Power gain estimation of an event-driven wake-up controller dedicated to WSN's microcontroller

Florent Berthier, Edith Beigne, Pascal Vivet, Olivier Sentieys

Power Gain Estimation of an Event-driven Wake-Up Controller dedicated to WSN's Microcontroller

Florent BERTHIER[1], Edith BEIGNE[1], Pascal VIVET[1], Olivier SENTIEYS[2]
[1]CEA LETI, MINATEC, 38054, Grenoble, FRANCE
[2]IRISA/INRIA, University of Rennes 1, 6 rue Kerampont BP80518 - 22305 Lannion Cedex, FRANCE
Email: [1]{florent.berthier, edith.beigne, pascal.vivet}@cea.fr, [2]olivier.sentieys@irisa.fr

***Abstract*—To deal with Wireless Sensor Node's energy constraints, new architectural solutions have to be found. This paper proposes to analyze a WSN microcontroller sub-system power consumption to extract the main power contributors according to different applicative execution phases. The objective is to come out with the energy reduction potentiality offered by an additional module called Wake-Up Controller. This block is able to substitute to the main CPU for current tasks like data transfers between sensors, memories or radio and fine grain power/frequency management of the entire node's sub-modules. Power simulations of a microcontroller sub-system based on FDSOI28 technology, with and without the Wake-Up Controller use, are proposed. Results are presented for applicative scenarios ranging from very low to high activity rates. This study exhibits power gains from 14.5% to 76% in the full range attesting the future design of this new module.**

I. Introduction

The Internet of Things is expected to comprise billions of connected devices, many of which will be wireless sensor nodes (WSN) communicating through a network. The nodes are spatially distributed and able to measure physical or environmental conditions while transferring data through a wireless link. Each node must be able to manage sensors measurements, data fusion, data transfers to other nodes and handle network protocol. One challenge this poses is energy efficiency of those autonomous nodes, as it will be cost-prohibitive or even impossible to regularly replace their batteries. The concern is even greater if WSN are recovering their energy from the environment as they should then tackle with changing energy levels. To mitigate energy issues and reduce the cost, future WSNs will be highly integrated comprising sub-system microcontroller, non-volatile memories, analog sensors and mixed-signal interfaces. We propose in this paper a complete power analysis of actual Wireless Sensor Nodes microcontroller sub-system in order to extract the major power contributors. Figures and models are extracted from FDSOI28 technology [1] and components datasheets. Our objective is to clearly show, by simulations, that contrary to conventional already published approaches [2] [3], we can reduce the node's energy by partitioning the sub-system microcontroller into two parts: a conventional CPU computing unit and a very low energy Wake-Up event Controller (WUC) activated only by several Wake-Up sources and in charge of simple data transfers and sub-modules power management. In [2], in spite of it's asynchronous design providing a wide voltage range and a unique low power idle state, the specific architecture and instruction set architecture can make the programming difficult contrary to classical market microcontrollers. SleepWalker [3] presents a 16-bit core based on MSP430 instruction set achieving a very low power consumption in standby mode (1,69 μW). However, no wake up sources are specified in this mode limiting deep sleep mode usage and the processor is always woken up when an event raises. We can also find in the literature WSN circuits demonstrating low standby, retention and active power. Among them, [4] demonstrates 11 nW standby power and 40 μW active power for the whole node in which an efficient Power Management Unit (PMU) controls the power delivery. In all the previously cited references, it appears that the main CPU is still in charge of simple tasks whereas it could be powered down when no computing tasks are requested. In [5], a partitioned approach is proposed but the "event processor" is coupled with an 8-bit main processor, and can manage only 9 power domains without selecting supply voltage/frequency. Moreover the Instruction Set Architecture (ISA) is very limited and cannot handle any type of standard radio or sensor interface for data transfer. The main contribution of this paper is to estimate the energy gains of our partitioned approach on applicative scenarios. We will thus present in section II scenarios example with different application phases used for power simulation of the system with and without Wake Up Controller which is specified in section III. Results in each application phases (section IV) and for complete applicative WSN scenario (section V) are finally exposed.

