Reconfigurable and power-saving integrated operating system design for supporting mobile nodes in sensor networks

Chen-Shie Ho and Kuo-Cheng Chiang
Department of Computer Science and Information Engineering
VaNung University
Chung-Li, Taiwan, 320
hocs@vnu.edu.tw

Abstract

The power management system design is a key factor to determine the stability of a sensor nodes-based network environment. In the meanwhile, an application-based resource reconfigurable operating system can enhance the availability, maintainability and flexibility to the required services. In this paper, we propose an integrated power-saving mechanism and reconfigurable approach for the OS design consideration and demonstrate the critical application in farm monitoring and ubiquitous healthcare. To support fast mobile OS porting and implementation, we also propose the on-line retrieval-on-demand design concepts to carry out highly flexible system architecture. Its corresponding applicability in sensor network environment is discussed. The system design cost is evaluated and several simulations for different parameters are conducted and verified.

1. Introduction

Wireless sensor network (WSN) is a convergence of technologies including information processing, data communication and environment interaction. It is also an emergence of paradigms covering networks of autonomous systems and plays as a role of bridges between cyberspace and physical world [1]. The cluster of sensor nodes in these kinds of networks is equipped with highly sensitive micro-sensors which are in charge of gathering the changeable status of environment. The data aggregated in a node will be processed and analyzed by the microprocessor-based module to produce useful information corresponding to a specified service request. This information will be exchanged with other nodes and sent back to the sink points (or base stations) for further processing or responds to the service manager.

In a sensor node, to provide guarantee of high reliability and stability to the customer, a power management system will supply all available battery power to each component in a node. This makes monitoring the power consumption in order to adjust the operating modes dynamically become required. In general, to reduce the power consumption and save more energy resource, periodic data collection and network management associated with a sleep/wakeup cycle scheme is adopted. That is, all data acquisition, information processing and message communication procedures are done in a short wakeup cycle and the sensor node is in sleep mode almost all the time. To avoid loss of critical information delivery, event triggered model can be designed to handle the infrequent occurrence of events, and limited energy can be further saved when enforcing the node operates in this mode. Although the battery technologies make rapid progress in recent years, an efficient power management system is still a key component in sensor node system or architecture design.

On the other hand, to perform appropriate management of the hardware system in the sensor node, a small and compact operating system kernel which has the responsibility to perform resource and information management is designed and embedded to the node. For example, multi-level scheduling mechanism is implemented to arrange the executions of threads or code segments in preempted manner. Other tasks of this sort of OS include I/O controlling for large sensor arrays, communication stack and command processing, and provide interfaces between the hardware components and various applications activated by the remote users. To implement and develop the embedded OS for a specified hardware platform quickly, different level of abstraction and modularized design philosophy are also used. All the

0-7695-2553-9/06 $20.00

OS functions presented above can be designed as components in this design scenario. Particularly, the power management can be integrated into the OS design phase which attempts to make efficient energy management in sensor network application.

Since energy resource is very valuable in sensor node operation, low power design will be taken into consideration in each design phase of the overall system [2][3][6]. In addition to the *horizontal* low-power optimization which means exploiting the minimal specification and communication models, the *vertical* low-power optimization scenario can also be used in each component of the node architecture. For example, low-power CPU core and related small sized flash memory interface will be adopted as the embedding units during hardware design cycle and, in the same time, the low-power design philosophy such as resource sharing is considered from instruction set architectural level to physical level. Since the embedded operating system (EOS) is aware of all activities will be performed and takes charge of the system resource management, some degree of self-adjusting of operating mode is implemented in the kernel code to keep all tasks running as smooth as possible. Especially, the latest energy status has to be exchanged between the kernel and the power management system (PMS) to switch the procedure code into the power-saving mode dynamically and the OS should try to extend the lifetime of the node as long as possible.

