International Journal of Management, IT & Engineering
Vol. 6 Issue 9, September 2016,
ISSN: 2249-0558 Impact Factor: 6.269
Journal Homepage: http://www.ijmra.us, Email: editorijmie@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

RUN-TIME MANAGEMENT FOR MULTICORE EMBEDDED SYSTEMS WITH ENERGY HARVESTING (SOLAR ENERGY)

Mr. Praveenkumar prabhakar pawar*

Abstract-

Through energy harvesting system, new energy sources are made available immediately for many advanced applications based on environmentally embedded systems. However, the harvested power, such as the solar energy, varies significantly under different ambient conditions, which in turn affects the energy conversion efficiency. There are two approach for designing power-adaptive computing systems to maximize the energy utilization under variable solar power supply, which I will b discussing in this paper. First paper "Power Adaptive Computing System for Solar-Energy-Powered Embedded Systems" uses the power adaptive computing systems to maximize the energy utilization under variable solar power supply. Here the geometric programming technique is exploited which can generate a customized parallel computing structure effectively. Second paper "Run-Time Management for Multicore Embedded Systems With Energy Harvesting" proposes a novel semidynamic algorithm (SDA)-based framework with energy budgeting that manages energy and workload allocation at runtime for multicore embedded systems with solar energy harvesting capability**.**

* **VTU**

I. Introduction

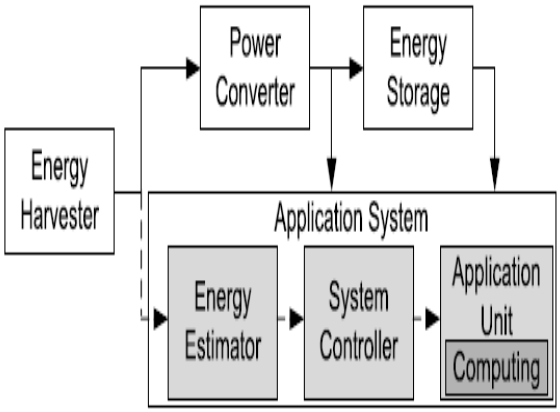


Fig. 1. Energy harvesting embedded system model.

Energy harvesting powered system needs to adapt to the unstable nature of the ambient energy. Power adaptability optimizes the system behavior according to the changing external power supply and thus allows the system to work in a long duration [5]. Fig. 1 shows a model of the energy harvesting embedded systems [1], [2]. The energy harvester converts ambient energy to electricity. The power converter extracts power from the harvester and performs ac–dc or dc–dc conversion with the goal of transferring as much power as possible to the energy storage or application system. The energy storage buffers the harvested energy temporally and supplies energy to the application system, the embedded application-specific functional unit. If the harvested energy exceeds the energy consumed by the application system, the system is powered directly by the harvester and the surplus will charge the energy storage.

The design goal of the power-adaptive computing systems is to maximize the system performance, under the constraint that the power consumption does not exceed the amount of power supply available. This constraint is also known as the energy neutral mode [5], where the energy consumption rate has to match with the availability of the power supply.

$$\begin{aligned} \min \quad & T_{\text{exe}}(\vec{x}) \\ \text{s.t.} \quad & P_C(\vec{x}) \le P_s(t) \\ & T_{\text{exe}}(\vec{x}) \le T_{\text{req}} \end{aligned} \qquad (\textbf{\textit{P1}})$$

where $Pc(\vec{x})$ is system power consumption, $T\text{exe}(\vec{x})$ is system execution time, Treq is execution time requirement, and $Ps(t)$ is system power supply changing over time. (*p1)* means that the system can run as fast as possible, while the power consumption constraint is not violated.

II. Adaptive Power Management

The basic idea of these approaches is to switch the system to different operation modes according to different workloads. The target applications of these approaches are those with varying run-time workloads. the target of the power-adaptive computing systems is the application scenarios, where the power supply varies over time. the systems with varying power supply have a more demanding design requirement. Works on adaptive power management in energy harvesting systems were presented in [2], [5]. There are two key elements for the adaptive power management: power modulation technique and power adaptation strategy. While the power adaptation strategies determine when and in which mode a system should work according to the harvested energy,

the power modulation techniques regulate the system power consumption so that a system can work in multiple power modes.