II. Conventional WSN's Applicative Phases

To analyze the energy benefits of our partitioned approach, we consider classical sensor node architecture [6] composed of four parts: (i) Radio module for packets reception and transmission; (ii) Sensors for physical measurements; (iii) sub-system microcontroller; (iv) Battery/Power Management Unit/Energy Harvesting System for energy storing and scavenging. The sub-system microcontroller is not only in charge of sensor data fusion and processing but also of network communication protocol handling and sub-modules power gating. It can, for example, power gate the radio and sensors to save power during long standby period and reconfigure them before further use. We can depict a conventional execution flow as in Figure 1 [6].This flow will be used all along the

978-1-4799-8893-8/15/$31.00

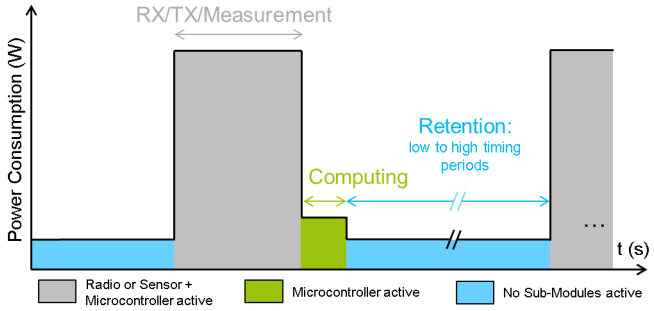


Fig. 1. Phase decomposition of an applicative WSN execution flow

paper to study the standby and active power contributions of each of the sub-modules running into different applicative configurations. Figure 1 shows a 3-phases decomposition of a WSN applicative flow valid for (very) low, mid and high duty-cycles executions. During Retention phase, main sub-modules clocks are gated and low power supply is maintained for data retention. RX/TX/Measurement phase consists in running sensors or RX/TX radio modules to perform data wireless transfer or data measurements. This phase duration is relative to the conversion accuracy or to the radio protocol required by the application. During the Computing phase, each microcontroller's sub-modules are powered on and run at their maximum voltage and frequency values. Those different phases will be considered to evaluate microcontroller's power consumption at different applicative scenarios. We focus on the microcontroller sub-system tasks that are decomposed into: data computation, data transfer and power management. It appears that, in addition to basic power management tasks, the microcontroller main task is to carry out sensors measurements from peripherals and send/receive converted data. Consequently, we propose to add a Wake-Up Controller, always on, dedicated only to data measurement/transfer tasks and fine grain power management. As this is not obvious that this additional hardware can reduce the overall energy of the node, we propose, in this paper, to give a complete analysis of main power contributors for different applicative scenarios ranging from very low to high activity rates. For each scenario, we analyzed the power consumption of both solutions: conventional CPU sub-system approach based on ARM M0+ microcontroller and partitioned approach with an additional Wake-Up Controller.

III. Wake Up Controller Specifications

The aim of this section is to detail the Wake-Up Controller main tasks and specifications. For power simulations scenarios proposed in the paper, the tasks described below will be either handled *only* by the main CPU (conventional microcontroller scheme) or shared between WUC and CPU. We chose for the simulations, ARM Cortex M0+ processor as the main CPU. Event processors architecture like [5] are well adapted to our WUC specifications that are (i) Manage all Power/Frequency modes of each sub-modules; (ii) Manage interruptions when M0+ is power gated; (iii) Execute Interrupt Service Routines

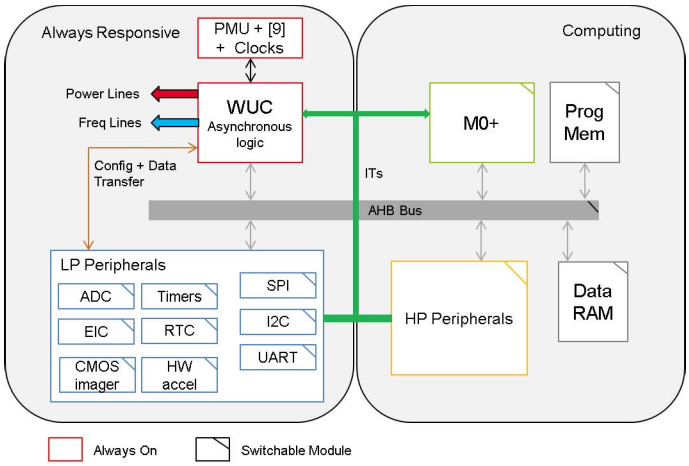


Fig. 2. Sub-system microcontroller: partitioning between Always Responsive Module and Computing Module

