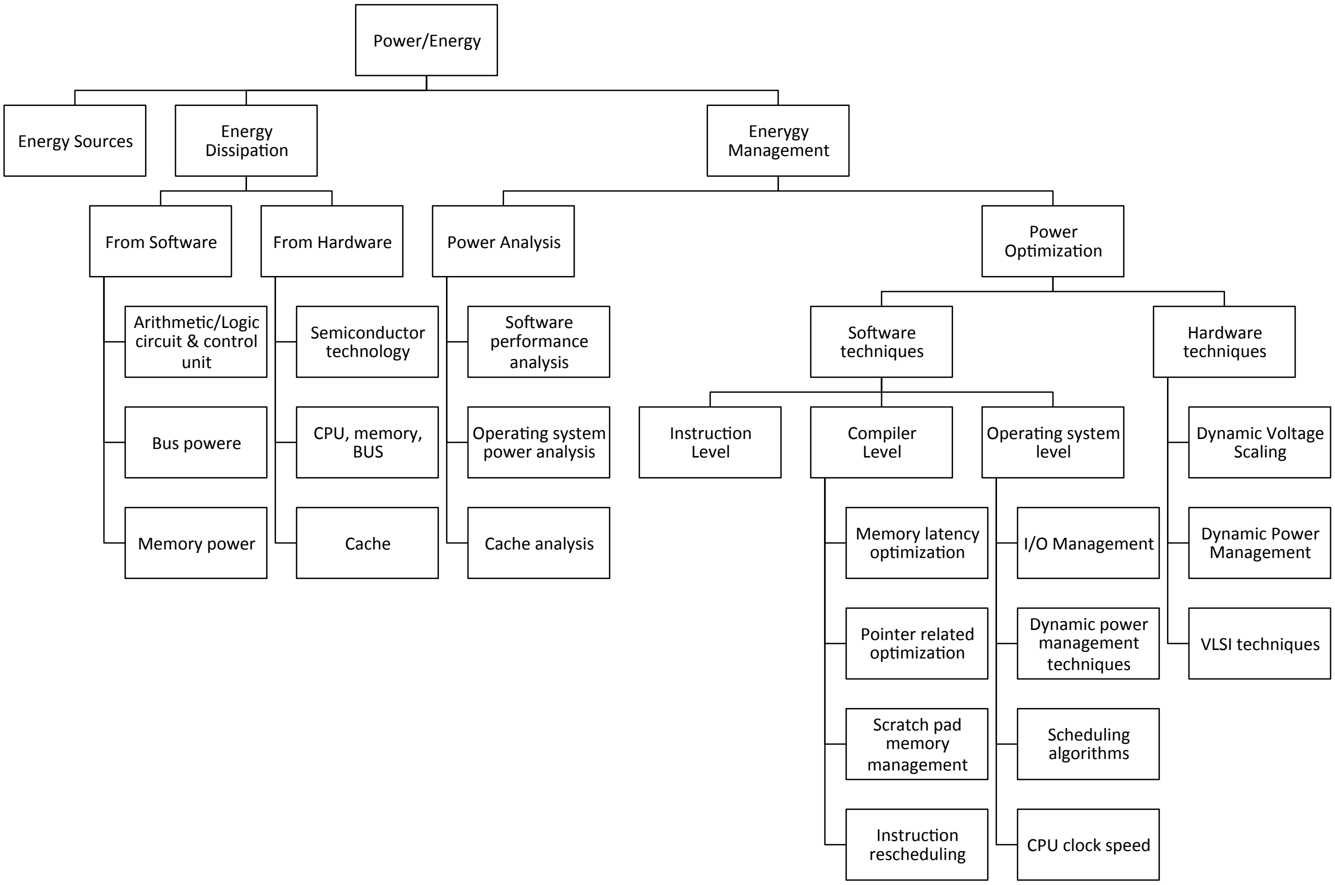

Figure 1 – Taxonomy for power/energy concerns in embedded system design

In the taxonomy we have defined the three main categories: (A) Energy sources, (B) energy dissipation and (C) energy management.

Below we provide a short description of each category and refer to primary studies that address it.

A. *Energy sources*

The source of energy refers to the origin of energy that is used for the operation and functioning of *the* embedded system. The generation of minimum energy from the power source at any instant should be higher than the maximum energy requirement of the energy utilizing system.

