Power-On Controller for high lifetime wireless sensor nodes

Mickaël Cartron, Nathaniel Seymour and Yannick Bonhomme

CEA, LIST, Laboratoire de Fiabilisation des Systèmes Embarqués, Boîte Courrier 94,
Gif-sur-Yvette, F-91191 France
mickael.cartron@cea.fr

Abstract. Power savings are nowadays crucial in embedded system contexts such as Wireless Sensor Networks (WSN) in order to increase the lifetime of sensor nodes. In this paper, we propose a new hardware structure called "Power-On Controller" (POC) for applying advanced control strategies for the "idle to active" node state transition. The proposed POC allows an optimization of power control by using event accumulation and spatial selectivity mechanisms. These new features allow to reduce the dynamic power consumption of roughly 60% compared to state-of-the-art power management solutions for a typical WSN applicative context, without altering the quality of service. The POC structure can be easily integrated in any sensor node based on system-on-chip design.

Introduction

Nowadays, distributed wireless devices are deployed more and more for multiple high value-added applications [19] (e.g. metering, monitoring, surveillance or tracking) in different contexts (e.g. industrial, military or medical). This success is directly linked to their low cost (i.e. for multi-purpose hardware platform, for maintenance and for network deployment and placement). Indeed, wireless nodes are self-powered with self-organized communication capability and as a consequence, they do not require any fixed infrastructure, neither for power, nor for communication.

Another specificity of wireless sensor networks is their applications which request low processing activity ("control-oriented applications") with no background tasks or intensive calculation.

Besides that, the lifetime of sensor networks nodes should be as high as possible under given design constraints (e.g. volume or weight) due to the environment of interest. This assertion leads to the fact that these systems are highly constrained in terms of energy consumption. In order to enhance the system autonomy, much progress has been obtained in battery technology (i.e. over 400% gain in 20 years [5]), in physical integration [25] and in dynamic and static power optimizations.

The work reported here was performed as part of the ongoing research program µSWN FP6-2005 IST-034642 and funded by the European Social Fund (ESF).

Please use the following format when citing this chapter:

Cartron, M., Seymour, N. and Bonhomme, Y., 2008, in IFIP International Federation for Information Processing, Volume 264; Wireless Sensor and Actor Networks II; Ali Miri; (Boston: Springer), pp. 49–61.

Some works aim at reducing dynamic and static power consumptions. The dynamic power consumption can be addressed by different kinds of techniques such as DVS (Dynamic Voltage Scaling) [6,7], AVS (Adaptive Voltage Scaling) [26], clock gating [20], asynchronous design [10] which has the particular advantage to reduce the peak power consumption. While current dynamic power optimization solutions are relevant for embedded system with high activity rates, they may be less relevant for sensor network nodes, due to their particular activity profile made of a lot of idle phases. Due to this special activity profile, solutions that address the static power consumption are also important for enhancing the system's lifetime. For that reason, static power reduction has been addressed by many works and more particularly in the sensor network context [21]. Among the techniques for reducing the static power, the use of MTCMOS (Multi-Threshold Complementary Metal Oxide Semi-conductor) technology [9] aims at reducing the leakage current with special transistors which act as power switches. Biasing solutions [22] are also proposed for saving static power.

Different works addressed the problem of power mode management. Some publications addressed the problem of the transition from active to idle state for micro-controller units [1,2]. Because this transition requires power consumption each time the system switches from an activity mode to another, it is necessary to change the power state only when necessary otherwise the gains observed by the use of a low-power mode might be counterbalanced by the losses due to transitions. Some works aim at parsing some information from the system and/or the software in order to take the best possible decisions for transiting from an active mode to a standby mode [1,2,3].

In this paper, we propose a specific hardware structure dedicated to the optimization of the transition from standby to active state. The most often used policy is to wake-up the whole system whenever a hardware event occurs. However, if this method allows the quickest possible reaction, it may be too much power consuming due to a multiplication of mode transitions, in case of unpredictable applications and/or bad transition choices from active to standby states. Actually, the "standby to active" transition requires some hardware and software applicative context knowledge. The proposed solution is a specific hardware component composed of a standard power management structure and an event driven filter in order to manage the "standby to active" transition.