Power modulation techniques such as duty cycling, dynamic voltage and frequency scaling (DVFS), power gating, and clock gating are widely used. Duty cycling changes the active time of the system components to adjust the power consumption. The DVFS technique controls system power consumption by adjusting the supply voltage and frequency. High-priority tasks execute with the required voltage and frequency. Through enabling/disabling functional units and their clock sources, clock gating technique regulates the system power consumption. These three techniques affect only the system dynamic power profile. The power gating technique modulates

both the dynamic and static power by powering ON/OFF parts of the systems. According DVFS and power gating introduce time delay between power mode transitions. In addition, DVFS has a limited dynamic range due to the system operational restrictions on clock frequencies and supply voltages. Power gating implemented on the top of clock gating can further increase the power modulation range, leading to more effective power-adaptive approach.

III. Computing unit

In the computing structure, system operations were divided into three steps: data input, computation, and data output. In the data input step, operand data processed by all Pus were loaded into corresponding on-chip memories from the off-chip global memory. Each datum was loaded only once. In the computation step, all PUs with the same functionality processed different data in parallel. The structure of PUs was customized for the target applications, and operations in each PU were pipelined. In the output step, computation results were transferred back to the off-chip memory. All three steps were pipelined. Parallel computing structure was defined based on the geometric programming technique. the peak power of the harvested solar energy, was considered to ensure the designed computing unit work reliably in that environment. Here the execution of the parallel computing system is a pipelining of data input, computation, and data output. Each processing units

performs only arithmetic computation and could contain adders/subtracters, multipliers or comparators for the target applications and only access its own local memory

IV. Run-Time Management using SDA

In the second paper the author Yi Xiang propose a novel semidynamic algorithm (SDA)-based framework with energy budgeting that manages energy and workload allocation at runtime for multicore embedded systems with solar energy harvesting capability. The novelty and main contributions of this paper
are summarized as follows.

1) SDA reacts to runtime energy shortages and fluctuations proactively to find greater scope for energy savings, especially in multicore platforms.

2) A hybrid energy storage system is designed to decouple the runtime management scheme from variations in energy harvesting, as well as to enhance charging/discharging efficiency.

3) The energy and task distribution heuristics in SDA take system heterogeneity into consideration by assigningworkloads with an awareness of variations due to within-die process variations.

4) At the core level, a novel dual-speed frequency selection method is deployed to combine two neighboring discretefrequency levels for superior energy efficiency with an awareness of voltage/frequency switching overhead.

5) This framework cooperates with basic throttling mechanisms to tackle processor overheating. In addition, it dynamically reallocates workload or shuts down cores for more proactive multilevel throttling to reduce the occurrences and overhead of system overheating. Overheating the systems leads to the sudden shutdown or some application may not work or run properly, this framework handles overheating intelligently.

V. System Model

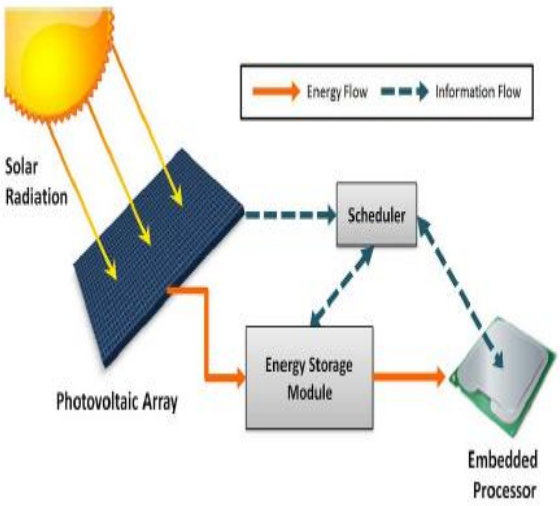


Fig.2. Real-Time embedded processing with solar energy harvesting.