B. *Energy dissipation*

Energy dissipation refers to the expulsion of the energy from the embedded system. Energy dissipation occurs when the system consumes and utilizes the given energy for its functioning and operation. The following studies address the energy dissipation in general. The energy dissipation is analyzed from both the software and the hardware perspective of the computer system.

1) *Dissipation from Software.*

Here the power expelled from the software refers to the energy dissipated during the execution of software codes on (i) CPU, (ii) memory, and their (iii) corresponding busses. In addition the switching activity in the CPU causes energy dissipation. The switching activity in the CPU corresponds to load/store instructions, branch instructions and arithmetic instructions. Charging and discharging of the bus lines contributes to the energy dissipate n form the bus. Energy dissipated from the memory unit corresponds to the power amplifier and the charging and discharging of word-line and bit-line capacitances.

2) *Dissipation from Hardware.*

The studies refer to energy disssipation related to different types of semicondacror technologies (e.g. CMOS)in general, then related to particular hardware component. The total energy dissipated from the memory consists of energy dissipated from the (i) cache decoders, (ii) buses connecting the cache and main memory, (iii) cache cell array, and (iv) main memory. Cache energy dissipation is based on the energy consumed by row and column decoders, word-lines, bit-lines and sense amplifiers. The energy dissipated from the cache depends on the number of cache misses, number of cache accesses, configuration of the cache and the degree to which energy efficient implementation techniques are followed. The total amount of energy dissipated from a set-associative cache includes the energy dissipated from bit-lines, word-lines, output-lines, address input-lines. Energy dissipated from the bus is influenced by factors such as bus capacitance, number of transactions and switching activity of the bus. The energy dissipated from the CPU depends on the factors such as clock speed of the processor, idle time of the processor and the number of instructions executed per second.

C. *Energy management*

Managing energy production and consumption include various techniques that are aimed at extending the lifetime of the source of energy which results in longevity of the battery source. Centralized power management technique, dynamic power management technique, static power management technique are certain types of energy management techniques. The management of power is divided into

1) power analysis, and
2) power optimization.

1) *Power analysis.*

Analysis of power consumption includes various measurement, experimentation and estimation methodologies and techniques that are used to evaluate and validate the power consumption in a system. Power analysis is used to model the power consumption of the various components of a computer system. These components include processors, memory units and other vital parts of the computer system. This analysis is related to:

a) software performance,
b) operation systems,
c) use of memory cache.

a) Software performance

Software performance analysis is essential to assign different energy inputs to different performance levels of the software. High power consumption is a limiting factor for the performance of the computer system. Software may have different performance levels depending on the application. Thus, assigning the precise energy input helps to optimize the energy depending on the energy requirement of the software based on its context and application.

b) Operation systems

Operating systems power analysis is essential for process and resource scheduling, choosing between different options of energy efficient operating systems.

c) Cache

Cache power analysis is important since a large amount of energy consumed by CPU goes to cache memory as cache memory is an integral part of the processor.

Surprisingly enough, operation system analysis is not as present in the literature as expected.

2) *Power optimization.*

The power optimization techniques are aimed at minimizing the energy consumption, either as total energy, average energy consumptions, and minimum and maximum (peaks) consumption in time units. In particular use of power with minimum energy budget and availability.

The power optimization can be achieved through

a) software techniques, and
b) hardware techniques.

a) Software Techniques.

The software power optimization techniques are applied at the consumer level in order to minimize the energy consumption by means of software. Instruction reordering and by this a generation of energy efficient code are some of the examples for software power optimization.

The software optimization techniques include optimization on the (a.1) instruction level, (a.2) compiler level, and (a.3) operating system level.

a.1) Instruction level: The information obtained from instruction level analysis is used to allocate precise power budget to a particular program.

a.2) Compiler level: Since a compiler plays an essential role in the run time behavior of an application (determines the type, order and number of instructions executed), it has a large impact on the energy consumption. There are certain specific techniques used in a compiler to achieve lower energy consumption, mostly related to the synthesis and compiler optimization techniques. These are Memory latency optimization, pointer related optimization, scratch-pad memory optimization, and instruction rescheduling.

Latency reducing transformations and latency hiding transformations are two techniques for compiler level latency optimizations. The former technique modifies program access pattern and memory layout of variables. The latter technique optimizes data locality in order to minimize the number of access to particular level in the memory hierarchy that needs longer access time. Memory latency optimization has significant impact on the energy consumption.

Compiler level pointer related optimization is implemented in hardware and software level. The technique used in hardware level related to *pointer synthesis* and the technique used in software level is *data layout transformation*. The processes followed in data layout transformation are “splitting into individual arrays”, “clustering algorithm”, “cost computation”.