The paper is organized as follows: in a first section, the architecture of interest is presented, so as the need for an adequate power control structure. A second section will be dedicated to the solution that is proposed for a new power control structure for sensor network applications. Then, the performance results are exposed and compared to a standard solution.

Related work - wireless sensor node architecture

Many sensor network platforms were proposed recently, some are commercial platforms and some are academic versions. Most of the mote platforms are based on general purpose low power micro-controllers (e.g. Texas Instruments MSP430 [8,15,13], Atmel ATMega128 [12] or Microchip PIC18xx [14]).

Then, some platforms appeared, based on a System-on-Chip (SoC) design. These architectures offer more flexibility in terms of multiple supply voltages and clock domains while simplifying the integration of analog parts. Some manufacturers already design SoC integrated platforms for sensor networks, such as Dust Networks SmartMesh-XD [16], Jennic JN513x [17], or Sensoria EnRoute500 [18].

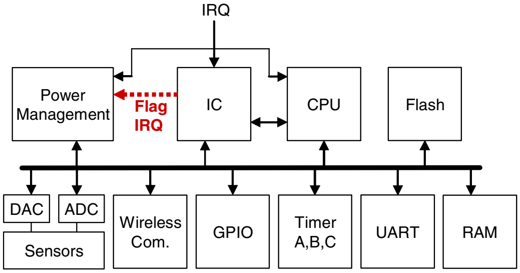

Fig. 1. Architecture of a wireless sensor node

The standard architecture considered in this paper belongs to the SoC family design and is illustrated on figure 1. In general, the digital part of the platform under study is composed of a processor core, an interrupt controller, a volatile memory block (RAM), a non volatile memory block (Flash), a General Purpose Input/Output (GPIO), several timers, a Universal Asynchronous Receiver Transmitter (UART), and an interconnect bus. The sensor part of the platform includes an Analog to Digital Converter (ADC), a Digital to Analog Converter (DAC), and the set of sensors. The communication part is constituted of a wireless communication component. Then, a power supply and clock supply block is used for powering components and driving the clocks of the platform.

The component dedicated to power management has different functionalities depending on the processor power state (i.e. “on” or “off”). During processor “on” state modes, the power manager component operates the Dynamic Power Management [24] (DPM) strategies selected by a software part by driving the supply voltage and the different clocks necessary for the system. We call this function the *“driving mechanism”*. During processor “off” state modes, the power manager component is in charge of the whole system wake-up, which is driven by hardware events (e.g. timer events or communication events) which play the role of the *“trigger mechanism”*. Most of the time, hardware events are interrupt requests (IRQ) and they are centralized by the interrupt controller which, in this case, plays the role of the trigger mechanism. Indeed, the power manager component incorporates a *“control mechanism”* that operates the reactivation of the whole system, ensuring that the wake-up procedures of components are respected.

For basic power managers, the trigger and control mechanisms are not optimal for the “standby to active” transition. In this basic approach, when a hardware event occurs, the power manager component strategy consists in activating the whole system. As it will be shown in this paper, this policy is not always optimal for decreasing the power consumption in case of low activity sensor network applications. Therefore, it is necessary to propose a solution compatible with either SoC or off-the-shelf designs of wireless platforms.

Proposed solution – Power-On Controller

The proposed solution is called Power-on-Controller (POC). The main idea of this work consists in using additional relevant information to the power management component. This information will be used for improving the flexibility of the reactivation mechanism. This section aims at describing the Power-On Controller (POC) constructed for adequately using this additional information. For describing the POC, first a description of the conception philosophy will be done. Then, a second part will describe how the POC structure has been implemented.