A PV array is used as a power source for our embedded system, converting ambient solar energy into electric power. Naturally, the amount of harvested power varies over time due to changing environmental conditions, like angle of sunlight incidence, cloud density, temperature, and humidity. To cope with the unstable nature of the solar energy source, rechargeable batteries and supercapacitors can be used to buffer solar energy collected by PV cells. The capacity of the energy storage device is limited and harvested energy will be wasted if the energy storage device is already fully charged. We consider an embedded system with a low power multicore processor that has a support for task preemption. We assume that the frequency of each core can be adjusted individually (i.e., the processor possesses per-core DVFS capability)

The utilization of a periodic task (*U*) is defined with respect to the full speed (maximum frequency) provided by the processor. A task's utilization is its execution time under the maximum frequency divided by its period

$$U_i = \frac{C_i / f_{\max}}{T_i}. \qquad \textbf{P2}$$

The utilization for an entire task set is simply the accumulation of the utilization for all the tasks in the set. In preemptive real-time systems, a task set is schedulable by the EDF algorithm for a frequency *j* if it meets the following condition:

$$U_{\text{total}} \leq \frac{f_j}{f_{\max}}. \qquad \textbf{(P3)}$$

we consider thermal management in an energy harvesting multicore processing environment. We assume that each core in the multicore processor has a digital thermal sensor (DTS) implemented to monitor runtime temperature independently. We set 85 °C as the thermal *setpoint* at which throttling is initiated to halt all processor executions (i.e., throttling threshold = 85 °C). When throttling is triggered, a core must halt execution and shift to idle state until its temperature drops to 80 °C.

Run-Time scheduler module is an important component of the system for information gathering and execution control. The scheduler dynamically gathers information by monitoring the energy storage medium and multicore processor state (Fig. 2). The gathered data, together with offline-profiled information about task execution times and energy consumption on cores informs a management algorithm in our scheduler that coordinates operation of the multicore platform at runtime. Each core is eventually assigned strategy by the scheduler to guide intracore task execution.

VI. Semidynamic Algorithm Overview

One of the underlying ideas behind SDA is to exploit time segmentation during energy management. At each specified time interval, there is a reschedule point, where the execution strategy can be adjusted based on the energy budget provided by the energy storage system. A time frame between two reschedule points is called a schedule window, within which the strategy specified at the prior reschedule point is in effect until the next reschedule point. Thus, reschedule points provide dynamic adaptively needed by the energy harvesting aware system to adjust the task execution strategy, while the schedule window enables stable execution that utilizes periodic task information for better energy efficiency. It can be observed that under low energy conditions, SDA maintains execution at an optimal low (critical) frequency with different number of cores activated. Cores only execute at higher frequency when the energy harvested is abundant. In this manner, SDA can provide a better execution efficiency to improve performance under variable solar radiance conditions.

At each reschedule point, we update the execution strategy for the upcoming schedule window with a rescheduling scheme composed of three stages.

1) Energy Budgeting: This stage estimates the energy budget available for the upcoming schedule window based on the status of the hybrid energy storage system. Estimating the energy budget decouples runtime system management from energy variations in the environment, making it possible to deduce a stable balanced execution strategy that maximizes energy efficiency.

2) Workload Estimation: This second stage evaluates the amount of workload that can be supported by the energy budget and forks into two separate paths.

3) Task Rejection and Allocation: Based on the amount of workload estimated by the previous stage, this stage takes the periodic task set and filters out the subset of tasks that are less important. The remaining tasks are accepted for execution and are allocated to cores with an awareness of core heterogeneity.

VII. Hybrid Energy Storage System

hybrid energy storage system and its management policy, determines the energy budget for the upcoming schedule window, thereby isolating runtime task scheduling from fluctuations in solar energy harvesting.

