Analog Integr Circ Sig Process (2013) 77:169–182
DOI 10.1007/s10470-013-0139-2

Modeling, design and implementation of a low-power FPGA based asynchronous wake-up receiver for wireless applications

Pons Jean-François · Brault Jean-Jules · Savaria Yvon

Received: 20 March 2013 / Revised: 27 June 2013 / Accepted: 6 September 2013 / Published online: 28 September 2013

Abstract Power consumption is a major concern for wireless sensor networks (WSNs) nodes, and it is often dominated by the power consumption of communication means. For such networks, devices are most of the time battery-powered and need to have very low power consumption. Moreover, for WSNs, limited amount of data are periodically sent and then the radio should be in idle or deep sleep mode most of the time. Thus using event-triggered radios is well suited and could lead to significant reduction of the overall power consumption of WSNs. Therefore this paper explores the design of an asynchronous module that can wake up the main receiver when another node is trying to send data. Furthermore, we implement the proposed solution in an FPGA to decrease the fabrication cost for low volume applications and make it easier to design, re-use and enhance. To decrease the static power consumption, we explore the possibility of reducing the supply voltage. The observed overall power consumption is under 5 μW at 250 kbps. Moreover, using a new asynchronous design technique, we observed that power consumption can be further reduced.

Keywords Wake-up receiver · Asynchronous · Low-power · Wireless sensor networks · FPGA

1 Introduction

Wireless sensor networks (WSNs) are one of the most studied fields in microelectronics over the last two decades. This is partly due to advances in low power radiofrequency (RF) transceiver architectures that makes it possible for wireless devices to last several days, weeks, or even years while being battery-powered [1, 2]. For example, Table 1 summarizes the power consumptions of several RF transceivers based on the popular IEEE802.15.4 standard that are available on the market (see also Fig. 1). As can be directly inferred from such power consumption figures, constantly powering wireless devices with standard batteries of only hundreds of mAh results in active times that are measured in few days. The same observation could have been done for other industrial standards. Fortunately, most WSNs applications only require using RF transceivers for a portion of their active time. Using idle, sleep or deep-sleep modes of most microcontrollers leads to substantial reduction in power consumption. For example, recent applications such as insect inspired robots or smart electric meters benefit from a low duty-cycle, since data they process are relevant only sparsely over time [3]. For instance, in a smart electric meter example, it may be sufficient to sample and transmit data monthly or even annually. Nevertheless, low power protocols such as IEEE802.15.4 [4], Bluetooth Low Energy [5] or even custom ones need information regarding whether messages

P. Jean-François (✉) · B. Jean-Jules · S. Yvon
Department of Electrical Engineering, Ecole Polytechnique de Montreal, Montreal, QC, Canada
e-mail: jean-francois.pons@polymtl.ca

B. Jean-Jules
e-mail: jean-jules.brault@polymtl.ca

S. Yvon
e-mail: yvon.savaria@polymtl.ca

Present Address:
P. Jean-François
STMicroelectronics, Grenoble, France

P. Jean-François
IM2NP Department, Aix-Marseille University and CNRS, Marseille, France

Table 1 Consumption of available RF transceivers

Transceiver	Consumption			Sensitivity (dBm)
	Emission (mA)	Reception (mA)	Sleep (μA)	
SPZB32W1x2.1	32	28	1.3	−95
STMicroelectronics CC2531 texas instruments	29	24	0.4	−97
MRF24J40MA microchip	23	19	2	−94
MRF24J40MB microchip	130	25	5	−102

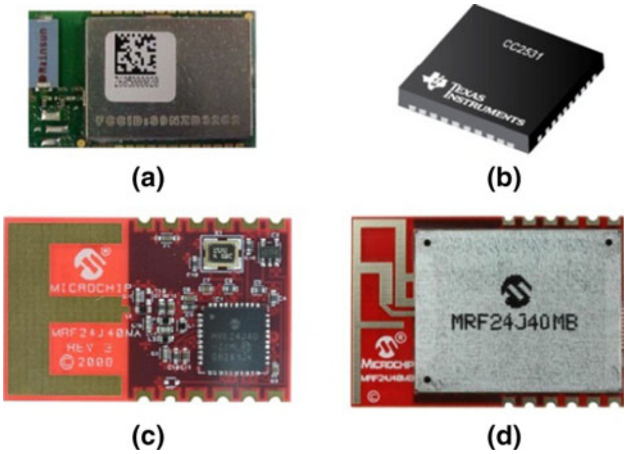


Fig. 1 Referenced transceivers: **a** SPZB32, **b** CC2531, **c** MRF24J40MA and **d** MRF24J40MB

have been sent during the sleep periods. This is typically the case when the device receives commands from another device such as an RF switched light for example. The most common way to do that is to wake the receiver up periodically and ask for missed messages. Despite its ease of deployment, this technique suffers from a major drawback when ultra-low power is a concern: power inefficiency. Effectively, this is due to the fact that the main receiver needs to be woken-up even if no message has been sent. One can then set a long period (or a well-adapted one) between two wake-ups like in the case of smart electric meters, for which the reporting time and period may be known a priori. However, the whole system would then suffer from a lack of adaptability: that could be the case if an update needs to be done in an electric meter. To address this issue, some research has been done [6–10] to anticipate and to take into account the need and throughput of information exchanges, but that is at the expense of some complex algorithms that inevitably induce significant energy consumption. Moreover the periodic wake-up technique requires using always-on parts, such as timers, responsible for wake-up period verification.

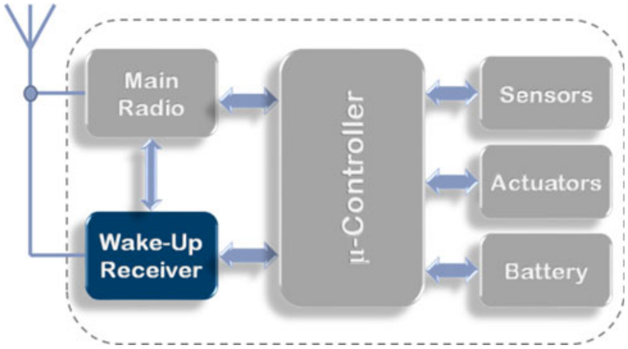


Fig. 2 Position of the WUR in the system architecture

To avoid the need of power consuming algorithms and to enable supporting variable network data throughput, we consider the use of an extra module called the wake-up receiver. Basically, this receiver is intended to sense for specific wake-up messages and to wake-up the main receiver when a relevant incoming message has been found.

Figure 2 presents the block diagram of a generic architecture for a WSNs node comprising a wake-up receiver (WUR). Depending on the application, this WUR could wake-up either the main radio directly or through a microcontroller. Several designs of such WUR could be found in the literature [11–16]. However, most of the time, the WUR is realized with a custom ASIC that is very expensive for low volume applications. To respond to this need, we explore a new asynchronous architecture implemented on a field-programmable gate array (FPGA). FPGAs offer the possibility to decrease development time and costs [production and non-recurring engineering (NRE)]. It is also a lot more flexible than an application-specific integrated circuit (ASIC) when there is a need to respond to changing applications. We also suggest using an asynchronous design that matches the nature of the wake-up signal. For [11–14, 16], the digital demodulator is synchronous, which is inefficient during inactive phases. It could have been possible to use clock gating like in [17, 18], but it would have required extra circuitry such as timer and signal detection to trigger the start of the WUR. To avoid this, we directly use asynchronous logic to design the WUR. To do so, we first use the NULL Convention Logic (NCL)[19, 20] as an asynchronous design guideline for the proposed WUR [21] and then proposed the State-Holding Free NULL Convention Logic (SHF-NCL) [22] as a new way of designing asynchronous circuits with the aim of decreasing resource usage and eventually power consumption.