2. Remote recharging mechanisms

2.1 Active recharging system

Since the sensor nodes in a WSN are installed once and will be expected to execute their pre-assigned power-consuming tasks continuously, the energy will be exhausted and some of the nodes become dead status eventually. The recharging activity can be accomplished by manual operation directly. Some kind of *active recharging* (also called *self-recharging*) systems are developed during these years. By equipped the energy transformation system which can transform the energy such as light, heat or vibration gathered from the natural environment to DC power providing to the system, the sensor node can maintain its live status till several years. This method eliminates the possibility of human intervention and extends the life cycle of sensor node efficiently. But even using the active recharging mechanism in sensor node design, some problems are still unavoidable and need to be carefully examined. For example, the energy resource is not stable enough which makes the charging model behaves non-linear, and the relationship between dynamic charging and node operation power consumption is difficult to evaluate. The system throughput will then be not as reliable as initial expectation. In some condition, for instance, the long rainy weather season, the energy is difficult to retrieve from light or heat, which still brings these nodes the risk to stop their operation at any instance. In addition, the extra add-on energy transformer cost is another factor which has influence on the price when considering the building revenue. In fact, how to maintain almost all nodes in a region in live mode is most important when designing the power generation and management unit in such systems.

2.2 Passive recharging system

Recently, the wireless power transportation (WPT) system architecture has become attractive due to remote biomedical diagnosis system [4][5][6]. In these systems, some micro-sensors with different monitoring functions are embedded into under-skin of the patients. To maintain the correct operation of these sensors, the DC power outside the body is transformed to radio frequency or microwave signal first. The transformed signal will be sent by antenna in the source node, and then be received to filter and rectify circuit by antenna in the destination node inside the body. Finally the signal is transformed to the DC power to be available to these sensors for long term tracking examination and control.

The wireless power transportation system can also be applied to the sensor network environment. In manual recharging system, the AA battery will be replaced by human beings, and the battery size and capacity is not so critical in such power refilling scenario. In active or passive recharging system, the requirement of the battery is more stringent. It should be long-efficiency, small-sized and rechargeable. Note that the operation mode of energy recharging depends on the applications. In a typical WPT system, the signals from sending node are modulated to from 900MHz to 5.8GHz frequency band by RF module in signal source, and amplified to proper power level for sending by dipole antenna. The impedance match circuit is necessary to achieve optimal noise matching and power amplifying efficiency, and it can be designed depends on the specification of the power amplifier component. The filtering and rectifying circuits can be implemented by simple LC and wave detection diode, and in general be designed as array layout form. All these components can be manufactured in integrated circuit technology. The power can achieve 30dB in distance with from 50cm of the sending node. The battery can be recharged to 4V after 60 minutes under 50 cm separations in distance. It should be noted that since the transmitting antenna is selected as omni-directional, all nodes around the transmitting node can receive the energy of the electromagnetic wave. That is, the *parallel recharging* in dense-distributed sensor nodes area is possible.

3. WPT-based WSN

We illustrate the wireless sensor network based on the WPT technology in Fig. 1. We assume each node in the network are equipped with bidirectional power transmission capability, i.e., they can act as either sending node or receiving node. In *hierarchical* operation model, two layers of networks are formed and each network can operate in their default operating domain. The hierarchical model is superior on flexibility of network resource allocation and power management. The WSN can be deployed to Fully WPT or partial WPT types. That is, the optimal least cost power transmittable nodes allocation can be determined by solving the objective function subject to location of grouped sensor nodes.

The operation of a WPT-based WSN can be divided into 3 phases and described as follows:

<u>Deployment phase</u>: all the sensor nodes are placed in static or dynamic manner. In static deployment, each node is put at fixed location and topology can be obtained in priori. In dynamic deployment, the location of each node is determined by different distributed scenario, and the topology of WSN is variable every time the network is deployed. WSN with location-movable nodes is also classified into this type.

<u>Auto-configuration phase</u>: the node has to exchange status information itself to other neighbors after all of them are placed. The neighbor discovery has to be accomplished in this phase. Note that the configuration of WSN can be implemented by customized or other optimization constraints. The successful ratio of deployment can be evaluated, and auto-recovery procedure will be activated if the failure node is equipped with intelligent self-healing capability.

<u>Operation phase</u>: all nodes will operate in event-driven model until they received a destroy command. The power recharging process is also activated after any node issues the *lack of power* (LOP) message. Note that no response of a node implies it's in dead status currently, and only power-related messages will lead to remote recharging. The nodes can also move to the new locations by remote control in this phase.