The proposed hybrid energy storage system with one Li-ion battery and two separate supercapacitors connected by a dc bus. During each schedule window, one capacitor is used to collect energy extracted from the PV array, while the other one is used as a power source for system operation or battery charging. At each reschedule point, the two supercapacitors switch their roles. Supercapacitors

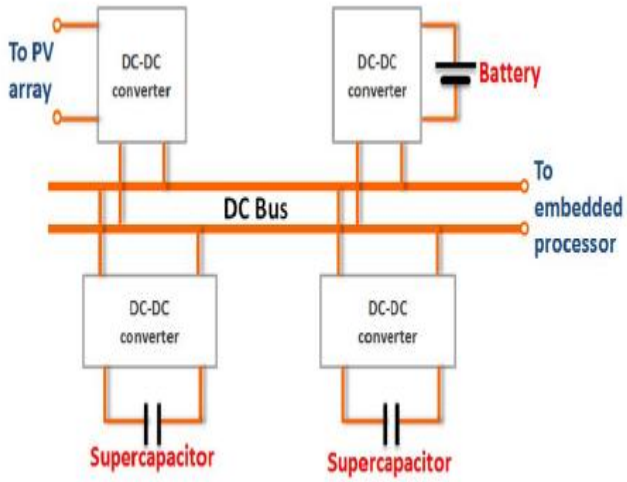
charge the battery only when their saved energy exceeds the peak requirements of processors running at full speed.

The PV array, battery, and supercapacitors are coupled with bidirectional dc–dc converters to serve the purpose of voltage conversions between components with maximum power point tracking and voltage level compatibility. This hybrid battery and dual-supercapacitor design has several advantages over a nonhybrid system.

1) The supercapacitors can support embedded processors directly, taking advantage of a much lower charging/ discharging overhead compared with a battery.

2) The electrochemical battery offers high capacity to preserve energy especially in scenarios with excessive harvested energy. On the other hand, the capacity requirement of supercapacitors is much smaller.

3) The supercapacitor with energy buffered during the last schedule window acts as a known stable energy source for the system in the upcoming schedule window. Thus, our energy budgeting does not require energy harvesting power predication.



Fi.3. Hybrid energy storage system

References

[1] V. Raghunathan and P. H. Chou, "Design and power management of energy harvesting embedded systems," in *Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, Oct. 2006, pp. 369–374.

[2] C. Moser, L. Thiele, D. Brunelli, and L. Benini, "Adaptive power management for environmentally powered systems," *IEEE Trans. Comput.*, vol. 59, no. 4, pp. 478–491, Apr. 2010.

[3] C. Lu, V. Raghunathan, and K. Roy, "Micro-scale energy harvesting: A system design perspective," in *Proc. 15th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2010, pp. 89–94.

[4] A. Khaligh, P. Zeng, and C. Zheng, "Kinetic energy harvesting using piezoelectric and electromagnetic technologies—State of the art," *IEEE Trans. Ind. Electron.*, vol. 57, no. 3, pp. 850–860, Mar. 2010.

[5] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 6, no. 4, p. 32, Sep. 2007.

[6] E. Humenay, D. Tarjan, and K. Skadron, "Impact of process variations on multicore performance symmetry," presented at the Conf. Design, Autom. Test Eur., San Jose, CA, USA, Apr. 2007, pp. 1–6.

[7] C. Li, W. Zhang, C.-B. Cho, and T. Li, "SolarCore: Solar energy driven multi-core architecture power management," presented at the Int. Symp. High Perform. Comput. Archit., San Antonio, TX, USA, 2011,pp. 205–216.

[8] X. Lin, Y. Wang, D. Zhu, N. Chang, and M. Pedram, "Online fault detection and tolerance for photovoltaic energy harvesting systems," presented at the IEEE/ACM Int. Conf. Comput.-Aided Design, San Jose,CA, USA, Nov. 2012, pp. 1–6.

[9] Y. Zhang, Y. Ge, and Q. Qiu, "Improving charging efficiency with workload scheduling in energy harvesting embedded systems," presented at the 50th ACM/EDAC/IEEE Design Autom. Conf. (DAC), Austin, TX,USA, May/Jun. 2013, pp. 1–8.

[10] C. Moser, D. Brunelli, L. Thiele, and L. Benini, "Lazy scheduling for energy harvesting sensor nodes," presented at the Conf. Distrib. Parallel Embedded Syst., Braga, Portugal, 2006, pp. 125–134.

[11] S. Liu, Q. Qiu, and Q. Wu, "Energy aware dynamic voltage and frequency selection for real-time systems with energy harvesting," presented at the Conf. Design, Autom. Test Eur., Munich, Germany, Mar. 2008, pp. 236–241.