Technical Report: Energy Management in Embedded Systems 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 how to manage and optimize energy consumption in the computer systems. The aim of this paper is to present the energy concerns addressed in the research literature and build up a knowledge that can be utilized in other domains. Based on a systematic review, the paper presents 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). 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/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. For these reasons energy consumption and energy management is plying an important role in embedded systems design. 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.

The rest of the paper is organized as follows. Section 2 describes the systematic review process we have used, and our research questions; Section 3 provides the taxonomy and a brief explanation of it; Section 5 concludes the paper.

II. SYETMATIC REVIEW PROCESS

To obtain information from the research papers we adopted a Systematic Literature Review (SLR). 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 the following databases ACM, IEEE Xplore, and SpringerLink.

The selected references, aka primary studies, were the result of the SLR selection procedure, based on inclusion and exclusion criteria, reading the abstracts, and including new references using backward reference searching (studies cited in the selected studies). The total numbers of research papers obtained were 1138 from the prescribed databases. Out of the 1138 research papers, 95 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 95 papers the 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.

III. THE POWER 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

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

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. In the taxonomy we have defined the three main categories: (i) Energy sources, (ii) energy dissipation and (iii) energy management. Each category (except energy sources, which is not in the focus of this paper) we divided in a set of hierarchical subcategories.

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 [S42].

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 ([S4], [S7], [S11], [S14], [S16], [S17], [S19], [S20], [S20], [S24], [S33], [S35], [S49], [S53], [S67], [S70], [S75], [S79], [S81], [S89], [S94]) .

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 [S20], (ii) memory [S60], [S73], [S83], and their (iii) corresponding busses [S60], [S73]. 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 dissipation 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 [S20].