associated to each interrupt vector; (iv) Manage data transfer tasks; (v) Reconfigure peripherals, radio or sensor that have been power gated during long standby period; (vi) Return control to the main CPU for specific computing tasks. The wake-up controller tasks can be performed while most of the node is in standby except some accelerators and memories. This implies some specific hardware like timers and an interface in a Real Time Operating System (RTOS). Figure 2 shows the architecture of the system partitioned into an Always Responsive part and a Computing part. The Always Responsive part is composed of an always on WUC, PMU, Clock generators, and Low Power Peripherals. The Computing part is composed of the main processor Cortex ARM M0+ and High Power Peripherals. According to wake-up interrupts or events arrival, the WUC can control transfers from sensor to internal data memory. If data communication is needed, the WUC can read or write into the radio module and reconfigure it to reduce power. Finally, as shown on figure 2, WUC can control any transfer related to the entire address space of the sub-system microcontroller. All modules unused are power gated and turning on when needed.

IV. Microcontroller Simulation In Each Applicative Phase

This section first reports the microcontroller power simulations, for each applicative phases (Figure 1) to attest the energy gains obtained by adding a Wake-Up Controller. Those simulations will be extended to a full scenario in Section V. We build a simulation model (Figure 3) including all the necessary sub-modules for high level applicative power estimation. This model is based on power figures from FDSOI28 technology [1], dedicated ultra low power microcontrollers and peripherals found in datasheets [7] [8]. Those power figures are estimated but we do not pretend to give precise absolute values, we aim at estimating relative power gains with and without the proposed WUC. For clarity of purpose and before entering in the detailed simulation results, it is necessary to explain the different modes that can be used for power reduction.

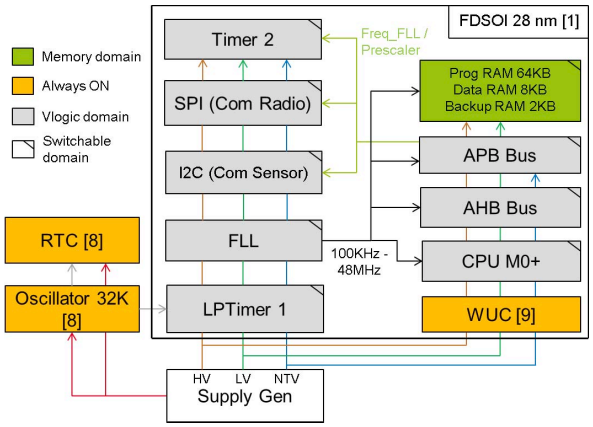
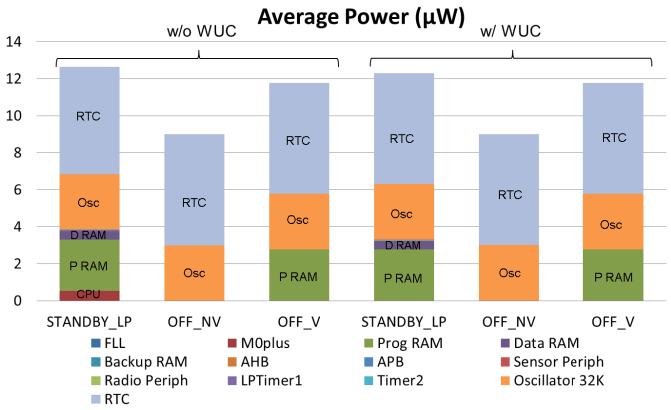
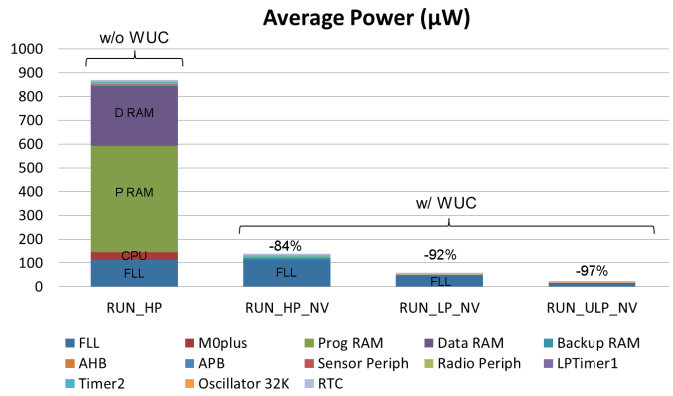


Fig. 3. Sub-modules Microcontroller Power Model for simulation (HV: High Voltage, LV: Low Voltage, NTV: Near Threshold Voltage)