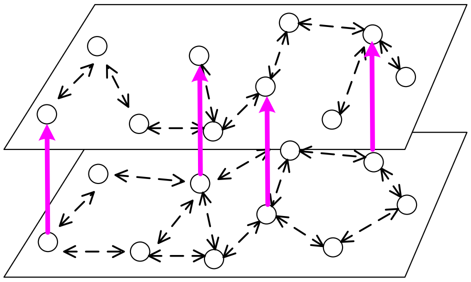
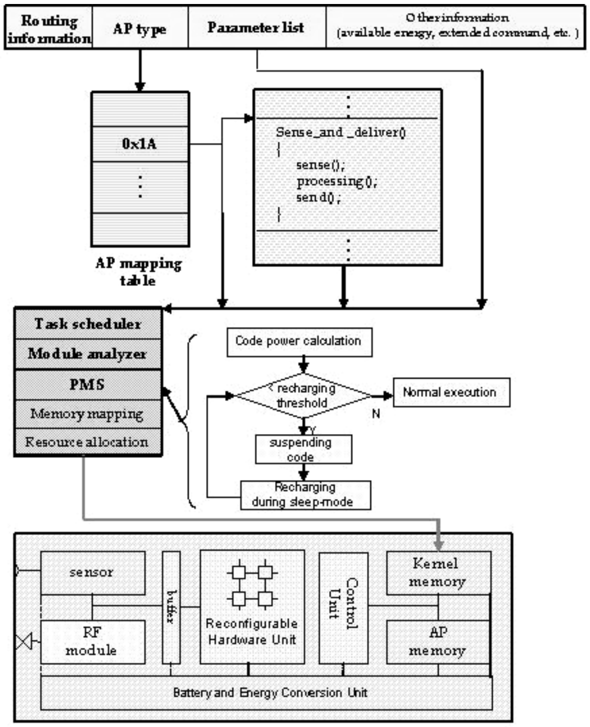


Fig. 1. Illustration of a hierarchical WPT-based WSN

4. Reconfigurable operating system design

The goal of a reconfigurable embedded OS includes providing constraint-based resource management and mapping. The constraints come from several metrics such as limited computation speed, storage space, real-time deadlines and message delivery cost. Since nodes in WSN are essentially autonomous, the control and management of WSN can be carried out by neighbor-based topology partition. All nodes in WSN will store the neighbor node table which records the adjacent nodes' IDs. The neighbor relationship will be determined in auto-configuration phase, and can be reconfigured during operation phase if some nodes are in stop mode, failure mode or deep-sleep mode. In this paper, we will consider the power budget limitation on resource allocation under local/group naming scheme. In group naming framework, the node ID will be replaced by group ID which is assigned after maximum broadcast range discovering procedure. The WSN is partitioned into several groups and each group has their own ID as message passing address. It's useful when only large-scale data aggregation application is necessary. To support most of other generic WSN operations, we still implemented local naming scheme in our simulations. Also note that in order to allocate sufficient resource for each running task efficiently, reconfigurable hardware platform is needed. Once a user request arrives at the sink node, the request is parsed and translated to standard command message, and then be sent to each group along the downstream direction to leaf nodes within each group. After received the command, OS in each node retrieves the command code from flash memory to the dynamic access memory and performs resource allocation which maps to reconfigurable hardware component by low-power constraints. The mapping table and the power-consumption table are recorded in each node in code module manner which is then called *module-based resource allocation* in our paper. It has the benefit when successes similar commands issued by different requests but translated to same modules. Only small portion of memory block has to be replaced in this situation and most of the codes are reserved in-place which then prevent further power consumption due to unnecessary memory bank transferring.



```
Sense_and _deliver()
{
    sense();
    processing();
    send();
}
```

Fig. 2. Illustration of operation flow for a generic user request

The events an OS will handle in our system include user requests and condition requests. The user request comes from the pre-defined system services and delivered by the sink node. The condition request is triggered by periodic sensing data when detecting abnormal monitoring value. Here we assume the command codes translated from user requests are stored in the flash memory and can be addressed by different AP-types. The commands are composed of several code modules which include several instructions mapping to one hardware components. This abstraction is beneficial to hardware-level low-power optimization.