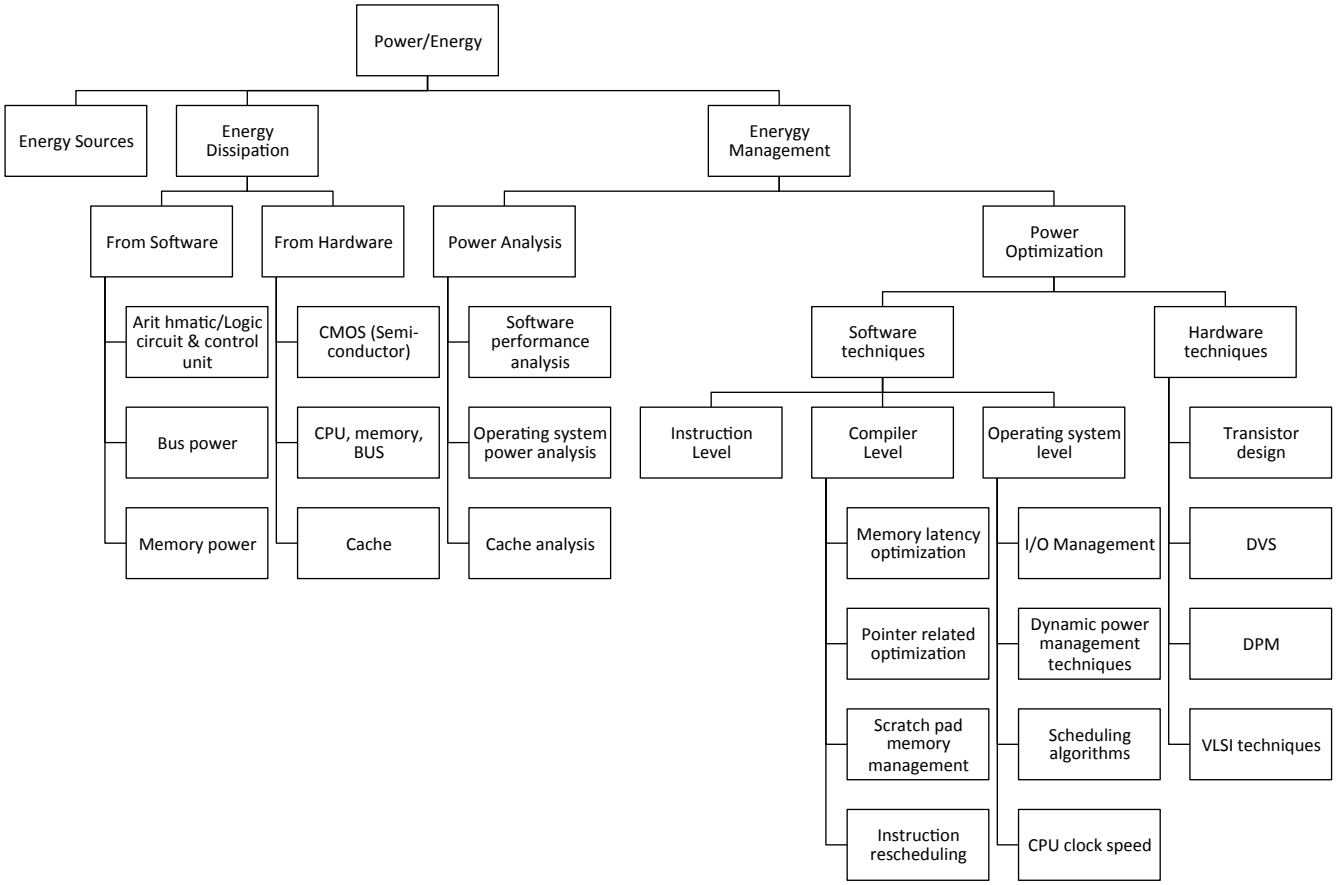

Figure 1 – Taxonomy for power/energy concerns in embedded system design

2) *Dissipation from Hardware.*

The energy dissipated from the CMOS circuits is summarized in equation below.

$$P_{\text{total}} = p_t(C_L \,.\, V \,.\, V_{dd} \,.\, f_{clk}) + I_{SC} \,.\, V_{dd} + I_{leakage} \,.\, V_{dd}$$

$C_L \rightarrow$ loading capacitance, $f_{clk} \rightarrow$ clock frequency, $p_t \rightarrow$ probability of occurrence of power dissipating transition, $V \rightarrow$ voltage swing, $V_{dd} \rightarrow$ supply voltage, $I_{SC} \rightarrow$ short circuit current, $I_{leakage} \rightarrow$ leakage current [S13], [S89].

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 [S39]. 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. Detailed information on dissipation of cache energy is explained in [S6], [S38], [S39],[S91].

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 [S9], [S60], [S74], [S77], [S87].

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 ([S8], [S18], [S19], [S24], [S28], [S42], [S48], [S46]).

The management of power is divided into (i) power analysis and (ii) 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 [S82].

This analysis is related to (i) software performance, (ii) operation systems, and in particular use of (iii) memory cache

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 [S3], [S57], [S66], [S65].

Operation systems power analysis is essential for process and resource scheduling, choosing between different options of energy efficient operating systems. [S1], [S19], [S43], [S54], [S57], [S79], [S86], [S82].

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. Power analysis related to cache is discussed in [S6], [S47], [S48], [S57], [S62], [S91].

Surprisingly enough, operation system analysis is not as present in the literature as expected [S79].

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 ([S25], [S30], [S36], [S54], [S39], [S69], [S82]).

The power optimization can be achieved through hardware techniques and software 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 [S25]. Instruction reordering and by this a generation of energy efficient code are some of the examples for software power optimization [S82].

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 [S51], [S82].

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, which is discussed in [S56], [S2], [S8], [S22], [S39]. 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.

Memory latency optimization. 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 ([S88], [S4], [S8], [S29], [S62], [S95], [S26], [S69], [S88]).

Pointer related optimization. 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". These techniques have impact on the energy consumption [S58].

Scratch-pad memory optimization. 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 ([S5], [S8], [S19], [S40], [S52] [S88]).