The rest of the paper is organized as follows; Sect. 2 provides background information regarding WUR in general, together with an overview of the asynchronous design techniques used. We also discuss the advantages and disadvantages of both FPGA implementations and asynchronous circuits. In Sect. 3, the proposed architecture is presented in details. Then in Sect. 4, results about complexity and power consumption are exposed and discussed, followed in Sect. 5, by a comparison with previously reported results. We will finally conclude on the proposed architecture and possible enhancements in Sect. 6.

2 Related information

This section begins with a discussion of modulation techniques leading to the selection of the on–off keying (OOK) modulation. Then, to unravel the advantages of WUR circuits in some applications, an energy consumption model is proposed. Finally, the asynchronous design techniques used to implement the WUR are reviewed.

2.1 Wake-up receiver (WUR)

As explained in Sect. 1, the basic goal of a WUR is to wake the main radio or the microcontroller up when messages need to be received. To do so, it is interesting to use specific wake-up messages that will enable the use of low-power RF receiver as compared with more energy consuming usual architectures, which include for example phase-locked loop (PLL), amplifier, mixers and complex digital baseband. OOK modulation is a popular solution adopted in several low-power wireless applications. It can be used to modulate wake-up messages. As depicted in Fig. 3(a), the basic idea behind OOK modulation is to code logical '0' and '1' using the presence or the absence of the carrier. One variant of this modulation more robust to noise is represented in Fig. 3(b): the logical '0' and '1' are coded with the presence of the carrier for different time duration. A key advantage of this type of modulation in a system leveraging a WUR is the ease with which the signal can be demodulated with a low power RF front end.

Actually, assuming the signal has sufficient amplitude, a simple envelope detector (see Fig. 4) can be used to demodulate the signal. However, for the second type of OOK modulation, more processing is needed to retrieve the emitted data.

Typically, we can divide the architecture of a WUR in two parts: the RF front end and the demodulator as depicted in Fig. 5. The first one is used to remove the carrier while the second one generates the wake-up or interrupt signal. In this paper, we propose a new asynchronous solution for the demodulator part. Additional parts could be added between the antenna and the envelope detector in order to enhance the sensitivity such as: a low noise amplifier, filters or charge pumps. In this work, we assume, as proposed in [23], that an RF front end

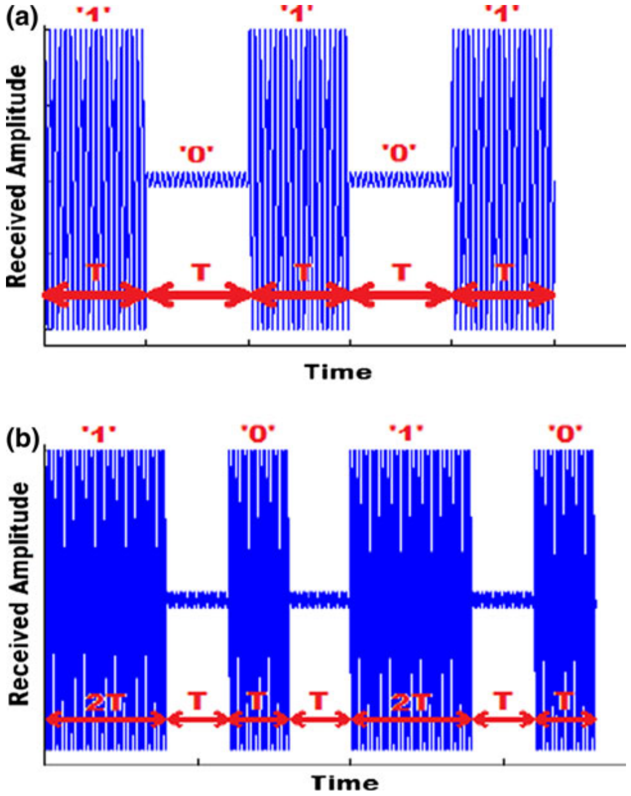


Fig. 3 OOK Modulation .**a** 0 coded as an absent (reduced amplitude) carrier, **b** 0 coded as a shorter full amplitude carrier with contiguous bits separated by time gaps with absent (lower amplitude) carriers

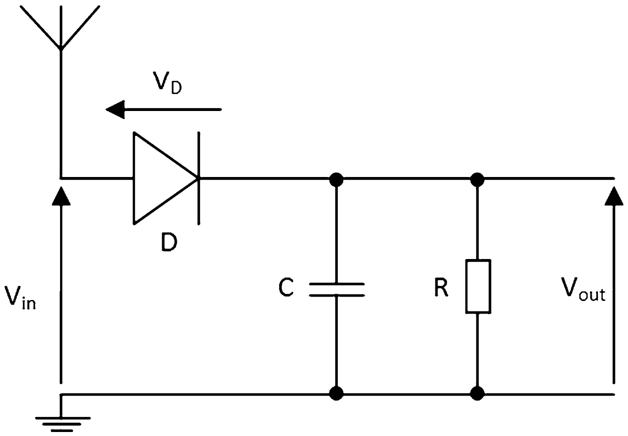


Fig. 4 Envelope detector

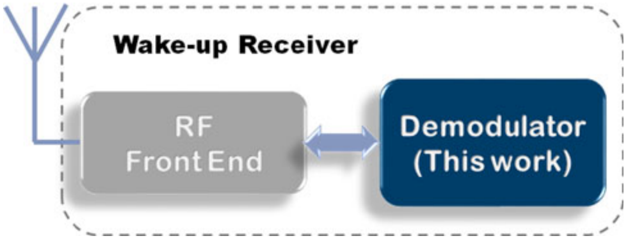


Fig. 5 WUR Parts

consuming as little as 0.1 µW is used. Our study mainly focuses on the digital aspect of the WUR.

2.2 Energy model

When a WUR is available, the energy consumption of the overall system can be considerably improved for some applications. It notably depends on how the radio-communication is used. In order to capture the essence of the impact of a WUR on the energy consumption of wireless devices, we propose in this section an energy model that captures some of their key relevant features. A classification of WSNs proposed by [24] is: wireless body-area networks (WBAN), wireless data collection networks, wireless location-sensing networks (WLSN), wireless multimedia sensor networks (WMSN) and wireless control-oriented sensor networks (WCOSN). This classification is not so representative when it comes to defining energy models. Indeed, the models that were proposed are concerned more with transmission latency and data throughput.[1] Therefore, we propose a new classification of various applications that is more representative of energy consumption profiles. This classification is based on whether the RF part is used continuously (CCOM), periodically (PCOM) or in an event-triggered (ETCOM) way. The notations used in the proposed energy consumption model are summarized in Table 2.