Table I shows different microcontroller's power modes handled conventionally by a Power Manager. Four power modes are defined (RUN, IDLE, STANDBY, OFF) leading to different configurations of logic domain clocks, logic domain voltage supply, memory domain voltage supply and wake up sources. Figure 4a reports power simulation results performed with Aceplorer (Docea Power) for an applicative Retention phase with and without our proposed WUC. During this phase, STANDBY and OFF modes are usable and reported on X axis. This simulation also takes into account the Non Volatile (_NV) memory option for program/data retention. If this option is available, the retention phase will consume less leakage power compared to the use of a standard Volatile SRAM (_V) for which the power supply is maintained at a minimum voltage. In the latter case, we propose to use a small backup memory (2KB) used to store critical data and context application. In STANDBY mode, the main power difference stands in the fact that the CPU can be fully power gated when a WUC is available. OFF modes simulations results exhibit almost the same power consumption with or w/o WUC what is expected as all the modules are switched off. However, as the WUC is event-driven, it can be activated on any wake-up sources at any moment without any latency. In Figure 4b, average power consumption for RX/TX/Measurement phase is presented. In this phase, in accordance with its specifications, the use of a WUC exhibits excellent power gains. The main CPU is still power gated and program memory (64KB) is either power gated or in retention depending on microcontroller memory configurations. With program and data memory power gated, from -84% to -97% power consumption reduction can be achieved compared to a classical CPU use. Finally, regarding the Computing phase, both configurations demonstrate the same power consumption as the CPU is fully used. However, the small power contribution of the WUC is added (1 to 2 μW estimated from work done in [9]).



(a) Retention Phase for Lower Power Modes



(b) RX / Measure / TX Phase in High Power Mode

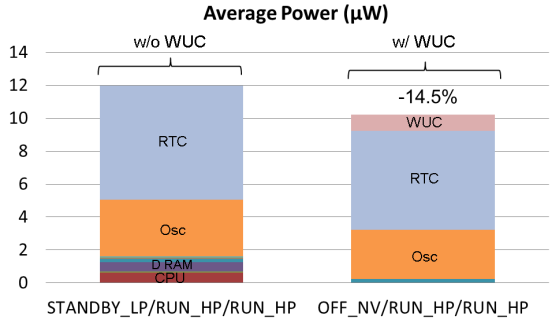
Fig. 4. Power Consumption in Retention and RX/Measure/TX Phase with and w/o Wake Up Controller

V. Microcontroller Simulation in a Complete Applicative WSN Scenario

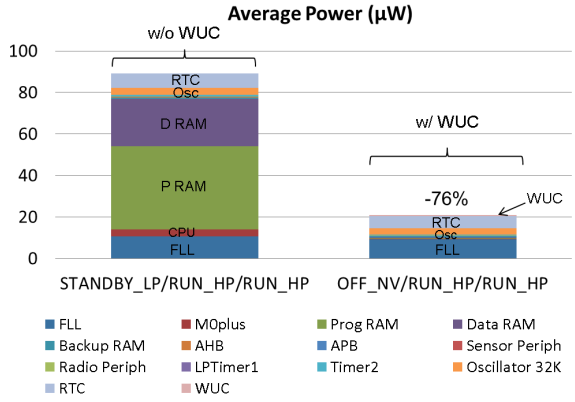
To complete this study, full applicative scenarios are studied with different wake-up periods. For clarity of purpose, we only present a reception scenario but similar power gain results were obtained for a transmission or measurement scenario. A Reception scenario is composed of Retention phase followed by RX and Computation phases. It is set up with ARM M0+, non-volatile data (8KB) and program (64KB) memories and a 32K oscillator and power figures corresponding to FDSOI28 technology. For retention phase, we use the STANDBY_LP mode. During RX and computing phases, we switch to RUN_HP mode corresponding to a frequency of 32 MHz and 1V voltage supply. In RX phase, 128 Bytes are received and 3000 instructions are executed during Computation. Moreover we also consider the transitions power and timing costs between different phases. It includes M0+ reboot cost, supply voltage stabilization duration, FLL/PLL locking time, peripherals configuration. Figure 5 discloses average power consumption results for different wake up periods in the reception scenario. In retention phase, conventional microcontroller power corresponds to the standby mode allowing requisite wake up sources for different applications [7]. We can observe a reduction of -14.5% of the average