Instruction rescheduling. 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 ([S45], [S59], [S82]).

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 ([S2], [S32], [S90]).

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 [S90].

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 ([S15], [S25]).

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 [S39].

CPU clock speed variation is also a means to reduce the power dissipation [S90].

b) Hardware Techniques

The hardware power optimization includes techniques directly applied on the hardware, independently on the operating system or application level ([S21], [S52], [S60], [S76]).

Dynamic voltage scaling, dynamic power management are some of the hardware optimizations techniques ([S45], [S53], [S56]). These techniques are usually implemented during the early design stage in the development of the system.

Dynamic voltage scaling is a hardware optimization technique by which the supplied voltage and frequency is dynamically varied according to the workload of the processor ([S2], [S25], [S34], [S63], [S71], [S72], [S73]).

Dynamic power management 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 ([S8], [S15], [S25], [S71], [S77], [S84], [S93]).

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 ([S23], [S37], [S39], [S52]).

IV. 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.

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.

V. SLR REFERENCES

[S1] Aaraj, N., Raghunathan, A., & Jha, N. K. (2008). Analysis and design of a hardware/software trusted platform module for embedded systems. *ACM Transactions on Embedded Computing Systems*, *8*(1), 1-31.

[S2] AbouGhazaleh, N., Mosse, D., Childers, B., Melhem, R., & Craven, M. (2003). Collaborative operating system and compiler power management for real-time applications. RealTime and Embedded Technology and Applications Symposium 2003 Proceedings The 9th IEEE (Vol. 5, pp. 133-141).

[S3] Andrade, E., Maciel, P., Falcão, T., Nogueira, B., Araujo, C., & Callou, G. (2010). Performance and energy consumption estimation for commercial off-the-shelf component system design. Innovations in Systems and Software Engineering, 6(1), 107-114. Springer.

[S4] Arora, D., Ravi, S., Raghunathan, A., & Jha, N. K. (2007). Architectural Support for Run-Time Validation of Program Data Properties. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 15(5), 546-559.

[S5] Avissar, O., & Barua, R. (2002). An Optimal Memory Allocation Scheme for Scratch-Pad Based Embedded Systems. ACM Transactions on Embedded Computing Systems, 6-26.

[S6] Bartolini, S., & Prete, C. A. (n.d.). An object level transformation technique to improve the performance of embedded applications. Proceedings First IEEE International Workshop on Source Code Analysis and Manipulation (pp. 24-32). IEEE Comput. Soc.

[S7] Bastani, F., & Cooper, K. (2005). A Model and Methodology for Composition QoS Analysis of Embedded Systems. 11th IEEE Real Time and Embedded Technology and Applications Symposium (pp. 56-65).

[S8] Benini, L., Macii, A., & Poncino, M. (2003). Energy-aware design of embedded memories: A survey of technologies, architectures, and optimization techniques. ACM Transactions on Embedded Computing Systems, 2(1), 5-32.

[S9] L. Benini, A. Bogliolo, and G. D. Micheli. "Dynamic power management of electronic systems". In International Conference on Computer-Aided Design 1998, pages 696–702

[S10] Borde, E., Carlson, J., Feljan, J., Lednicki, L., Lévêque, T., Maras, J., Petricic, A., et al. (2011). PRIDE - An Environment for Component-Based Development of Distributed Real-Time Embedded Systems. 2011 Ninth Working IEEE/IFIP Conference on Software Architecture (pp. 351-354). IEEE.

[S11] Boulis, A., & Srivastava, M. B. (2003). Node-level energy management for sensor networks in the presence of multiple applications. Energy, 10(6), 737–746. Kluwer Academic Publishers.

[S12] Bueno, D., Conger, C., George, A. D., Troxel, I., & Leko, A. (2007). RapidIO for radar processing in advanced space systems. ACM Transactions on Embedded Computing Systems, 7(1), 1-38.