It is expected that nodes are rarely CCOM except for those that are acting like data collectors for a large group of other nodes. One example is the coordinator for the IEEE802.15.4 [4]. Theses nodes must be ON most of the time, and they should preferably be powered by (quasi-) inexhaustible power sources. It is then possible to represent the energy needed by CCOM nodes as:

$$E_{tot,CCOM} = E_{ON} + p_{TX}\, E_{TX} + E_{RX} + E_{act} \qquad (1)$$

The second model concerns nodes with a PCOM application profile, which is frequently used since it requires no extra circuit and is a good alternative to the power hungry CCOM profile. For this kind of application, instead of letting the main radio always-ON, it is periodically turned-ON to seek for pending messages, then turned back to OFF until the next wake-up period. For PCOM devices, we propose the following energy model:

Table 2 Definitions of variables used in the proposed energy consumption model

Parameters	Meaning
$E_{\mathrm{tot,i}}$	Total energy consumed
E_{ON}	Energy consumed during active mode (except WUR)
E_{RX}	Energy consumed while receiving
E_{TX}	Energy consumed while transmitting
E_{OFF}	Energy consumed during sleep mode (except WUR)
E_{act}	Energy eventually consumed by actuators
$E_{\mathrm{WUR\|ON}}$	Energy consumed by the WUR while receiving
$E_{\mathrm{WUR\|OFF}}$	Energy consumed by the WUR while not receiving
p_{ON}	Probability to receive a message
p_{TX}	Probability to transmit a message
T_{batt}	Lifetime of the battery
T_{ON}	Active time in periodic mode
$m_{\mathrm{TX\|ON}}$	Probability to transmit a message while in active state

$$E_{tot,PCOM} = \frac{T_{ON}}{T_{batt}}\left(E_{ON} + m_{TX|ON}\, E_{TX} + E_{RX}\right) + E_{act} + \left(1 - \frac{T_{ON}}{T_{batt}}\right) E_{OFF} \qquad (2)$$

PCOM devices can consume much less energy, and stretch the autonomy in battery powered applications. However, they can be inflexible and energy inefficient when the exchange of messages is rare or have a random distribution over time. It is possible to envision a mode of operation where the wake up time of a device is set right before going OFF. Nevertheless, there are applications where the most desirable wake up time cannot be known at the time the device is put in the sleep mode. The example of a smart electric meter was mentioned earlier to highlight the flexibility issue. Moreover when reactivity is needed, without knowing when and if an event is going to happen, like for alarm systems, the PCOM mode of operation is largely inefficient. For applications for which the exchanged message distribution over time is unknown an ETCOM mode of operation should be used. For ETCOM devices, we can derive the following energy model:

$$E_{tot,ETCOM} = p_{ON}\left(E_{ON} + E_{RX} + E_{WUR|ON}\right) + p_{TX}\left(E_{ON} + E_{TX}\right) + E_{act} + \left(1 - p_{ON}\right)\left(E_{OFF} + E_{WUR|OFF}\right) \qquad (3)$$

While these models have similarities, the main difference between the two last models comes from the fact that in the PCOM type, the duty-cycle $\frac{T_{ON}}{T_{batt}}$ is fixed (although some attempt has been done to adapt it [6, 7, 9, 25]) while for the ETCOM type, it directly depends on the message (or event) occurrences. It is worth noting that while the ETCOM mode of operation can outperform the PCOM mode of operation, in some cases for which the distribution of the data exchange is

[1] See the Network Simulation Framework OMNeT++ for example. (Available at http://www.omnetpp.org/ [accessed June. 22 2013]). This simulation library and framework, being extensible, could be used to verify the proposed energy model.

not sparse over time, the additional continuously consumed energy of the WUR could make the ETCOM model less adapted. In the rest of this paper, we will suppose that the application targeted requires the use of a WUR and that the ETCOM model is well adapted.

2.3 Asynchronous design techniques

One important property of a WUR is that it is always-ON. As a result, its consumption should be as low as possible. Equation (3) makes it clear that whether or not the system is in OFF mode, the WUR consumes energy. Nevertheless, we distinguished $E_{WUR|OFF}$ and $E_{WUR|ON}$ as for some circuits like asynchronous ones, the energy consumed during active (when a wake-up message is received) and inactive mode could be very different. This remark together with the asynchronous nature of the wake-up message leads us to consider an asynchronous circuit. The main benefit for such circuits is that there is no internal activity inside the WUR when no wake-up message is processed. Thus, for wake-up messages with low probability of occurrence, the WUR will spend most of its time doing nothing and then only (or mostly) consume static power due to technology related static leakages.

To design such an asynchronous circuit, we adopted a well-defined asynchronous technique: NULL Convention Logic (NCL) [19, 20]. NCL is one of the so-called quasi-delay insensitive techniques (QDI), which almost avoids the need to make timing assumptions and then make it easier to design and verify circuits. Other techniques, such as bounded-delay, require that the dataflow is controlled by specific delays. While this solution involves less chip area, it is harder to use, reuse and verify, and it is less efficient, since worst case delays limit the operating frequency, while QDI implementations provide operating frequency inversely proportional to average delays in the circuit. Moreover, as explained in the introduction, we decided to use an FPGA as compared to more expensive solutions like an ASIC, despite its potentially better power performances. However, this choice was mainly guided by the desire to have a low cost and fast prototyping platform (while not excluding an ASIC implementation). The choice of an FPGA-compatible option discourages the use of any bounded-delay technique, since precise guaranteed delays are not easily obtained and may require substantial amount of resources.

Among several asynchronous design techniques such as [26–28], we selected the NCL to implement our asynchronous WUR [21], mainly because of the possibility to implement complex circuits, but also because it is well-documented as it is used in industry.[2] The NCL paradigm is first based on the *isochronic fork* assumption that governs all QDI techniques:

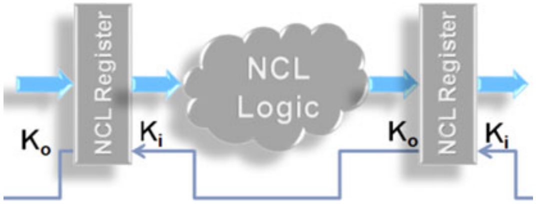


Fig. 6 Representation of the NCL pipeline

> Within basic components, if a transition happens on one end of a fork and this transition has been acknowledged, it is assumed that all the transitions on the outputs of the other branches of the fork have also happened and have been acknowledged when relevant.

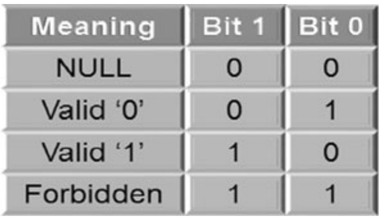
All the modules are then ordered in a pipeline-like fashion as with synchronous circuits: combinational parts are sandwiched between asynchronous registers as depicted in Fig. 6. Moreover, the "synchronization" between two different stages of the pipeline is done using the data itself. To do so, the representation of the data needs to be changed for a complete one.

A complete representation means that the validity of the data is contained in the data itself, unlike with synchronous circuits, for which the validity is ensured with a clock signal. Among other representations such as quad-rail, we decided to use the dual-rail representation for simplicity.

As summarized in Fig. 7, each bit is coded using two wires leading to the possibility to detect valid data by only evaluating these two wires. Then the dataflow of the NCL technique is governed by the alternation of a NULL front and a DATA front that is made possible by using NCL registers, as depicted in Fig. 8, where a basic 1-bit register is represented (the TH22 gates are Müller C-elements and the TH12 gate is a usual OR gate complemented in this case). Thus, DATA propagation is possible only if the next stage requires data and the previous stage is providing DATA. When DATA goes out of the register, it indicates to the previous stage that it is now waiting for a NULL front and it will wait until this front is provided and the next stage is requiring it. Moreover, in order to ensure correct operation of the overall circuit, and most specifically data integrity, two rules must be obeyed by valid NCL circuits as explained in [19, 20]: input-completeness and observability. These rules are enforced by the use of NCL-registers and by the use of 27 NCL state-holding gates also described in [19, 20].

Finally, while implementing the WUR solution in an FPGA, we observed that NCL requires a lot of latch elements. This is mainly due to the state-holding property of

[2] A comprehensive DARPA project report on NCL can be found at https://www.technologyalliance.ms/pointe-innovation/view.php?entryID=3158 [accessed March 10 2013].