We illustrate the detailed flow in Fig. 2. After receiving the command message from communication unit, the message is delivered through the protocol stack and the OS in sensor node finds the AP type field in the message. It will be loaded by OS kernel in the form of the mapped command code. The task manager in kernel implements the scheduler which is in charge of code module scheduling for queuing tasks. Before entering next stage, the power estimation step will be performed by comparing the latest battery available power and the AP code power consumption retrieved from the stored table. So the battery energy is examined when the scheduler or power checking routine is invoked. The result of comparison associated with the estimated power consumption value is forwarded to *operation mode processing routine* which will determine if the LOS message has to be issued by looking up the power-mode mapping table. We define the residue available power as 10 distinct levels, and classify them as normal, sleep, deep-sleep, LOS, and stop modes. The tasks operating in sleep and deep-sleep mode will be enforced to be replaced their code modules to low-power codes by OS. The tasks operating in LOS mode will be postponed to reactivate after the power refilling process is completed. The tasks operates in this mode can also be queued and not issued depends on the choice of OS designer. Note that the low-power code may be translated to low-power hardware mechanism as voltage scaling scheme which will slow down the operating speed [7][8][9]. The module analyzer then analyzes the incoming modules with the under-running models to check their dependency for further resource efficiency. Since the memory allocation policy will have great influence to the low-power operation performance, we isolate the memory mapping process as an independent kernel unit to calculate the memory requirements. After afford all related information, the resource allocation unit finally issues all control signals to drive all hardware components to accomplish the task execution. We ignore some detailed kernel units such as device drivers since all I/O port, control signals initialization and self-test process are completed during booting process in auto-configuration phase.

As stated before, the command codes are modularized by reconfigurable hardware units and defined by system designer. We list the command types as follows:

- Self_test() : the sensor node will execute self-test task and return the results including the power information to the sink node.
- Sense_and_Save() : the sensor node will execute sensing task and store the results in local memory for comparison in next time.
- Sense_and_Retrieve() : the sensor node will execute sensing task, compare the result with previous value or user defined parameter and return the results to the sink node.
- Sense_and_Group_Retrieve(): the sensor node will execute group sensing task, compare the result with previous value or user defined parameter and return the results to the sink node. Note that by implementing this operation, much of

communication bandwidth can be saved if the user only concern about some geographical local aggregated information.

To support on-line porting of OS kernels, we use fixed-location kernel module assignment scheme when allocating the OS codes. On-line porting is a very efficient memory-saving and highly flexible scheme when performing application to support resource-shortage device. It has benefit on low cost system establishment and ease to implementation of code mobility. In our system, we only consider the PMS module mobility, that is, PMS module is optional in some of the nodes in WSN. These nodes will issue the kernel module request to adjacent nodes, and install this module dynamically at any instance. Note that this operation is negligible if the WSN system is designed as no power critical applications. In such cases, the node which lacks of power will just stop their operation and leave itself there wait for advanced processing.

5. Simulation study

We conducted extensive simulation experiments to evaluate the performance of the proposed architecture. Since lacking of appropriate simulation tools for our design infrastructure, we developed a simulation environment written in Java multi-threads and focus on the module-based power consumption model. We assume all timing cost for each operation in components is constant and this information is stored in power consumption table. Initially 200 sensor nodes are random distributed in a 500meter × 250meter area in deployment phase, and all modules except the PMS are instantiated by member functions in node object. After auto-configuration phase all nodes store the neighbor relationship in a table by neighbor node ID and group ID. The neighbor discovery process is activated every 5 minutes by a simple handshaking protocol to reflect the latest network topology. Only one sink node exists and no isolated group is allowed, i.e., at least one wireless link have to build among different groups. The groups are formed by distance limitation by range of communication. The user request events are injected to the sink node following Poisson distribution and will be queued in sensor nodes if they received too many requests issued from the sink nodes simultaneously. The condition events are also injected in random distribution. The battery energy decays as the following function [3]:

$$E = [E' / (1+kI)] - Pt$$

where E' is the previous remaining energy and P is the power consumed in the time unit; k denotes the discharge rate dependence parameter and which determines how the value of the current affects the discharge rate. The radio model of transmitting power follows [2]:

$$E_{Tx}(k, d) = E_{elec} \times \eta + \varepsilon_{amp} \times \eta \times d^2$$

where η is the number of transmitted bits; d means the transmitting distance; E_{elec} is circuitry energy consumption of transceiver and ε_{amp} is transmitting amplifier coefficient. The typical values assumed here are $k = 0.5$, $E_{elec} = 50\text{nJ/bit}$ and $\varepsilon_{amp} = 100\ \text{pJ/bit/m}^2$. We will examine 4 parameters in the following:

- *Average delay due to OS intervention* D_{OS}
- *Average delay due to low-power code replacement* D_{CR}
- *Average bandwidth overhead due to power recharging* BW_{rech}
- *System design cost* C_{sys}