[S13] A. Chandrakasan, S. Sheng, and R. Brodersen, "Low-power CMOS digital design," IEEE J. Solid-State Circuits, vol. 27, pp. 473–484, Apr. 1992

[S14] Caspi, P., Folher, G., Garcia-Valls, M., Kopetz, H., Lakhnech, Y., Laroussinie, F., Lavagno, L., et al. (2005). Guidelines for a graduate curriculum on embedded software and systems. ACM Transactions on Embedded Computing Systems, 4(3), 587-611.

[S15] H. Cheng and S. Goddard, "Online energy-aware I/O device scheduling for hard real-time systems," in Proceedings of Design Automation and Test in Europe (DATE 2006), pp. 1055–1060

[S16] Cloth, L., & Haverkort, B. R. Quantitative Evaluation in Embedded System Design: Predicting Battery Lifetime in Mobile Devices. 2008 Design, Automation and Test in Europe (pp. 90-91).

[S17] Corrêa, U. B., Lamb, L., Carro, L., Brisolara, L., & Mattos, J. (2010). Towards Estimating Physical Properties of Embedded Systems using Software Quality Metrics. 2010 10th IEEE International Conference on Computer and Information Technology (pp. 2381-2386)..

[S18] Courbot, A., Grimaud, G., & Vandewalle, J.-J. (2010). Efficient Off-board Deployment and Customization of Virtual Machine Based Embedded Systems. ACM Transactions on Embedded Computing Systems.

[S19] Coussy, P., Casseau, E., Bomel, P., Baganne, A., & Martin, E. (2006). A formal method for hardware IP design and integration under I/O and timing constraints. ACM Transactions on Embedded Computing Systems, 5(1), 29-53.

[S20] V. Dalal and C.P. Ravikumar, "Software Power Optimizations in an Embedded System," Proc. 14th Int'l Conf. VLSI Design (VLSI 01), IEEE CS Press, 2001, pp. 254-259

[S21] Damez, L., Sieler, L., Landrault, A., & Dérutin, J. P. (2011). Embedding of a real time image stabilization algorithm on a parameterizable SoPC architecture a chip multi-processor approach. Journal of Real-Time Image Processing, 6(1), 47-58.

[S22] Dean, A. G. (2006). Software thread integration for embedded system display applications. ACM Transactions on Embedded Computing Systems, 5(1), 116-151.

[S23] Devadas, S., Malik, S.: "A survey of optimization techniques targeting low power VLSI circuits". In: Proceedings of the 32nd ACM/IEEE Conference on Design Automation. (1995) 242 – 247

[S24] Doblander, A., Zoufal, A., & Rinner, B. (2009). A novel software framework for embedded multiprocessor smart cameras. ACM Transactions on Embedded Computing Systems, 8(3), 1-30.

[S25] Fei, Y., Zhong, L., & Jha, N. K. (2008). An energy-aware framework for dynamic software management in mobile computing systems. ACM Transactions on Embedded Computing Systems, 7(3), 1-31.

[S26] Geelen, B., Ferentinos, V., Catthoor, F., Lafruit, G., Verkest, D., Lauwereins, R., & Stouraitis, T. (2010). Modeling and exploiting spatial locality trade-offs in wavelet-based applications under varying resource requirements. ACM Transactions on Embedded Computing Systems, 9(3), 1-26.

[S27] Girault, A., Nicollin, X., & Pouzet, M. (2006). Automatic rate desynchronization of embedded reactive programs. ACM Transactions on Embedded Computing Systems, 5, 687-717.

[S28] Gorjiara. B, Bagherzadeh. N, Pai Chou. "Integrating Power Management into Distributed Real-time Systems at Very Low Implementation Cost", Design Automation Conference, 2007. ASP-DAC '07. Asia and South Pacific, Publication Year: 2007, Page(s): 872 – 877

[S29] Grun, P., Dutt, N., & Nicolau, A. (2003). Access pattern-based memory and connectivity architecture exploration. Transactions on Embedded Computing Systems, 2(1), 33-73.