Inadequacy of the IC related information for power re-activation

The POC has been designed from the following observation. In previous state-of-the-art structures, IRQs are used for actuating the interrupt mechanism, which is the primary function, but they are also used as wake-up events in case the platform is in a low-power state, which is an extended function. To our opinion, the Interrupt Controller is no valuable structure for controlling the platform power management. Indeed, in platforms that use power control, the IRQs are above all dedicated and designed for interrupt processing purposes. So, a clearer identification of functions of interrupt processing and of power control is necessary in order to propose optimal hardware structure for power-on control, rather than using the structure initially dedicated to interrupt management.

So, on the one hand, the interrupt control aims at controlling the interrupt mechanism of the processor, which consists in stopping the normal execution of the processor in order to run a specific routine called the handler. On the other hand, the power-on structure aims at operating a total or partial platform reactivation consecutively to events observed on the platform. There are similarities between the two mechanisms, since they are both consecutive to the occurrence of an hardware event. For that reason, they are often based on the same information, which is presented in the last column of the table 1.

Table 1 presents the relevant information for applying adequate reaction for the interrupt control mechanism on the one hand and for the power-on mechanism on the other hand. It appears that the interrupt control related information is only a second best for being used as power-on information. In general, the interrupt masking is used as-is and the priority information is ignored and some very simple hardware is used instead, which is based on OR gates. This simple hardware allows a basic strategy of reactivating the whole platform immediately when any unmasked IRQ is detected.

The proposed POC component uses the most relevant information for processing the reactivation, which corresponds to the first column of table 1. As a consequence, the POC takes IRQ as inputs, so as the usual interrupt controller, as illustrated on figure 2. The specific information is used by the POC for applying advanced power management policies such as event accumulation and domain selection at reactivation while being easy to integrate in any SoC node platform. It should be noticed that the POC can been seen as an IC extension as well, but for simplifying the presentation, it appeared to be clearer to give a specific name to this extension. Another reason why the POC can be understood like a component clearly separated from the IC is that it is generic and it can be used within any platform. However, this adaptation would require some effort.

Table 1. Comparison of the relevant information for the interrupt mechanism and the power-on mechanism

	Power-on control	**Interrupt control**
Events	IRQ	
Masking Policy	Enabling/disabling the reactivation transition	Enabling/disabling the CPU interrupts
Priority Policy	Information of the urgency of the reaction (defines if an additional latency can be tolerated or not)	Information of the priority level of the HW event (helps defining which IRQ will get the CPU resource first)
Necessary HW	Information about the platform parts to be reactivated for a given IRQ	Unspecified

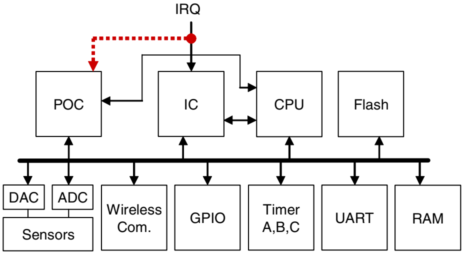

Fig. 2. POC integration in a standard wireless node architecture

Event accumulation principle:

Every time a reactivation occurs, it is associated to an energetic cost. For that reason, switching too often from a power mode to another is highly energy consuming. The additional knowledge of the urgency level of events, which is proposed with the POC, allows to apply a delayed wake-up in case the routine associated with the IRQ can tolerate any latency. This leads to event accumulation which is the fact of buffering the events which can be processed with additional delay without altering the quality of service. When an event is considered as non urgent, it is stored until an urgent one occurs, or until an internal time-out expires. Then, the system is powered-on and all pending IRQs are processed by the CPU. The energy related to the standby to active transition is consumed only once. When an urgent event occurs, it immediately triggers the reactivation transition.

Of course, event accumulation adds delay for processing non urgent events, but when events are labelled as urgent, there is no additional latency. In that sense, the quality of service remains optimal for urgent events. The event urgency labelling is done by low-level software.