Meaning	Bit 1	Bit 0
NULL	0	0
Valid ‘0’	0	1
Valid ‘1’	1	0
Forbidden	1	1

Fig. 7 Dual rail representation

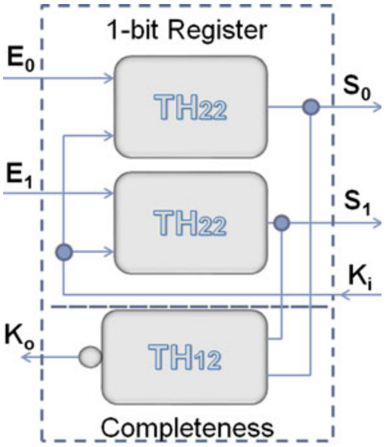


Fig. 8 NCL 1-bit register

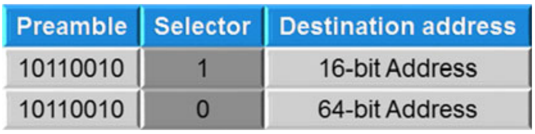
the 27 gates that NCL uses. In order to decrease the use of this specific type of resources, we proposed in [22] a new asynchronous technique called State-Holding Free NCL (SHF-NCL). The basic idea behind the SHF-NCL is to observe that some latches could be deleted if some conditions are met, without compromising data integrity. The details of this technique are out of the scope of this paper and the interested reader could find more information in [22]. In Sects. 4 and 5, we will discuss the advantages of SHF-NCL over NCL in terms of complexity and power consumption.

3 Architecture design

In this section, we detail the proposed WUR solution through its architecture and its modes of operation. The structure of the wake-up messages to which the WUR respond is also explained.

3.1 Wake-up message format

Recalling that the main purpose of the WUR is to wake the main radio up only when relevant input messages are available, while consuming much less power than the main receiver, the need to define a new format for the wake-up message appears obvious. It begins with the use of a modulation that is more easily demodulated as presented the previous section. The main idea is to avoid power hungry components inside the WUR such as mixer, PLL or amplifiers. OOK modulation is an ideal candidate for this purpose, since it only requires passive components in its basic form to demodulate the RF waveform.



Preamble	Selector	Destination address
10110010	1	16-bit Address
10110010	0	64-bit Address

Fig. 9 Frame format for the wake-up message

In order to avoid false alarm and wake-up message miss [29], the wake-up message needs to contain enough information to determine when it is necessary to enable the wake-up. Otherwise, the WUR would be equivalent to an RF energy detection module and would wake-up too frequently the main radio. Consequently, we decided to include information concerning the targeted node into the portion of the message processed by the WUR. The aim of the proposed WUR is therefore to decode this information and to generate the wake-up only when the received information corresponds to a local reference. To ensure flexibility and compatibility with popular standards such as IEEE802.15.4, the information carried by the wake-up message is a node address that can be defined using 16 or 64 bits. The frame format of the wake-up message is then presented in Fig. 9. The first part of the frame is an 8-bit preamble to avoid confusion with other existing interferers. This preamble is followed by a 1-bit selector that indicates whether the following address is 16 or 64-bit long. In order to decrease the resources required for implementing the WUR (antenna, emitter…), the main transceiver could also use OOK modulation.

The basics of the OOK modulation used are discussed in Sect. 2. Considering that the latency between the emission and the decoding of a wake-up message should not exceed a certain time, it is possible to fix the value needed to code ‘0’ and ‘1’. Moreover, to make the demodulation more reliable, the high-time period 2T used for a ‘1’ would be twice the one used to code ‘0’. This increases the discrimination between the two types of symbols. Furthermore, to be able to distinguish two different frames, an inter-frame gap of 5T is required between two wake-up messages. A schematic representation of a part of the wake-up message waveform is represented in Fig. 10. Recall that the high-times of all symbols are separated by an inactive gap period of duration T. It can easily be shown that the maximum total duration of a signal (given when all the symbols are equal to ‘1’ and the address is 64-bit long) is given by:

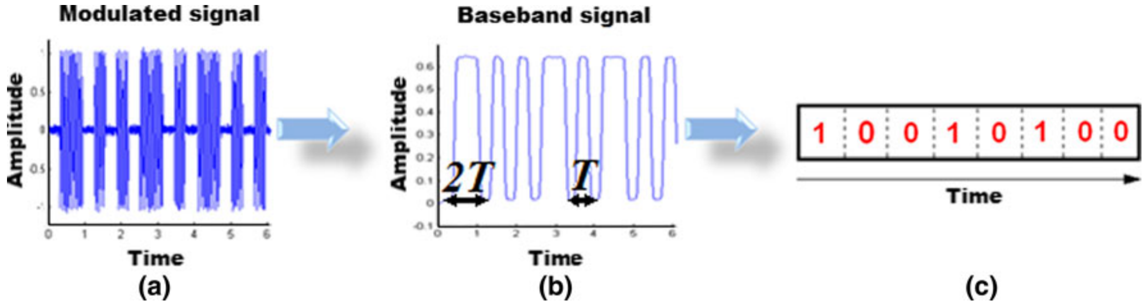


Fig. 10 Wake-up message. **a** modulated, **b** baseband, **c** demodulated

Table 3 Simulation parameters

Fixed values	Implied worst case values	Condition
Min data rate = 250 kbps	Max latency = 293 µs	Only '1' 64-bit address
	Max toggle rate = 724 kHz	Only '0' 64-bit address
Max latency = 10 ms	Max latency = 10 ms	Only '1' 64-bit address
	Max toggle rate = 22 kHz	Only '0' 64-bit address

$$T_{msg} = T_{preamble} + T_{selector} + T_{addr} + T_{IGAP}$$
$$T_{msg} = 20 \times T + 2 \times T + 64 \times 3 \times T + 5 \times T$$
$$T_{msg} = 219 \times T \quad (4)$$

Therefore, imposing either a maximum latency for the transmit time of a wake-up message, or a minimum data rate gives the corresponding value reported in Table 3 for the maximum toggle rate defined as the inverse of T.

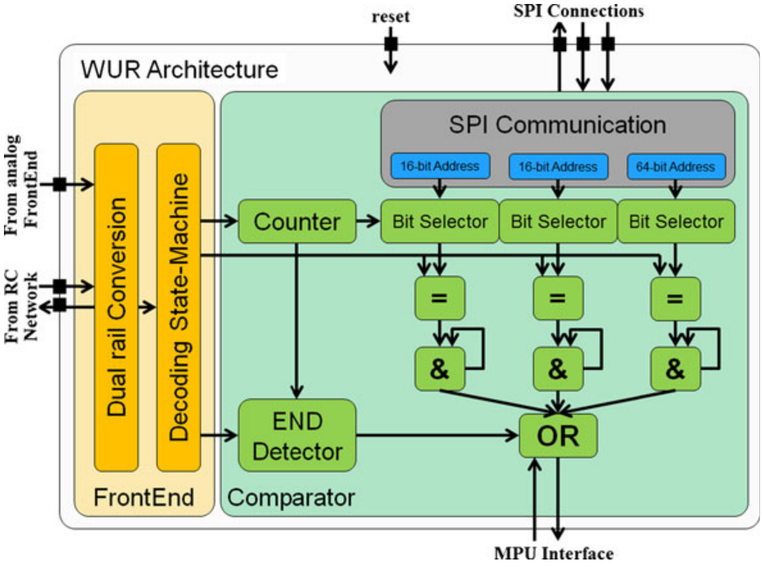
3.2 Overall architecture

The proposed architecture is depicted in Fig. 11. It is assumed that the receiver processes information received from the antenna and upon reception of a suitable message, sends a wake up signal to a main processing unit (MPU). It is mainly divided in two parts called FrontEnd and Comparator, plus a third part used only for configuration through the SPI (Serial Peripheral Interface) protocol shown in grey. Few external components such as RC circuits or delay lines are also required to generate the time constants necessary to decode the message (T, 2T and 5T needed). The power consumption induced by the suitable RC delay elements has been evaluated at around 2.5 nW.