[S30] Gurun, S., Krintz, C., & Wolski, R. (2008). NWSLite. ACM Transactions on Embedded Computing Systems, 7(3), 1-36.

[S31] Heng-Ruey, H., Jian-Jia, C., & Tei-Wei, K. (2006). Multiprocessor Synthesis for Periodic Hard Real-Time Tasks

under a Given Energy Constraint. Proceedings of the Design Automation & Test in Europe Conference (pp. 1-6)..

[S32] Hsin-hung, L., & Chih-Wen, H. (2006). COS: A Configurable OS for Embedded SoC Systems. 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'06) (pp. 242-245).

[S33] Hu, J., Li, F., Degalahal, V., Kandemir, M., Vijaykrishnan, N., & Irwin, M. J. (2009). Compiler-assisted soft error detection under performance and energy constraints in embedded systems. ACM Transactions on Embedded Computing Systems, 8(4), 1-30. ACM Press.

[S34] Im, C., Ha, S., & Kim, H. (2004). Dynamic voltage scheduling with buffers in low-power multimedia applications. ACM Transactions on Embedded Computing Systems, 3(4), 686-705.

[S35] Islam, S., & Lindstrom, R. (2006). Dependability Driven Integration of Mixed Criticality SW Components. Ninth IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC'06) (pp. 485-495).

[S36] Jacob, B., & Bhattacharyya, S. (2003). Introduction to the two special issues on memory. ACM Transactions on Embedded Computing Systems, 2(1), 1-4.

[S37] Jones, A. K., Hoare, R., Kusic, D., Mehta, G., Fazekas, J., & Foster, J. (2006). Reducing power while increasing performance with supercisc. ACM Transactions on Embedded Computing Systems, 5(3), 658-686.

[S38] Kamble, M. B. and Ghose, K. "Analytical energy dissipation models for low power caches," In Proceedings of International Symposium on Low Power Electronics and Design, ACM, Year 1997, pages 143–148

[S39] Kadayif, I., Kandemir, M., Chen, G., Vijaykrishnan, N., Irwin, M. J., & Sivasubramaniam, A. (2005). Compiler-directed high-level energy estimation and optimization. Trans on Embedded Computing Sys, 4(4), 819-850.

[S40] Kandemir, M.; Ramanujam, J.; Irwin, M.J.; Vijaykrishnan, N.; Kadayif, I.; Parikh, A, "Dynamic management of scratch-pad memory space," In proceedings of Design Automation Conference, IEEE, Page(s): 690 – 695, Publication Year: 2001

[S41] Kangas, T., Kukkala, P., Orsila, H., Salminen, E., Hännikäinen, M., Hämäläinen, T. D., Riihimäki, J., et al. (2006). UML-based multiprocessor SoC design framework. ACM Transactions on Embedded Computing Systems, 5(2), 281-320..

[S42] Kansal, A., Hsu, J., Zahedi, S., & Srivastava, M. B. (2007). Power management in energy harvesting sensor networks. ACM Transactions on Embedded Computing Systems, 6(4), 32-es. ACM.

[S43] Katoen, J.-P. (2008). Quantitative Evaluation in Embedded System Design: Trends in Modeling and Analysis Techniques. 2008 Design, Automation and Test in Europe (pp. 86-87)..

[S44] Keutzer, K., Newton, A. R., Rabaey, J. M., & Sangiovanni-Vincentelli, A. (2000). System-level design: orthogonalization of concerns and platform-based design. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 19(12), 1523-1543.

[S45] Kim, M., Banerjee, S., Dutt, N., & Venkatasubramanian, N. (2008). Energy-aware cosynthesis of real-time multimedia applications on MPSoCs using heterogeneous scheduling policies. ACM Transactions on Embedded Computing Systems, 7(2), 1-19.