Fig. 3 shows the average delay time for a task in a sensor node after 10 runs of different number of injected events. The ratio of number of user events and condition events is 8:2, and the condition events occur in each group in random manner. Note that the OS related reconfiguration, process interruption and hardware resource shortage will cause delayed completion of a task. Here we only concern about the task execution time delay due to OS scheduling and power recharging. The low energy status of a node will cause slow execution of all its running tasks and power recharging will further make the delay longer. From the figure we note that the OS intervention duration is about 10% for a task in average in normal status, about 18% in low energy (sleep and deep-sleep) status and about 24% in LOS status. This reveals that one fourth time costs will pay in order to recharge for the battery to maintain the normal operation. Although the recharging technique can be modified to support performing recharging and operating simultaneously, we don't encourage this since this will increase the recharge time also makes the system operate unstably or prone to fail.

Since the OS will replace low-power code module when the battery energy becomes insufficient, we also measure the influence resulting from this action in Fig. 4. It will not increase extra memory swapping power consumption because it is replaced in place [10]. The figure shows that the number of code replacement will significantly increase the task execution time delay about 11% in 200 events case. The low power code in our implementation has about 8% power saving than non-optimized codes. As the low power design methodology can be carried out in hardware level, the simulation results point out the code replacement may be not a necessary mean to improve the energy utilization efficiency.

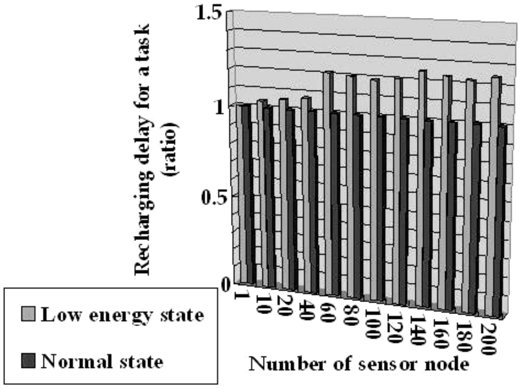


Fig. 3. Task recharging delay under different sensor nodes

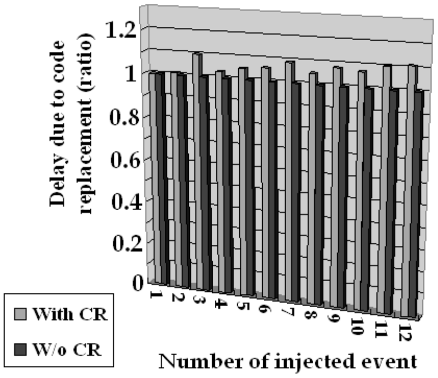


Fig. 4. Code replacement effect under different number of event

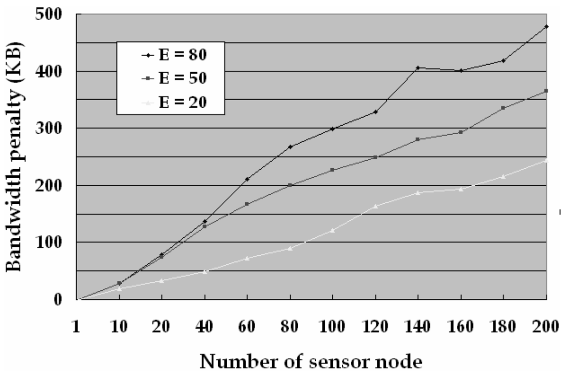


Fig. 5. Bandwidth overhead under different sensor nodes

The average bandwidth overhead due to power recharging is plotted in Fig. 5. The recharging radio can be modulated to 916MHz to prevent contention with normal operation by paying the cost for dual-frequency antenna. In our experiment we only assume 2.4GHz in-band communication, so all the tasks have to stop their operations under recharging phase. The extra BW overhead consists of recharging radio and control messages. It shows about 3% ~ 82% overhead of data transmission will occur in average in the figure. The worst case occurs at the situation that a task will not return any data upstream but be enforced to the recharging phase which then results in delayed task completion. Some scheduling optimization in group level will make improvement if appropriate protocol can be developed.

Finally, we estimate the system cost to setup the WPT-based WSN environment. Obviously the cost relates to the implementation mechanisms and adopted low power strategies. The total hardware cost including WPT components plus the $40 sensor node is about $100 with off-the-shelf devices, and the cost will keep down as integrated technology makes progress. The software modules are estimated as 26 modules and translated to about 2200 lines of targeted ARM 7 core assembly code. Since the total cost is proportional to the number of nodes in the network, trade-off between the system scale and service availability will be made under network planning phase.