3.3 General operation

The WUR proposed has two different modes: the configuration mode and the normal mode. In the normal mode, it is assumed that a demodulated signal is available at the input of the WUR. The role of the WUR FRONTEND is basically to decode the incoming data. To do so an

Fig. 11 Wake-up receiver architecture



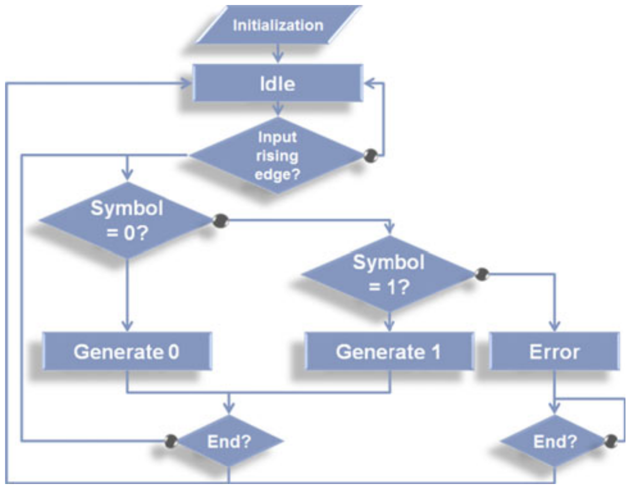


Fig. 12 WUR FrontEnd flow chart

asynchronous state machine is implemented. A simplified version of its flow chart is shown in Fig. 12. When a rising edge is detected, the control of some external RC circuits or delay lines is re-activated. The goal is to detect whether the falling edge appears after a period around T or 2T within a certain interval. Similarly, when a falling edge is detected, a time constant is generated through the external components and the next rising edge of the input demodulated signal should happen after a period around T. If not, this is interpreted as the end of the received message and the validity of this condition is confirmed by comparing the number of received symbols and the expected ones (16 or 64 depending on the selector). If the number of received symbols does not match one of the valid values, an error state is reached and the receiver stays in that mode until an inter-frame gap of 5T is detected to prepare the WUR to receive another message. No action is taken or initiated by the WUR upon declaring a message invalid. Robust communication requirements could set a need for suitable data-link layer protocol features such as acknowledgements, checksums or automatic repeat request. This is left for future research.

In reference to Fig. 12, each received symbol is then sent to the COMPARATOR part that compares simultaneously the received symbol with the corresponding symbols of three reference addresses. To do so, an asynchronous counter is used to select the position of the reference symbol within the reference addresses. This counter is also used to detect whether or not the end of the received message is reached. It also allows the detection of messages that are incomplete or too long. For each received symbol, the result of the symbol-by-symbol comparisons (one for each of the three reference addresses) is stored in asynchronous cumulative AND gates. Note in Fig. 12 that the output of the asynchronous AND gates are fed back to their inputs which makes them 'cumulative' by creating a form of asynchronous state machine. Finally, the results of these three AND gates are combined through an OR gate and this result is taken into account only when the end of the message is detected. This result is the one used to wake-up the main radio or to interrupt the MPU.

The three references addresses can be configured using the SPI module which supports a simplified SPI customized to have a low complexity. The clock required by the SPI module is provided by the MPU and it is therefore inactive during the normal operation of the WUR (implicit clock gating).

Among the three addresses, two of them are 16-bit long and one is 64-bit long. The idea behind this choice is to allow the user to use the network (16 bits) and medium access control (64 bits) addresses specified in protocols such as IEEE802.15.4 and to have the possibility to have a 16-bit long address to target a sub-group in the network. However, the user can change the use of these addresses to adapt them to the application.

4 Results and discussion

In this section relevant results concerning the complexity and the power consumption of the proposed solution are described. The results focus on the use of the NCL technique, however comparisons with the same WUR solution using SHF-NCL [22] and a functionally equivalent synchronous architecture are proposed. These results have been obtained using the AGLN250V2 available with the ACTEL development kit depicted in Fig. 13.

4.1 Complexity

Table 4 summarizes the resources used for the proposed solution using NCL and SHF-NCL. Moreover, the table also reports the complexity of an equivalent synchronous solution that uses counters to estimate the duration of input fronts, so as to determine whether the signal received is '1'

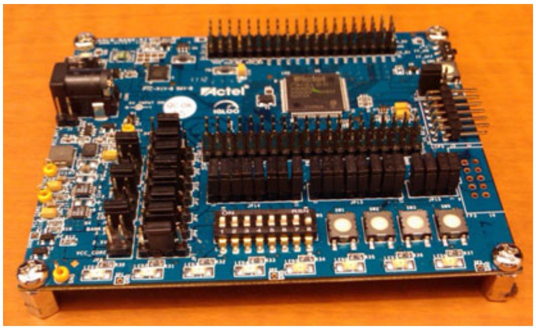


Fig. 13 Actel Development Kit embedding the AGLN250V2 FPGA

Table 4 Resources requirements measured in terms of ACTEL basic elements (*versatile*)

	NCL	SHF-NCL	Synchronous
Combinational resources	2,728	2,579	1,048
Sequential resources	1,547	1,334	263
Total	4,275	3,913	1,311

or a ‘0’. The synchronous design also uses synchronous circuits for address comparisons.

It appears clearly that an asynchronous solution using complete data representation introduces an important overhead in terms of resources used, when compared with an equivalent synchronous solution. This is also due to a simpler processing when decoding data using timers instead of a complex asynchronous state-machine. The synchronous design that was implemented is 69 % less complex than the NCL design. However, the SHF-NCL design was 8.5 % simpler than the NCL design. This resources reduction may seem low; however in some cases it could suffice to change the FPGA size.

4.2 Power consumption

In order to estimate the power consumed by the proposed solution, we used the *SmartPower* tool provided by ACTEL. This software has several modes of operation to estimate the power consumption based on a model predicting the static and dynamic power consumption of the FPGA. This model is known to produce estimates very close to the real values. We used stimuli responses of all the nets of our architecture as inputs of the *SmartPower* tool to have more accurate results. The results for the NCL solution are then summarized in Table 5 for different clock frequencies when the WUR is in configuration mode (SPI) and for the normal mode. The toggle rate used for configuration mode are usual SPI protocol clock rate (1 and 2 MHz), and a typical low power MPU clock (32 kHz). For the normal mode, we used the frequencies previously derived from latency and data rate constraints (see Table 3). Moreover, Table 7 shows the static and dynamic power of the proposed architecture when implemented using NCL, SHF-NCL or synchronous design techniques for an average toggle rate of 22 kHz. For the synchronous design, we assumed that the clock rate is ten times the inverted value of T so as to have around ten points per period T.

Table 5 Power consumption of the NCL solution in the configuration and normal modes for a supply voltage of 1.2 V

Mode	Toggle rate	SMARTPOWER values (µW) [at 25 °C]		
		Static power	Dynamic power	Total
Configuration	32 kHz	29.8	3.5	33.3
	1 MHz		90.0	119.7
	2 MHz		179.2	208.9
Normal	22 kHz		22.5	52.3
	724 kHz		776.9	806.7

Table 6 Power consumption for NCL, SHF-NCL and an equivalent synchronous solution for a supply voltage of 1.2 V and an average input toggle rate of 22 kHz

	NCL	SHF-NCL	Synchronous
Static power (µW)	29.8	29.8	27.7
Dynamic power (µW)	22.5	11.4	223.7

Tables 5 and 6 showed that the dynamic power consumption of the proposed solution is quite high for the targeted low power applications. This is mainly due to the choice of using an FPGA solution with extra unused internal resources, bank of inputs/outputs (I/O) with important capacitances and unused pins.