[S46] Klues, K., Xing, G., & Lu, C. (2010). Link Layer Driver Architecture for Unified Radio Power Management in Wireless Sensor Networks. ACM Trans Embed Comput Syst, 9(4), 41.

[S47] Krishnaswamy, A., & Gupta, R. (2005). Dynamic coalescing for 16-bit instructions. Trans on Embedded Computing Sys, 4(1), 3-37. ACM Press.

[S48] D. Kudithipudi, S. Petko, and E. John, "Cache leakage power analysis in embedded applications," in Proceedings of the 47th Midwest Symposium on Circuits and Systems, vol. 2, pp. II–517–II–520vol.2, 25-28 July 2004

[S49] Le Nours, S., Barreteau, A., & Pasquier, O. (2011). A state-based modeling approach for fast performance evaluation of embedded system architectures. 2011 22nd IEEE International Symposium on Rapid System Prototyping (pp. 156-162). IEEE.

[S50] Lloyd, S., & Snell, Q. (2009). A packet-switched network architecture for reconfigurable computing. ACM Transactions on Embedded Computing Systems TECS, 9(1), 7. ACM.

[S51] Loghi, M., Benini, L., & Poncino, M. (2007). Power macromodeling of MPSoC message passing primitives. ACM Trans Embed Comput Syst, 6(4), 31.

[S52] Mathew, B., & Davis, A. (2004). A loop accelerator for low power embedded VLIW processors. Hardware/Software Codesign and System Synthesis, 2004. CODES + ISSS 2004. International Conference on, 6-11.

[S53] McLoughlin, I. V., & Bretschneider, T. R. (2010). Reliability through redundant parallelism for micro-satellite computing. ACM Transactions on Embedded Computing Systems, 9(3), 1-25.

[S54] Mishra, P., Mamidipaka, M., & Dutt, N. Processor-memory coexploration using an architecture description language. ACM Transactions on Embedded Computing Systems, 3(1), 140-162.

[S55] Niu, L. (2011). Energy efficient scheduling for real-time embedded systems with QoS guarantee. Real-Time Systems, 47(2), 75-108. Springer Netherlands.

[S56] D. A. Ortiz and N. G. Santiago, "High-level optimization for low power consumption on microprocessor-based systems," in MWSCAS 2007. IEEE Computer Society, 2007, pp. 1265–1268

[S57] Otoom, M., & Paul, J. M. (2011). Workload Mode Identification for Chip Heterogeneous Multiprocessors. International Journal of Parallel Programming, 40(2), 184-224. Springer Netherlands.

[S58] P. R. Panda, L. Semeria, and G. de Micheli, "Cache-efficient Memory Layout of Aggregate Data Structures," Proceedings of the 14th international symposium on Systems synthesis, pp. 101 - 106, Montreal, Quebec, Canada, October 1 - 3, 2001.

[S59] M. Pedram, "Power optimization and management in embedded systems", In Proceedings of the Asia and South Pacific Design Automation Conference, pages 239–244, ACM, 2001

[S60] Puttaswamy, K., Kyu-Won, C., Jun Cheol, P., Mooney, V. J., Chatterjee, A., & Ellervee, P. (2002). System level power-performance trade-offs in embedded systems using voltage and frequency scaling of off-chip buses and memory. System Synthesis, 2002. 15th International Symposium on, 225-230. Retrieved from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1227182

[S61] Quan, G., & Hu, X. S. (2007). Energy Efficient DVS Schedule for Fixed-Priority Real-Time Systems. ACM Transactions on Design Automation of Electronic Systems, 6, 1-30.

[S62] Rajeev, K., Mahlke, S., & Austin, T. Memory system design space exploration for low-power, real-time speech recognition. Hardware/Software Codesign and System Synthesis, 2004. CODES + ISSS 2004. International Conference on, 140-145.