Scratch-pad memory is a type of high-speed on-chip memory. The techniques used for scratch-pad memory optimization are memory layout detection and loop transformation, memory space partitioning and code modifications, related to the energy consumption.

Instruction scheduling is a power saving compiler technique used to improve the code performance. Instruction rescheduling is used to remove pipeline stalls that are caused due to load delay, branch delay and delay slot. The basic ideology present in instruction rescheduling is to reorder the queue of instructions to reduce the hamming distance of consecutive instructions.

a.3) Operating system (OS) level. Power reduction through operating system is known as Task Based Power Management (TBPM). TBPM utilizes the information present in the operating system for managing power which is performed by dividing the requests according to the tasks that relates to OS processes. The OS process is present in one of its states such as new, running, waiting etc. A proper management of these states has significant impact on the energy consumption.

The same is valid for other parts of an operation system: Input output (I/O) management, Dynamic Power Management, Scheduling algorithms, and CPU clock speed management.

Input output (I/O) management is essential since it assists in the smooth transition from one OS process state to another.

Dynamic Power Management (DPM) is an operating system level technique used to conserve energy mainly by the process of resource shutdown. In DPM, power management process is decided during the run time in order to adapt to current situation of the system workload and available resources.

Scheduling algorithms are used to minimize the energy consumption of the input-output devices of a computer system. Examples of scheduling algorithms related to power optimizations are Low Energy Device Scheduler, energy-optimal device scheduler.

CPU clock speed variation is also a means to reduce the power dissipation.

b) Hardware Techniques

The hardware power optimization includes techniques directly applied on the hardware, independently on the operating system or application level.

Dynamic voltage scaling, dynamic power management are some of the hardware optimizations techniques. These techniques are usually implemented during the early design stage in the development of the system.

b.1) Dynamic voltage scaling (DVS) is a hardware optimization technique by which the supplied voltage and frequency is dynamically varied according to the workload of the processor.

b.2) Dynamic power management (DMP) is a technique which is used to monitor and control the power and performance levels of the digital circuits and digital systems using a power monitor in order to control the state transitions and to optimize the power consumption.

b.3) VLSI techniques. The various VLSI techniques used for power optimization are applied at different levels such as circuit level, logic level, architectural or behavioral level, system level and software level.

III. Conclusion and Future Work

We presented various techniques for energy consumption monitoring and energy saving. In a system usually several of these techniques are used simultaneously. In applying several techniques it is important to keep control of overall computer performance and accurate functioning to meet the specific critical requirements. Example: Delimitation of power supplied to CPU may result in increased time for switching of device-power state which may cause deadline miss for a task, thereby causing serious effects in a time/safety critical system. While there exists many publications that treat a particular aspect of energy or power management, there are quite few publications that provide a general overview of it. In [4] an overview of several techniques for energy management is presented, but this is only a small subset of all possible approaches.

The presented taxonomy is the first step in the analysis of power/energy concerns in embedded systems. The systematic review done was related to energy/power as an important extra-functional property. The next step in this research will be to provide a deeper analysis of the studies, and find the relations between different techniques, as well as the relation to other concerns. A new set of research questions, focused on particular aspects of energy saving would provide additional information and elements in the taxonomy. For example the techniques related to hardware and software can be separated and further analyzed. Also, different types of ES have different requirements. ES with resource constraints (typically small systems) have different requirements with regards to energy consumption than ES with no low energy constraints. Finally an important question is whether the techniques used in ES domain can be applied in other domains. Many principles can be reused directly while other would require certain adjustments, depending on the application types, and type of the platforms these applications are running on.

References

[1] K.V.K.K. Prasad, Embedded Real Time Systems: Concepts, Design Prog Bb, Dreamtech Press, Nov 12, 2003

[2] B.A. Kitchenham, “Procedures for Performing Systematic Reviews,” Technical Report TR/SE-0401, Keele University, and Technical Report 0400011T.1, NICTA, 2004.

[3] Umesh Balaji Kothandapani Ramesh, Séverine Sentilles, Ivica Crnkovic, Technical Report: Energy Management in Embedded Systems Taxonomy, http://www.mrtc.mdh.se/index.php?choice=publications&id=2769

[4] Massoud Pedram, Power optimization and management in embedded systems. In Proceedings of the 2001 Asia and South Pacific Design Automation Conference (ASP-DAC '01). ACM, New York, NY, USA, 239-244.