4.3 Average dynamic power

Tables 5 and 6 reported above supposed that wake-up messages are continuously received, which is in direct contradiction with the ETCOM assumption. Therefore, in Fig. 14, we show the evolution of the average dynamic power consumption with the occurrence frequency, noted f_{WU}, of the wake-up messages. We get these results for an average input toggle rate of 250 kbps and a supply voltage of 1.2 V. The difference in the static power consumption of the synchronous solution is explained by a reduced number of I/O banks used for the same FPGA.

As a result, despite the relatively high dynamic power consumption, one of the main benefits from an asynchronous solution is shown in Fig. 14: the dynamic power consumption depends on f_{WU}. For ETCOM applications, this frequency of occurrence of wake-up messages is

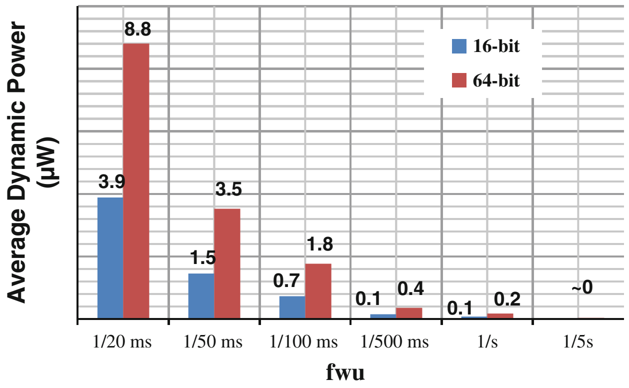


Fig. 14 Simulated evolution of the dynamic power consumption with the occurrence frequency of wake-up messages of the NCL solution

supposed to be low and then, with Fig. 14, it appears that the dynamic power consumption for asynchronous solutions becomes completely negligible when f_{WU} is less than one wake-up message by 5 s (which is well above the expected f_{WU} with some intended applications). If we use 5 µW as a significance threshold (this is the minimum static power consumption we measured at Vdd = 0.8 V, as reported in the next section) for dynamic power, this was only observed when the message rate exceeds 20 per second (see Fig. 14). Clearly, the dynamic power consumption of the asynchronous solutions appears negligible in the ETCOM mode of operation and for most practical cases, the overall power consumption is determined by the static power consumption. Figure 14 also proves that using a synchronous solution is inefficient for this type of application since the dynamic power consumption cannot be neglected. Moreover, Table 6 showed that the SHF-NCL solution reduced dynamic power consumption by 50 % when compared with NCL.

4.4 Static power

Since the static power consumption appears extremely important for ETCOM applications, we tried to decrease it. One solution is to use an ASIC solution that notably avoids unused resources. However, while using FPGAs, it is also possible to decrease the supply voltage. Figure 15 shows the evolution of the static power when the supply voltage is decreased. This technique could be used to decrease the static power consumption as will be explained in Sect. 5. Figure 16 presents photographs of the least significant bit (LSB) of an asynchronous counter implemented on the same FPGA to validate the correct operation of the circuit when the power supply voltage is decreased as shown (1.52 vs. 0.84 V). It also shows that asynchronous circuits can adapt to a wide range of environmental parameters (the output frequency decreases to accommodate the increase of critical delay path while keeping the same functionality).

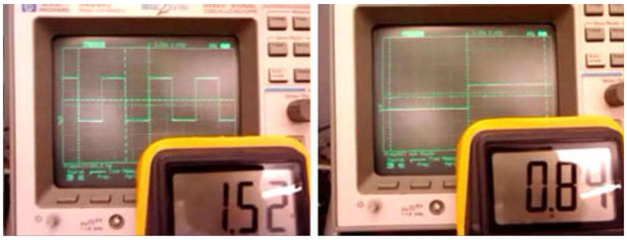


Fig. 16 Capture of the LSB of a working asynchronous counter when VDD is decreased

The results reported in Figs. 15 and 16 showed that for the selected AGLN250V2 FPGA, the supply voltage could be reduced to 850 mV, leading to a measured static power consumption below 5 µW. Figure 16 confirms the functionality of an asynchronous counter that has timing characteristics comparable to our WUR with a supply voltage of a 840 mV. Significant reductions of the dynamic power consumption are also expected when the supply voltage is decreased as proposed.

4.5 System power consumption

Finally, Fig. 17 presents the evaluation of the system power consumption for different frequencies of occurrence of a wake-up message, f_{WU}, for the NCL solution, the SHF-NCL solution and the equivalent synchronous solution.

The results have been derived using (3) based on power consumption components evaluated from simulations, from measured values and with *SmartPower*, while assuming the following components and their specified characteristics for the system (node) architecture:

- The main radio controller is the MRF24J40MA;
- The MPU is an eXtreme Low Power PIC®

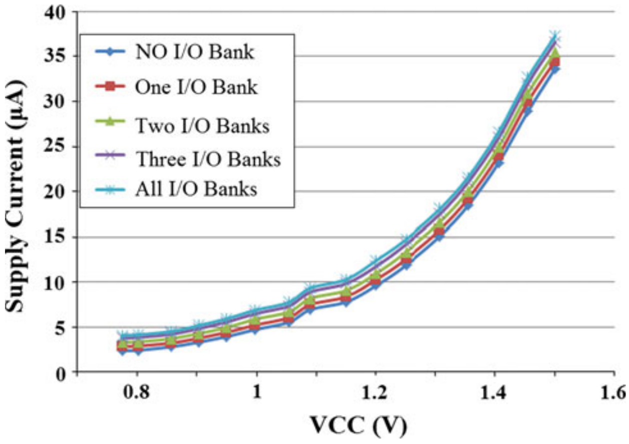


Fig. 15 Measured evolution of the static supply current with the supply voltage

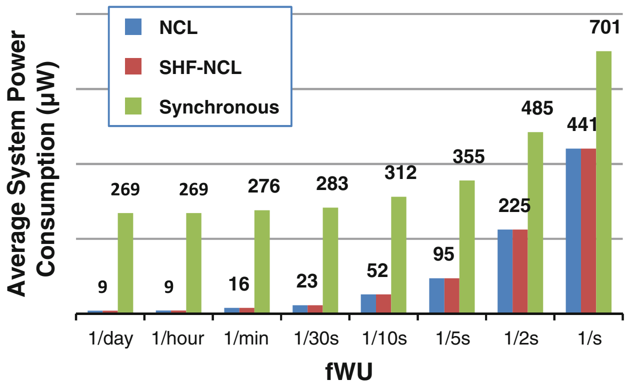


Fig. 17 Evolution of the average system power consumption with the occurrence of wake-up messages

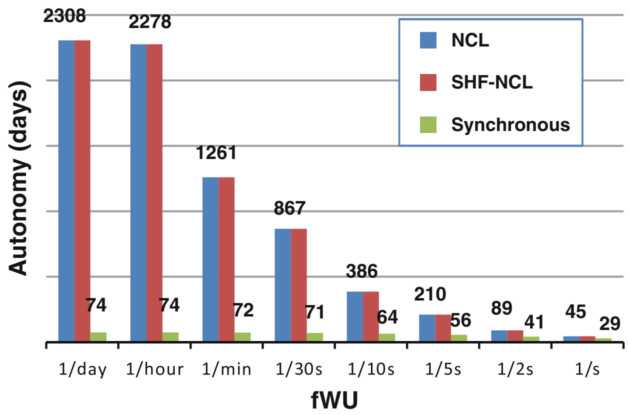


Fig. 18 Evolution of the system autonomy with the occurrence of wake-up messages

These two components benefit from several sleep modes and more specifically the MPU consumes only tens of nanoamps in deep-sleep mode. It is then possible, see Fig. 18, to estimate the system battery lifetime assuming a 400 mAh Li-Ion battery. Similar analyses were performed assuming different battery capacities and even though quantitative results were different, the results were qualitatively the same. Considering the complete system including a microcontroller and the main radio, we showed in Fig. 18 that the expected battery lifetime could be more than 1 year.