[S63] R. Racu, A. Hamann, R. Ernst, B. Mochocki, and X. Hu, "Methods for power optimization in distributed embedded systems with real-time requirements," In Proceedings of the 2006 international conference on Compilers, architecture and synthesis for embedded systems. ACM, 2006, p. 388

[S64] Rakhmatov, D., & Vrudhula, S. (2003). Energy management for battery-powered embedded systems. ACM Transactions on Embedded Computing Systems, 2(3), 277-324. ACM Press.

[S65] Rao, R., & Vrudhula, S. (2007). Energy optimal speed control of a producer--consumer device pair. ACM Transactions on Embedded Computing Systems, 6(4), 30-es. ACM.

[S66] Rauwerda, G. K., Heysters, P. M., & Smit, G. J. M. (2004). Mapping Wireless Communication Algorithms onto a Reconfigurable Architecture. The Journal of Supercomputing, 30(3), 263-282.

[S67] Ravi, S., Raghunathan, A., Kocher, P., & Hattangady, S. Security in embedded systems: Design challenges. ACM Transactions on Embedded Computing Systems TECS, 3(3), 461-491.

[S68] Richling, J., & Malek, M. (2006). Message Scheduled System (MSS): A Composable Architecture for Distributed Real-Time Systems. Model-Based Testing, ITGA FA 6.2 Workshop on and GI/ITG Workshop on Non-Functional Properties of Embedded Systems, 2006 13th GI/ITG Conference -Measuring, Modelling and Evaluation of Computer and Communication (MMB Workshop), 1-11. Retrieved from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5755438

[S69] Rodric M. Rabbah, K. V. P. (2003). Data remapping for design space optimization of embedded memory systems. ACM Transactions in Embedded Computing Systems, 2, 186--218. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?

[S70] Schliecker, S., & Ernst, R. (2010). Real-time performance analysis of multiprocessor systems with shared memory. ACM Transactions on Embedded Computing Systems, 10(2), 1-27.

[S71] Schmitz, M. T., Al-Hashimi, B. M., & Eles, P. (2003). A co-design methodology for energy-efficient multi-mode embedded systems with consideration of mode execution probabilities. 2003 Design, Automation and Test in Europe Conference and Exhibition (pp. 960-965). IEEE Comput. Soc.

[S72] Seth, K., Anantaraman, A., Mueller, F., & Rotenberg, E. (2003). FAST : Frequency-Aware Static Timing Analysis. RTSS, 3(1), 40-51. IEEE Comput. Soc. doi:10.1109/REAL.2003.1253252

[S73] L. Shang, L.-S. Peh, and N. K. Jha, "Dynamic voltage scaling with links for power optimization of interconnection networks", In Proc. Intl. Symp. on High-Performance Computer Architecture, California, Jan. 2003, pp. 79-90

[S74] Shim, H., Joo, Y., Choi, Y., Lee, H. G., & Chang, N. (2003). Low-energy off-chip SDRAM memory systems for embedded applications. ACM Transactions on Embedded Computing Systems, 2(1), 98-130.

[S75] Sutter, B. D., Put, L. V., Chanet, D., Bus, B. D., & Bosschere, K. D. (2007). Link-time compaction and optimization of ARM executables. ACM Transactions on Embedded Computing Systems, 6(1), 5.

[S76] Swahn, B., & Hassoun, S. (2003). Hardware Scheduling for dynamic adaptability using external profiling and hardware threading. Computer Aided Design, 2003 International Conference on. ICCAD-2003, 58-64.

[S77] Swaminathan, V., & Chakrabarty, K. (2002). Pruning-based energy-optimal device scheduling for hard real-time systems. Proceedings of the Tenth International Symposium on Hardware/Software Codesign. CODES 2002 (IEEE Cat. No.02TH8627) (pp. 175-180).

[S78] Swaminathan, V., and Chakrabarty, K, "Energy-conscious, deterministic I/O device scheduling in hard real-time systems", IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems, vol 22, pages 847.858, July 2003

[S79] Taha, S., Radermache, A., Gerard, S., & Dekeyser, J.-L. An Open Framework for Detailed Hardware Modeling. 2007 International Symposium on Industrial Embedded Systems (pp. 118-125).