Domain selection at reactivation

Another additional knowledge handled by the POC is the information about the platform parts to be reactivated for a given IRQ, as indicated in table 1. This information is used by the control part for triggering an adequate reaction when some hardware events occur. The goal is to select in advance the power supply and clock domains that will have to be reactivated when an actual reactivation event occurs.

This allows a finer control than a total reactivation, which is the only possibility on any solutions based on interrupt control information only. By doing this way, the parts that are reactivated for the event processing are in accordance with a specific event. More details about this mechanism will be given in the results section.

Flexible integration

Since the POC component does not rely on interrupt related information such as interrupt masking or priority, the component can be integrated in any standard architecture with a processor core associated to an interrupt controller (as is the case for almost all sensor network platforms).

However, it should be noticed that the event buffering capability of the interrupt controller might need to be enhanced. The enhancement could focus on the increase of event buffers sizes or on the implementation of a “buffer full” flag from the interrupt controller to the POC which would give the order to reactivate the platform.

Implementation of the POC at system level

In this section, the integration of the POC at system level is described in order to enhance the power control for peripherals and for the processor core.

Power control of peripherals

The power management of peripherals is done as follows and is summed up in table 2. During processor “off” states, the POC takes full control on the peripherals’ power modes. During processor “on” states, the power management of peripherals is managed by low level software (which can be called DPM, for Dynamic Power Management) running on the processor and the POC only applies the orders of the DPM software part.

Table 2. Mechanisms involved for the power control of peripherals and processor core

	Trigger	Control	Driving
Processor “On” state	SW	SW	POC
Processor “Off” state	POC	POC	

Power control of the processor

For the processor case, the driving mechanism is operated by the POC. The control mechanism for power-on feature is done by a special FSM (Finite State Machine) that runs in the POC. This control task cannot be ensured by the processor itself because the processor may be unavailable (when “off”). Then, the trigger mechanism is ensured either by the interrupt controller or the POC. This organization is summed up on table 2.

Implementation of the POC at component level

The system view of the POC has been presented in the previous section. The current section presents the hardware structures that will apply the new policies of event accumulation and spatial selectivity for wake-up (which have been previously presented).

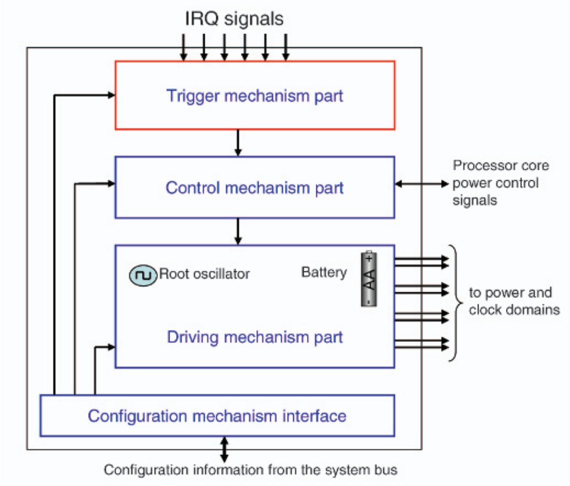
IRQ signals
Trigger mechanism part
Control mechanism part
Processor core power control signals
Root oscillator
Battery
+ AA -
Driving mechanism part
to power and clock domains
Configuration mechanism interface
Configuration information from the system bus

Fig. 3. Hardware structure representation of the POC

The schematic of the POC is presented on figure 3. The subcomponents are as follows.

- The first subcomponent is the driving mechanism part, which consists in the PWM (Pulse Width Modulation) signal for the supply voltage generation by the adequate analog structures such as “Buck” DC/DC converters (Direct Current/Direct Current) [11]. The generation of the different clock outputs is realized with a root PLL-based oscillator structure. Then clocks and supply voltages are routed to the whole platform.
- The second subcomponent of the POC is the trigger mechanism part. First, this part determines whether a platform reactivation is necessary or not, with or without any tolerance on the reactivation delay. Second, the trigger part defines the partial system reactivation policies to use according to the received events. The inputs of this

block are the IRQ lines driven by the peripherals. In our implementation, these IRQ lines are the same as those used for standard interrupt control.