5 Comparison

In this section, a comparison with existing WUR designs is provided. To the knowledge of the authors, there is no similar asynchronous FPGA-compatible WUR architecture reported in the literature. However, several WUR architectures, mostly ASIC based, have been proposed as in [11–14, 16, 23].

Therefore in Table 7, we propose a summary of the main properties of different state-of-the-art WUR architectures. The first observation is that, except when compared with [23], the proposed solution has a power consumption in the lower range of previously reported implementations, ranging between 2 and 50 μW, despite the penalties caused by the FPGA. Moreover, the proposed architecture has three reference addresses with which the received wake-up message is compared. This reduces false alarm that would be drastically damageable for the overall power consumption. In [23], the type of application was WBAN. Despite an extremely low reported power consumption, each time a wake-up message not targeting the considered node is received, the main radio is turned-on and the overall power consumption increases strongly. It seems that the probability of wrong detection has been neglected in [23].

Moreover, being FPGA-compatible, our solution could be implemented for low volume and low cost applications, avoiding NRE costs. Besides, it allows for quick prototyping and adaptability to changing application requirements.

Finally, in Table 7 where (A) stands for asynchronous and (S) stands for synchronous, we propose the only solution that offers totally asynchronous digital signal processing. The asynchronous aspect has proved to be very important for low power design in low duty cycle applications since internal activity is directly dependent on the wake-up message presence. Indeed, when no wake-up

Table 7 WUR characteristics comparison

Reference	Type of work	Technology	Type	Power (μW)	Data rate (kbps)	Sensitivity (dBm)	Carrier frequency	Supply voltage (V)	Address	Configurable
This Work	Simulation	FPGA *AGLN250V2*	A	5–30	250	–	2.4 GHz	0.85–1.2	16 bits	Yes
	Digital baseband	CMOS 130 nm							64 bits	
[14]	Realization	ASIC	S	52	50–200	−72	2 GHz	0.5	–	–
	RF Front End	CMOS 90 nm								
[12]	Realization	ASIC	S	2.4	100	−71	868 MHz	1	64 bits	Yes
	Complete	CMOS 130 nm								
[13]	Simulation	ASIC	S	~19	50	~−50	2.4 GHz	1	8 bits	–
	Complete	CMOS 130 nm								
[16]	Partial Realization	ASIC	S	<3.72	>200	−18	–	1.2	8 bits	–
	Digital Baseband	CMOS 90 nm								
[11]	Realization	Mixed ASIC and FPGA	S	12.5	100	~−55	2.4 GHz	1.5	–	–
	Complete	CMOS 120 nm								
[23]	Realization	ASIC	A	0.098	100	−41	915 MHz	1.2	–	–
	Demodulator	CMOS 130 nm								

message is detected, the proposed WUR stays in an inherent sleep mode only consuming static power consumption which is not the case for all the others referenced designs. The power consumption of our WUR certainly goes up compared to its static consumption when there is RF activity that does not lead to detection of a valid message for the specific node (noise, corrupted message or address not matching), but this is an area where our proposed solution could prove highly beneficial. Characterizing power consumption of our WUR when stimulated by invalid messages or by messages targeting other nodes is left for future research. Nevertheless, it is expected that the contribution to the overall energy budget of processing invalid messages should be low. The solution proposed in this paper is the only one allowing for configuring reference addresses, which appears to be a key functionality in a system where there are many addressable nodes with specific complementary functions.

6 Conclusion

In this paper, we proposed an FPGA-compatible asynchronous WUR that enables the wake-up of a main radio only when specific messages are sent. To be energetically efficient, the power consumption of such WUR needs to be well below the power consumption of the main radio and must wake it up while avoiding false alarms and missed messages. In this context, we proposed a totally asynchronous WUR architecture able to decode a received OOK modulated wake-up message.

We proposed a first version of our architecture using the NULL Convention Logic asynchronous design technique. Then, based on an asynchronous design technique proposed elsewhere by the authors, we implemented a SHF-NCL version in order to decrease the resources complexity and the power consumption. We showed that, while decreasing the complexity by only 8.5 %, the SHF-NCL technique reduced power consumption of the proposed WUR architecture by 50 %. Besides, to emphasize the advantage of using asynchronous circuit, we compared the proposed solution with an equivalent synchronous one and showed that despite a three times less complex architecture for the synchronous solution, the dynamic power consumption of a functionally equivalent synchronous design was ten to twenty times higher.

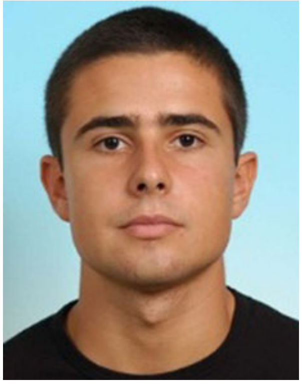
As our WUR is asynchronous, we showed that its overall power consumption is dominated by its static power consumption when the occurrence frequency of wake-up messages is below 20 per second, and it is negligible when the frequency of occurrence is below one message every 5 s. This makes the asynchronous solution much more efficient than the synchronous ones that cannot benefit from inactive wake-up message phases. We then tried to decrease the static power consumption of the FPGA used by decreasing the supply voltage and we reached a point around 850 mV where functionality is retained and the static power consumption is 5 μW.

Consuming only 5 μW, the proposed solution is competitive with expensive ASIC solutions and it shows that WURs implemented on FPGAs are a viable solution for applications targeting several years of autonomy while using small sized batteries.

References

1. Ali, N. A., et al. (2011). IEEE International Conference on *Deployment of MICAz mote for wireless sensor network applications, in computer applications and industrial electronics (ICCAIE)*, 2011, pp. 303–308.
2. Georgoulas, D, & Blow, K. (2008). *Intelligent mobile agent middleware for wireless sensor networks: a real time application case Study, in telecommunications, 2008*. AICT '08. Fourth Advanced International Conference on, pp. 95–100.
3. Wang, T.-l., & Xiong, J.-h. (2010). International Conference on *Design of ladder-type and multi-rate power meter based on zigbee, in intelligent system design and engineering application (ISDEA)*, 2010, pp. 578–581.
4. IEEE Standard for Local and metropolitan area networks—Part 15.4. (2011). *Low-rate wireless personal area networks (LR-WPANs)*, IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006), pp. 1–314, 2011.
5. Bluetooh-SIG. (2010). Bluetooth Specification, in Core Package v4.0, ed.
6. Sang Hoon, L., et al. (2007). IEEE International Conference on *"AMAC: traffic-adaptive sensor network MAC protocol through variable duty-cycle operations," in communications, 2007*. ICC '07, pp. 3259–3264.
7. Kumar, P., et al. (2009). IFIP International Conference on *a real-time and energy-efficient MAC protocol for wireless sensor networks*, in wireless and optical communications networks, 2009. WOCN '09, pp. 1–5.
8. Feng, S., et al. (2006). Optimizing the IEEE 802.15.4 MAC, in TENCON 2006. 2006 IEEE Region 10 Conference, pp. 1–4.
9. Hao, C., et al. (2008). 4th International Conference on *traffic adaptive duty cycle mac protocol for wireless sensor networks*," in wireless communications, networking and mobile computing, 2008. WiCOM '08, 2008, pp. 1–4.
10. Cano, C., et al. (2009). A low power listening MAC with scheduled wake up after transmissions for WSNs. *Communications Letters, IEEE, 13*, 221–223.
11. Durante, M. S., & Mahlknecht, S. (2009). *Third International Conference on, an ultra low power wakeup* receiver for wireless sensor nodes, in Sensor Technologies and Applications, 2009. SENSORCOMM '09, pp. 167–170.