6. Applications

The first application of our infrastructure is the farm and environment monitoring system. We assume the active recharging mechanism is implemented in such system because it is outdoors in native. Moreover, the light energy conversion equipment is the proper choice due to abundant sunlight energy resource. Because the position and covering range of sunlight will change as the sun moves, some WSN areas will be full of energy but some are not. The *power transfer* technique can be used to further improve the system lifetime and reduce extra power usage detection. The basic idea is the energy afforded from the nodes within the lighted area can be used as the energy for them but also used as the shared power resource for the nodes within other areas don't covered in sun lighted areas. The power storing scale will increase in such way and make the WSN keep in the live status as long as possible. Note that the WPT cost has to be included when performing the cost evaluation.

Another application of WPT-based WSN is on ubiquitous healthcare. Some smart body bio-parameter measurement equipments are invented these days and can be wore or installed on wrist, clothes and the body of patients. Some of them are battery powered and suffers the same energy shortage problem. In other outdoor situation, the sensor nodes used to monitor the disease distribution will proceed for a long time. The power transfer technique can also be used to this kind of application. Disease infection control in rural region needs stringent monitoring in dense population areas. The outdoor living behavior makes these people have more possibility to infect diseases from the environment. Since the monitoring characteristics in such areas are continuous and real-time, the seamless data sensing and aggregation is necessary for advanced investigation. By implementing the WPT mechanism to wireless sensor regions, the application programs resided in relay equipment can then maintain the running state longer than original battery lifetime and achieve the goal of anomaly detection.

7. Conclusion

WSN meets the strict challenge on power consumption which makes such network do not work well and miss expected functionality. How to reduce the power consumption or increase the system availability plays an important role when designing the architecture of this kind of network. In this paper, we propose an operation model for each sensor node in wireless power transportation technology based network architecture and present reconfiguring approach for the OS design consideration. From the simulation results, the group-based sensor node management is beneficial to low-power oriented system design. Our future research topics include development of efficient group partition scheme including considerations on each implementation layers, control protocol for power recharging and OS level optimization strategies to support customized user requests.

8. References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cyirci,"Wireless sensor networks: A survey", Computer Networks, vol. 38, no. 4, 2002, pp. 393-422.

[2] W. Heinzelman, A. Chandrakasan and H. Balakrishnan,"Energy-efficient communication protocol for wireless microsensor networks", Proc. 33rd Hawaii Int'l Conf. System Sciences, Jan., 2000.

[3] S. Park, A. Savvides, and M. B. Srivastava, "Battery capacity measurement and analysis using lithium coin cell battery", Proc. Int'l Symp. on low power electronics and design, ACM press, 2001, pp. 382-387.

[4] Y. H. Suh, K. Chang, "A high-efficiency dual-frequency rectenna for 2.45- and 5.8-GHz wireless power transmission", IEEE Trans. on Microwave Theory and Techniques, vol. 50, no. 7, July, 2002, pp. 1784-1789.

[5] T. C. Yo, C. H. Luo, "Implementation of biomedical microwave wireless charge system", Master thesis, Department of EE, National Cheng Kung University, Tai-Nang, Taiwan, June, 2003.

[6] T. Simunic, L. Benini, and G. DeMicheli, "Energy efficient design of portable wireless devices", International Symposium on Low Power Electronics and Design, 2000, pp. 49-54.

[7] A. Manzak, C. Chakrabarti, "Variable voltage task scheduling algorithms for minimizing energy/power", IEEE Trans. on VLSI systems, vol. 11, no. 2, April, 2003, pp. 270-276.

[8] A. Manzak, C. Chakrabarti, "A Low Power Scheduling Scheme With Resources Operating at Multiple Voltages", IEEE Trans. on VLSI systems, vol. 10, no. 1, February, 2002, pp. 6-14.

[9] G. Quan, X. Sharon Hu, "Minimal Energy Fixed-Priority Scheduling for Variable Voltage Processors", IEEE Trans. on CAD of IC and systems, vol. 22, no. 8, August, 2003, pp. 1062-1071.

[10] O. Avissar, R. Barua, D. Stewart, "An Optimal Memory Allocation Scheme for Scratch-Pad-Based Embedded Systems", ACM Transactions on Embedded Computing Systems, Vol. 1, No. 1, November, 2002, pp. 6–26.