- Third, the *configuration mechanism interface* is connected to the system bus for the configuration phases. This configuration interface is located between the component and the processor. In our implementation, the configuration interface is Wishbone [4] compliant. This interface is accessed with dedicated low-level software for adequately configuring the system.
- The last part of the POC is the *control part* which manages the power mode of the processor core and peripherals. The module controls the driving part of the peripherals and of the processor core according to the external events, to the processor power control signals and to the configuration interface. It has been implemented using a Finite State Machine (FSM).

Experimental results

For evaluating the structures proposed by the POC, a realistic applicative context has been considered. According to this scenario, two approaches are compared. The first approach is a prior state-of-the-art solution such as what can be found in most of commercial or academic sensor network platforms. The results exposed here are compared to a platform that would use the Texas Instrument MSP430 microcontroller wake-up structure. It operates an immediate and total reactivation of the chip each time an unmasked event is detected. The second approach uses the POC enhancements for lowering power consumption.

The chosen scenario takes place in a sanatorium. Some patients are equipped with a bracelet which monitors their heart rate activity (this application is called “Heart monitoring”) and sends alerts when a problem occurs and allows them to call for some help when needed (“Help button”). Also, the bracelets are supposed to send global running information about embedded sensors at regular intervals of 10 minutes (“Execution report”). Beside all this, a battery test is done hourly for ensuring that the bracelet never runs out of power at a bad time (“Battery test”). Table 3 shows four applications running on a given sensor node of the network and specifies the frequency of hardware events and the urgency level of this event. Table 3 also shows which hardware component is in charge of generating the event.

Table 3. Mechanisms involved for the power control of peripherals and processor for the sanatorium scenario

Application	Triggering Hardware	Event frequency	Event urgency level
Heart monitoring	Sensor part	Punctual (rare)	Urgent
Execution report	Timer	1 event / 10 min	Non urgent
Help button	GPIO	Punctual (once a day)	Urgent
Battery test	Timer	1 event / 1 hour	Non urgent

Results for event accumulation

Figure 4a shows the action of event accumulation. One can see that the number of events processed at the occurrence of a reactivation phase is superior or equal to one, while this number is one for a classic method. The curve is floored at 1 for lower values of the accumulation period that defines the maximum waiting period in case that no urgent event occurs for emptying the event queue. When the accumulation timer period is set above a particular value which depends on the application, the average number of events accumulated becomes strictly superior to one. For our application this number grows linearly for time interval configurations superior to 5 minutes.

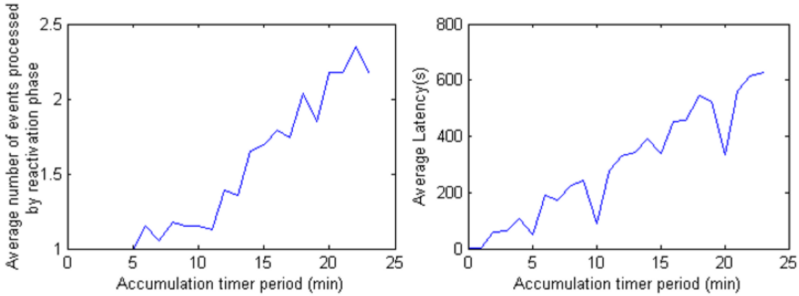

Fig. 4a

Fig. 4b

Fig. 4a and 4b. Average number of events processed for one reactivation phase and average latency

The average delay between the reception time of an event and its processing time is shown on figure 4b. Logically, the average delay for processing events grows approximately linearly with the accumulation timer period. The irregularities that appear on the curve are due to the applicative scenario where events are generated periodically. When the applicative process meets particularly well the accumulation timer it has for consequence to reduce the average delays. From this observation, we see that a good knowledge of the application can lead to keeping the average latency at reasonable levels while significantly reducing power consumption, and without impacting the delay at all for urgent signals since the average delay for processing urgent events remains approximately the same for both solutions.