12. Hambeck, C., et al. (2011). IEEE International Symposium on *a 2.4 μW Wake-up Receiver for wireless sensor nodes with − 71 dBm sensitivity*, in Circuits and Systems (ISCAS), 2011, pp. 534–537.
13. Le-Huy, P., & Roy, S. (2008). *Low-power 2.4 GHz wake-up radio for wireless sensor networks*, in networking and communications, 2008. WIMOB '08. IEEE International Conference on Wireless and Mobile Computing, pp. 13–18.
14. Pletcher, N. M., et al. (2008). *A 2 GHz 52 μW wake-up receiver with -72 dBm sensitivity using uncertain-IF architecture*, in solid-state circuits conference, 2008. ISSCC 2008. Digest of Technical Papers. IEEE International, pp. 524–633.
15. Rosello, V., et al. (2011). *Ultra low power FPGA-based architecture for wake-up radio in wireless sensor networks*, in IECON 2011—37th Annual Conference on IEEE Industrial Electronics Society, pp. 3826–3831.
16. Yan, Z., et al. (2011). International Symposium on *a 3.72 μW ultra-low power digital baseband for wake-up radios*, in VLSI Design, Automation and Test (VLSI-DAT), pp. 1–4.
17. Nagaraju, M., & Rakesh, M. (2012). International Conference on, *high-speed and low-power ASIC implementation of OFDM transceiver based on WLAN (IEEE 802.11a)*, in Devices, Circuits and Systems (ICDCS), 2012, pp. 436–439.
18. Chelho, C., et al. (2011). 18th IEEE International Conference on *fully integrated ultra-low-power 900 MHz RF transceiver for batteryless wireless microsystems*, in Electronics, Circuits and Systems (ICECS), 2011, pp. 196–199.
19. Fant, K. M., & Brandt, S. A., (1996). *Proceedings of International Conference on NULL Convention Logic^{TM}: a complete and consistent logic for asynchronous digital circuit synthesis*, in Application Specific Systems, Architectures and Processors, 1996. ASAP 96, pp. 261–273.
20. Smith, S. C., & Di, J. (2009). *Designing asynchronous circuits using NULL convention logic (NCL)*. New York: Morgan & Claypool Publishers.
21. Pons, J., et al. (2012). *An FPGA compatible asynchronous wake-up receiver for wireless sensor networks*, in New Circuits and Systems Conference (NEWCAS), 2012 IEEE 10th International, pp. 373–376.
22. Pons, J. F., et al. (2012). IEEE 55th International Midwest Symposium on *State-holding free NULL Convention Logic^{TM}," in Circuits and Systems (MWSCAS)*, 2012, pp. 322–325.
23. Roberts, N. E., & Wentzloff, D. D. (2012). "A 98nW wake-up radio for wireless body area networks," in Radio Frequency Integrated Circuits Symposium (RFIC). *IEEE, 2012*, 373–376.
24. Mabrouki, I., et al. (2007). *Architecural classification for the design of wireless sensor networks*," in Journées Doctorales en Informatique et Réseaux, Marne La Vallée.
25. Jae-Han, L., & Byung Tae, J. (2008). *Dynamic duty cycle adaptation to real-time data in IEEE 802.15.4 based WSN*," in Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE, pp. 353–357.
26. Singh, N. P. (1981). *A design methodology for self-time systems*. Cambridge: Massachusetts Institute of Technology.
27. David, I., et al. (1992). An efficient implementation of Boolean functions as self-timed circuits. *IEEE Transactions on Computers, 41*, 2–11.
28. Sutherland, I. E. (1989). Micropipelines. *Communications of the ACM, 32*, 720–738.
29. Yan, Z., et al. (2009). IEEE 20th International Symposium on *an analytical model for energy efficiency analysis of different wakeup radio schemes*, in Personal, Indoor and Mobile Radio Communications, 2009, pp. 1148–1152.



Jean-François Pons received the B.Ing. degree in Electrical Engineering from Ecole Polytechnique (France) in 2010 and the M.Sc.A. degree in Electrical Engineering from Ecole Polytechnique de Montréal (Canada) in 2012. He is now a Ph.D. student with STMicroelctronics (Grenoble, France) and IM2NP Department, Aix-Marseille University (Marseille, France). His fields of interest are RF microsystems, digital signal processing and digital design.

Jean-Jules Brault (A'01) received the B.Eng. degree in Engineering Physics, and the M.A.Sc. and Ph.D. degrees in Electrical Engineering from Ecole Polytechnique de Montreal, QC, Canada in 1980, 1983, and 1988, respectively. In 1990, he joined the staff of the Electrical Engineering Department, Ecole Polytechnique de Montreal, where he is presently an Associate Professor. His current interests are mainly in the area of machine learning and artificial neural networks applied to pattern recognition and classification in various areas. He also conducted research in hardware acceleration in bayesian networks.

Yvon Savaria FIEEE (S'77, M'86, SM'97, F'08) received the B.Ing. and M.Sc.A in Electrical Engineering from École Polytechnique Montreal in 1980 and 1982 respectively. He also received the Ph.D. in Electrical Engineering in 1985 from McGill University. Since 1985, he has been with Polytechnique Montréal, where he is currently professor and Chairman of the Department of Electrical Engineering. He has carried work in several areas related to microelectronic circuits and microsystems such as testing, verification, validation, clocking methods, defect and fault tolerance, effects of radiation on electronics, high-speed interconnects and circuit design techniques, CAD methods, reconfigurable computing and applications of microelectronics to telecommunications, aerospace, image processing, video processing, radar signal processing, and digital signal processing acceleration. He is currently involved in several projects that relate to aircraft embedded systems, green IT, wireless sensor network, virtual network, computational efficiency and application specific architecture design. He holds 16 patents, has published 104 journal papers and 358 conference papers, and he was the thesis advisor of 145 graduate students who completed their studies. He was program co-chairman of ASAP'2006 and the general co-chair of

ASAP'2007. He has been working as a consultant or was sponsored for carrying research by Bombardier, CNRC, Design Workshop, Dolphin, DREO, Genesis, Gennum, Hyperchip, ISR, LTRIM, Miranda, MiroTech, Nortel, Octasic, PMC-Sierra, Technocap, Thales,Tundra and VXP. He is a member of the Regroupement Stratégique en Microélectronique du Québec (RESMIQ), of the Ordre des Ingénieurs du Québec (OIQ), and was a member of CMC Microsystems board since 1999 and chairman of that board from 2008 to 2010. He was awarded in 2001 a Tier 1Canada Research Chair (www.chairs.gc.ca) on design and architectures of advanced microelectronic systems. He also received in 2006 a Synergy Award of the Natural Sciences and Engineering Research Council of Canada.