TABLE I
POWER MODE DEFINITION IN CURRENT POWER MANAGER

Mode Name	Effect on Vlogic Domain clocks	Effect On Vlogic domain Voltage	Effect on Memory domain Voltage
RUN_HP	Clk Periph On Demand, CPU Clk ON, RAM Clock ON	High Voltage	High Voltage
RUN_LP		Low Voltage	Low Voltage
RUN_ULP		Near Threshold Voltage	Low Voltage
IDLE_HP	Clk Periph On Demand, CPU Clk OFF, RAM Clock OFF	High Voltage	High Voltage
IDLE_LP		Low Voltage	Low Voltage
IDLE_ULP		Near Threshold Voltage	Low Voltage
STANDBY_LP	VLogic + Memory Domain Clocks OFF	Low Voltage	Low Voltage
STANDBY_ULP		Near Threshold Voltage	Low Voltage
OFF		Power Gated	Implementation Dependant



(a) Power Consumption Comparison for 60s wake up period



(b) Power Consumption Comparison for 100 ms wake up period

Fig. 5. Power Consumption comparison (in μW) of a Microcontroller without and with Wake Up Controller in Reception scenario for different wake up periods (Retention phase/RX phase/Computing phase)

power consumption for ultra low activity applications (60 s wake up period in Fig 5a) and -76% power reduction for medium activity applications (100 ms wake up period in Fig 5b). In higher activity applications, reduction in average power consumption will be even more important. Contribution of each sub-module is also studied. For ultra low activity application, the main power contributors are Oscillator and RTC modules, whereas for medium activity application, FLL, memories and processor are predominant. Finally this study shows that the WUC can bring between 14.5% and 76% power reduction when included in a microcontroller sub-system for wireless sensor nodes. Those results depend on the activity ratio and sampling rate but allow us to go further in the WUC implementation.

VI. DISCUSSION AND CONCLUSION

In this paper a power model of targeted microcontroller for WSN has been simulated with different application targets. The objective was to promote the use of specific hardware called Wake Up Controller (WUC) to reduce the node's overall power consumption. The sub-system microcontroller is thus partitioned into this WUC and and a conventional CPU dedicated to computing only. By contrast, the WUC manages current WSN's tasks like data transfer and fine grain power management. This dedicated controller is event-driven to avoid extra power cost during retention phases. We showed that the power gains can reach up to 76% depending on the application activity and sampling rate. Specifications of the WUC and design is in progress. It will be implemented in asynchronous logic to naturally fit event-driven requirements. FDSOI28 technology will be used for performances modulation between different applicative phases.

REFERENCES

[1] Beigne, E. et al., "A 460 MHz at 397 mV, 2.6 GHz at 1.3 V, 32 bits VLIW DSP Embedding F MAX Tracking,"*Solid-State Circuits, IEEE Journal of, vol.50, no.1, pp.125,136, Jan. 2015*

[2] Virantha Ekanayake, Clinton Kelly, IV, and Rajit Manohar, "An ultra low-power processor for sensor networks", *SIGPLAN Not. 39, 11 (October 2004), 27-36*

[3] Bol D. et al., "SleepWalker: A 25-MHz 0.4-V Sub-mm2 7-uW/MHz Microcontroller in 65-nm LP/GP CMOS for Low-Carbon Wireless Sensor Nodes," *Solid-State Circuits, IEEE Journal of , vol.48, no.1, pp.20,32, Jan. 2013*

[4] Yoonmyung Lee et al., "A Modular 1 mm^3 Die-Stacked Sensing Platform With Low Power I C Inter-Die Communication and Multi-Modal Energy Harvesting," *Solid-State Circuits, IEEE Journal of , vol.48, no.1, pp.229,243, Jan. 2013*

[5] Hempstead M., Brooks D., Gu-Yeon Wei, "An Accelerator-Based Wireless Sensor Network Processor in 130 nm CMOS," *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on , vol.1, no.2, pp.193,202, June 2011*

[6] Yoonmyung Lee, Yejoong Kim, Dongmin Yoon, Blaauw, D., Sylvester, D, "Circuit and system design guidelines for ultra-low power sensor nodes", *ACM/EDAC/IEEE , vol., no., pp.1037,1042, 3-7 June 2012*

[7] STMicroelectronics, "Datasheet STM32L063C8 STM32L063R8 DocID025660 rev 2," *April 2014*

[8] ATMEL, "Datasheet SAM D20 doc 42129J-SAM-12/2013," *2013*

[9] Christmann J. F. et al., "Bringing Robustness and Power Efficiency to Autonomous Energy Harvesting Microsystems," *Asynchronous Circuits and Systems (ASYNC), 2010 IEEE Symposium on , vol., no., pp.62,71, 3-6 May 2010*