Figure 5 and table 4 present the average power consumption due to events processing for the basic solution on the one hand and with the use of the POC component on the other hand. Here, significant dynamic power savings can be highlighted up to 58% for 20 minutes of timer period accumulation. To our opinion, these results show that the gains compensate the POC's consumption overhead.

Table 4. Dynamic power reduction results

Accumulation timer period (min)	5	10	15	20
Dynamic power reduction (%)	1.96	17.65	44.12	58.82

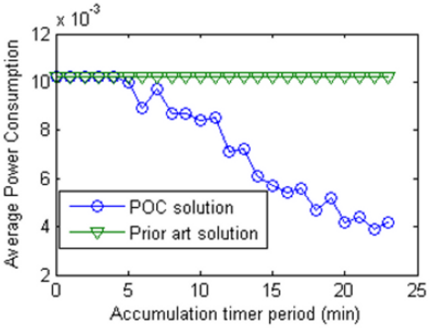

Fig. 5. Average power consumption due events handling

Results for spatial selectivity

In this second typical use case, the platform considered has distinct physical domains (i.e. clock and power supply domains) that can be handled separately. The different components of the platforms are assigned to physical domains as follows.

- **Domain – Power and power control.** It includes the POC component and the batteries. This domain is always active and allows the operation of the other domains.
- **Domain – Monitoring.** The components in this domain are able to initiate events that lead to reactivation. In order to enhance the control granularity this domains is split into 3 sub-domains:
- – Sub-domain – Timer Monitoring
- – Sub-domain – ADC/Sensor Monitoring
- – Sub-domain – Wireless Communication Monitoring
- **Domain – Processing.** This domain includes the minimum blocks needed for processing ; that is the processor core, the interrupt controller, the volatile memory, and the interconnect bus.
- **Domain – Internal Communication.** It includes blocks such as a Universal Synchronous and Asynchronous Receiver Transceiver module (USART), a Serial Peripheral Interface controller (SPI) or an Inter Integrated Circuit bus (I^2C). This domain is activated for allowing a communication with additional peripherals.
- **Domain – External Communication.** This domain includes the modules used for communicating with other network entities. In our applicative context, it is composed of the wireless communication hardware.
- **Domain – Storage.** This domain is activated when access to storage structures is needed. Typically, non volatile memory elements such as flash memory elements are parts of this domain.

We define a mode as a combination of domain states. We consider seven modes using the domains defined above:

- "**Timer monitoring**" – One or several timers are activated for monitoring purpose
- "**Sensor Monitoring**" – ADC/Sensor is activated for monitoring purpose

- "**Wireless Monitoring**" – A part of the wireless communication is activated for monitoring purpose.
- "**Recording**" – Storing information from the processor core to the non volatile memory.
- "**Active Processing**" – The processor core is active, while no peripheral is active.
- "**Active Se**" – The sensor and processing domains are active.
- "**Active Wi**" – Wireless communication is active.
- "**Active Wi + Se**" – Wireless communication and Sensor domains are active

Table 5. Platform modes and domains

	Power and Power Control	Timer Monitoring	ADC / Sensor Monitoring	Wireless Communication Monitoring	Processing	Internal Com.	External Com.	Storage
Timer monitoring	on	on	off	off	off	off	off	off
Sensor Monitoring	on	off	on	off	off	off	off	off
Wireless Monitoring	on	off	off	on	off	off	off	off
Recording	on	off	off	off	on	off	off	on
Active Processing	on	off	off	off	on	off	off	off
Active Se	on	off	on	off	on	on	off	off
Active Wi	on	off	off	on	on	on	on	off
Active Wi+Se	on	off	on	on	on	on	on	off

Table 5 presents the organization of the platform modes according to the structural domains. This organization is compatible with SoC based approaches. In the applicative context defined above, different monitoring modes can be activated simultaneously in order to trigger events from the ADC/sensor, wireless communication, timer monitoring or GPIO. In previous solutions, when an event occurs all the systems is activated. With the POC, partial activation strategies can be defined. For example when a wireless event is raised; the POC can select the “Active Wi” mode in order to process the communication information corresponding to the event.