[S80] Tan, T. K., Raghunathan, A., & Jha, N. K. (2005). Energy macromodeling of embedded operating systems. ACM Transactions on Embedded Computing Systems, 4(1), 231-254.

[S81] Tavli, B., Bicakci, K., Zilan, R., & Barcelo-Ordinas, J. M. (2011). A survey of visual sensor network platforms. Multimedia Tools and Applications, 1-38. Springer Netherlands. doi:10.1007/s11042-011-0840-z

[S82] V. Tiwari, S. Malik, and A.Wolfe, "Power analysis of embedded software: a first step towards software power minimization," IEEE Trans. on VLSI Systems, vol. 2, no. 4, pp. 437–445, Dec. 1994.

[S83] Trajkovic, J., Veidenbaum, A. V., & Kejariwal, A. (2008). Improving SDRAM access energy efficiency for low-power embedded systems. ACM Transactions on Embedded Computing Systems, 7(3), 1-21. ACM.

[S84] Wagner, F. R., Cesário, W., & Jerraya, A. A. (2007). Hardware/software IP integration using the ROSES design environment. ACM Transactions on Embedded Computing Systems, 6(3), 17-es.

[S85] Walther, K., & Nolte, J. (2006). Event-Flow and Synchronization in Single Threaded Systems. Model-Based Testing, ITGA FA 6.2 Workshop on and GI/ITG Workshop on Non-Functional Properties of Embedded Systems, 2006 13th GI/ITG Conference - Measuring, Modelling and Evaluation of Computer and Communication (MMB Workshop), 1-8. Retrieved from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5755434

[S86] Wehrmeister, M. A., Pereira, C. E., & Becker, L. B. (2006). Optimizing the Generation of Object-Oriented Real-Time Embedded Applications Based on the Real-Time Specification for Java. Proceedings of the Design Automation & Test in Europe Conference (pp. 1-6).

[S87] Weiser, M., Welch, B., Demers, A., and Shenker, S. "Scheduling for reduced CPU energy". In Proceedings of the First USENIX Symposium on Operating System Design and Implementation (OSDI), pages 13–23, Monterey, CA, November 1994

[S88] W. Wolf, M. Kandemir,"Memory System Optimization of Embedded Software," Proceedings of the IEEE, Vol. 91, No. 1, pp. 165-182, January 2003.

[S89] Yun, & Kim, J. (2003). On energy-optimal voltage scheduling for fixed-priority hard real-time systems. ACM Transactions on Embedded Computing Systems TECS, 2(3), 393-430.

[S90] Yung-Hsiang Lu; Benini, L.;De Micheli, G.; "Operating-system directed power reduction", In Proceedings of the '00 International Symposium on Low Power Electronics and Design, 2000. ISLPED '00. IEEE, Publication Year: 2000, Page(s): 37 – 42

[S91] Zhang, C., Vahid, F., & Najjar, W. (2005). A highly configurable cache for low energy embedded systems. ACM Transactions on Embedded Computing Systems, 4(2), 363-387. ACM.

[S92] Zhang, W., & Allu, B. (2007). Reducing branch predictor leakage energy by exploiting loops. ACM Transactions on Embedded Computing Systems, 6(2), 11-es.

[S93] Zhang, Y., & Chakrabarty, K. Dynamic adaptation for fault tolerance and power management in embedded real-time systems. ACM Transactions on Embedded Computing Systems, 3(2), 336-360.

[S94] Zhou, X., & Petrov, P. (2008). Direct address translation for virtual memory in energy-efficient embedded systems. ACM Transactions on Embedded Computing Systems, 8(1), 1-31.

[S95] Zhuang, X., & Pande, S. (2007). Power-efficient prefetching for embedded processors. ACM Transactions on Embedded Computing Systems, 6(1), 3.