The overhead of the structure varies as a function of two parameters: the number of domains (*Ndom*) and the number of IRQ (*Nirq*). The impact of these parameters on the POC area has been evaluated for each POC sub component. The area of the control part is approximately constant with a variation of *Nirq* or *Ndom*. The area of the driving part is approximately proportional to *Ndom* and does not depend on *Nirq*. The trigger part divides into two parts. One grows proportionally to *Nirq*, and does not depend on *Ndom*. The other part of the trigger is proportional to the product of *Ndom* and *Nirq*. Then, the configuration component is constituted of: one constant area part, one that grows proportionally to the product of *Nirq* and *Ndom*, one that grows proportionally to *Nirq*, and a last one that grows proportionally to *Ndom*.

To sum up, the area occupation of the POC grows linearly with the two main parameters *Nirq* and *Ndom*. This area overhead has been evaluated for an instance of the POC with *Nirq=16* and *Ndom=4* and the corresponding results are presented on table 6. The evaluation was performed with Synopsys Design Compiler tools [27] with a 90 nm CMOS standard cells library technology.

Table 6. Overhead due to the POC for 16 IRQ and 4 clock/voltage domains

	Trigger	Control	Driving	Configuration	Total
Number of equivalent gates	477	732	268	312	1789
Silicon Area (µm²)	2860	4393	1607	1869	10729

Conclusion

The proposed component addresses the problem of the lifetime of autonomous nodes in WSN applicative context. WSN applications are generally less restrictive in terms of quality of service and present a low activity rate. The Power-On Controller (POC) takes advantage of these properties in order to save dynamic power consumption. The proposed solution interacts with IRQs and system core components for applying adequate event accumulation and spatial selection mechanisms.

The event accumulation mechanism proposes to limit the energy cost due to the system on chip reactivations. Indeed, system reactivation is an important power-consuming source. Event accumulation mechanism succeeds to save dynamic power consumption up to 58% on the proposed WSN application. Moreover, the POC proposes a spatial selectivity mechanism in order to wake-up hardware parts that are relevant for a given IRQ. Indeed, previous state-of-the-art solutions generally wake-up the whole system without considering the source of IRQ. The POC integration has an area overhead of 10729 µm² for the 90nm CMOS technology and its gate usage is 1789 gates.

It has been focused in section IV that the periodicity that may be observed on typical sensor network applicative context, such as the one described in table 3, could be exploited to properly configure the accumulation period of the POC in order to keep the average latency at reasonable levels even for non urgent events. Future work will be realized on low-level software for properly tuning the accumulation period parameter with an adaptive methodology and/or advanced applicative knowledge.

Reference

[1] Simunic, T.; Benini, L.; Glynn, P.; De Micheli, G., “Event-driven power management,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* , vol.20, no.7, pp.840-857, Jul 2001

[2] Chi-Hong Hwang; Wu, A.C.-H., "A predictive system shutdown method for energy saving of event-driven computation," *Computer-Aided Design, 1997. Digest of Technical Papers., 1997 IEEE/ACM International Conference on* , vol., no., pp.28-32, 9-13 Nov 1997

[3] Bellosa, F., “The Benefits of Event-Driven Energy Accounting in Power-Sensitive Systems”, *Proceeding of the 9th workshop on ACM SIGOPS European workshop*, pages 37-42, 2000.

[4] Wishbone System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores Revision B.1, Silicore Corporation, 2001.

[5] Roundy, S.; Steingart, D.; Frechette, L.; Wright, P. and Rabaey, J., “Power Sources for Wireless Sensor Networks”. *In Wireless Sensor Networks*. Springer Berlin / Deidelberg, vol. 2920, pages 1-17, 2004.

[6] Chandrakasan, A.P.; Sheng, S.; Brodersen, R.W., "Low-power CMOS digital design," *Solid-State Circuits, IEEE Journal of*, vol.27, no.4, pp.473-484, Apr 1992

[7] Chandrakasan, A.; Min, R.; Bhardwaj, M.; Cho, S.; Wang, A., "Power aware wireless microsensor systems," *Solid-State Circuits Conference, 2002. ESSCIRC 2002. Proceedings of the 28th European*, vol., no., pp. 47-54, 24-26 Sept. 2002

[8] Texas Instruments, “MSP430x1xx Family User’s Guide, Rev. F”, 2006.

[9] Calhoun, B.; Honore, F.; Chandrakasan, A., “Design Methodology for Fine-Grained Leakage Control in MTCMOS”, *2003 International Symposium on Low Power Electronics and Design, Proceedings of the*, ACM, pages 104-109, 2003.

[10] Ammar, Y.; Buhring, A.; Marzencki, M.; Charlot, B.; Basrour, S.; Matou, K.; Renaudin, M., “Wireless Sensor Network node with asynchronous architecture and vibration harvesting micro power generator”, *2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies , Proceedings of the,* pages 287-292 , 2005.

[11] Mohan, Undeland, Robbins, "Power Electronics: Converters, Applications and Design", Wiley, 1989.

[12] Crossbow. http://www.xbow.com/

[13] Moteiv. http://www.moteiv.com/

[14] Convergix. http://www.convergent-electronics.com

[15] Dr.-Ing. habil. Jochen H. Schiller, Dr.-Ing. Achim Liers, Dr.-Ing. Harmut Ritter. Scatterweb at Free University of Berlin. Institute of Computer Science, Freie Universität Berlin, Germany. Website: http://www.inf.fu-berlin.de/inst/ag-tech/scatterweb_net/index.html

[16] Dust Networks. http://www.dustnetworks.com

[17] Jennic. http://www.jennic.com

[18] Sensoria. http://www.sensoria.com/

[19] D. Estrin et al. “Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers” National Research Council Report, 2001.

[20] Raghunathan, A.; Dey, S.; Jha, N.K., “Register transfer level power optimization with emphasis on glitch analysis and reduction” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol.18, no.8, pp.1114-1131, Aug 1999

[21] Hempstead, M.; Wei, G.-Y.; Brooks, D., “Architecture and Circuit Techniques for Low-Throughput, Energy-Constrained Systems across Technology Generations” *Compilers, Architecture, and Synthesis for Embedded Systems (CASES), IEEE/ACM International Conference on*, October 2006

[22] Kesharvarzi, A.; Narenda, S. et. al., “Technology scaling behavior of optimum reverse body bias for leakage power reduction in Ics,” *Low Power Electronics and Design, Intl. Symp. on*, pp. 252-254, 1999.

[23] Microchip Technology Inc., “Pic18f2585/4585/2680/4680 data sheet.” [Online]. Available: http://ww1.microchip.com/downloads/en/DeviceDoc/39625b.pdf

[24] Benini, L.; Bogliolo, A.; De Micheli, G., “A Survey of Design Techniques for System-Level Dynamic Power Management” *IEEE Trans. on VLSI Systems*, Feb.2000. pp.299-316.

[25] Zeitzoff, P.M.; Chung, J.E., “A perspective from the 2003 ITRS: MOSFET scaling trends, challenges, and potential solutions” *Circuits and Devices Magazine*, IEEE, vol.21, no.1, pp. 4-15, Jan.-Feb. 2005.

[26] Elgebaly, M.; Sachdev, M.,“Variation-Aware Adaptive Voltage Scaling System” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on* , vol.15, no.5, pp.560-571, May 2007.

[27] Design Compiler Reference Manual, Synopsys, V.11, 2000.