



A Low Cost System for Self Measurements of Power Consumption in Field Programmable Gate Arrays

Juan P. Oliver^{1,*}, Francisco Veirano¹, Diego Bouvier¹, and Eduardo Boemo²

¹Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay

²Digital System Laboratory, School of Engineering, Universidad Autónoma de Madrid

(Received: xx Xxxx xxxx; Accepted: xx Xxxx xxxx)

This paper presents a specific system to measure power consumption in FPGAs. It is based on a current-frequency conversion block. This tool allows an application running inside the FPGA to know its own consumption in real time. The proposed system includes a small external circuit that performs current-to-frequency conversion and an associated VHDL core designed to register the power consumption inside the FPGA. The external circuit is built with off-the-shelf low-cost components and has low power consumption; the VHDL core uses very few on-chip resources. The complete system has an error lower than 1% in the FPGA power consumption measurement. As it can be triggered by internal complex conditions, it makes it possible to obtain detailed power consumption profiles of FPGA designs using a very simple procedure.

Keywords: Field Programmable Gate Arrays (FPGA), Power Consumption Measurement, Energy Aware Design.

1. INTRODUCTION

Field programmable gate arrays (FPGAs) are employed in a wide range of high-end applications: industrial and communications equipment, embedded electronics, custom digital signal processors, and computer systems. However, high throughput and intensive use of FPGA resources always lead to high power consumption. Yet, power consumption is a variable of interest for both chip manufacturers and EDA developers.

In spite of their disadvantages in terms of power, FPGAs are gaining ground in the area of battery-operated systems, as evidenced by the large number of recent publications. In Refs. [1] and [2], surveys of application of FPGAs in Wireless Sensor Networks (WSN) are presented. The development of a node with a mixed design based on a microcontroller and an FPGA for the processing layer is presented in Ref. [3], and power management techniques for the same platform are analyzed in subsequent works.^{4–6} In Ref. [7], the authors use an FPGA as a high-performance coprocessor attached to an external ZigBee transceiver. Similar approaches using FPGA boards in autonomous systems are presented in Refs. [8–11]. In Ref. [12], a reconfigurable node that

includes a low-power flash-based FPGA is explained. In Ref. [13], a Virtex-4 FPGA is applied to implement a dynamically reconfigurable sensor node. These works show that including reconfigurable devices in WSNs offers certain benefits in terms of high performance and flexibility, but power consumption and power management are two of the main drawbacks.

In order to employ FPGAs in low-power applications, it is important to know in detail the power budget for different modes of operation. This can be done using power estimation tools. However, in several cases, these tools present high error rates.¹⁴ The development of FPGA power consumption estimation tools and their integration into commercial and academic FPGA EDA packages is relatively recent in comparison with synthesis tools: Xilinx announced Xpower in December 2000;¹⁵ a power model was integrated into the VPR tool in 2002, and it is commonly employed by the research community;¹⁶ and finally, Altera introduced PowerPlay in 2004.¹⁷

Area-time-power (ATP) are the three main physical variables that determine a circuit feature; but it is power that presents the greatest difficulties in its estimation, for two main reasons. First, it depends on each circuit node activity and hence, on the input vectors. For example, a state-machine can receive thousands of data and remain in an idle state, while a particular sequence of a few bits can

*Author to whom correspondence should be addressed.
Email: jpo@fing.edu.uy

move it from one state to another. The second source of error in power estimation comes from the spurious transitions or glitches. Their propagation on a combinatorial circuit strongly increases the activity on most of the nodes of the circuit and therefore the power consumption.¹⁸

Several works have reported direct on-board FPGA power measurements. Two of the first studies using Xilinx XC3000 and XC4000 series were published by Refs. [18] and [19]. In Ref. [20], dynamic power consumption was analyzed for Virtex II devices, defining three factors that contribute to total power dissipation: capacitance, resource utilization, and switching activity. Comparisons between power measurements and estimations in different Virtex-based dynamically reconfigurable cases were developed in Ref. [21]. A similar study between Xpower and PowerPlay was done in Ref. [22].

A technique for early estimation of FPGA dynamic power consumption was presented in Ref. [23]. Applying this methodology in a Spartan-3 device, the tool error was 18% with respect to the measured value. Another work,²⁴ presented several differences between measured and estimated power. The resulting error varied from 15% to 208% for the Xilinx devices (Virtex II-Pro, and Spartan 3), and from 5% to 32% for the Altera ones (Cyclone II). Security cores were utilized as benchmark circuits. In a more recent work,²⁵ the same authors identified the effects of different synthesis settings on FPGA power consumption in a Virtex II-Pro. They stated that XPower Analyzer overestimates power consumption in a range from 17% to more than 200%.

A different idea based on the utilization of switched capacitor was presented in Ref. [26]. This method makes it possible to measure the static and dynamic energy involved in each cycle. The authors reported that Xpower highly overestimates the predicted values. Another study²⁷ presented a dynamic power estimation methodology for the embedded multipliers in Xilinx Virtex-II PRO chips. On-board measurements of MicroBlaze power using Chip-Scope were performed in Ref. [28], where estimation errors of 34% were reported. Finally, a more recent work²⁹ showed a new estimator with errors within 10% when compared with both on-board measurements and low-level XPower estimations. Binary divider cores are utilized as benchmark circuits.

There are multiple methods for measuring power consumption.^{30–32} Since voltage is constant in most applications, the measurement of the current is enough to obtain the power. A shunt resistor is usually placed between the power supply and the device under test. The voltage drop is then recorded with an A/D converter,^{33,34} or with a voltage-to-frequency converter (VFC).³⁵ In other cases, a current mirror circuit is utilized instead of the shunt resistor. The resulting current is again converted to voltage and acquired with an oscilloscope,³⁶ or utilized to repeatedly charge a capacitor and count charge cycles.³⁷ Another way

of measuring energy is to power the device under test with a pair of capacitors that are repeatedly charged and discharged.^{38,39}

Most of the previous methods need complex electronic instruments like PCs or oscilloscopes, and are not suitable for field measurements. Moreover, some of these techniques introduce a certain amount of ripple in the power supply. As far as we know, the only work that proposes a system for in-field measurements in FPGA-based systems and uses these measurements inside the FPGA is Ref. [40]. However, it focuses on the external circuit, while the processing inside the FPGA is not well described and the resources used are not quantified. Additionally, the consumption of the external circuit is not detailed and neither is the impact on the total consumption.

In this work, we present a very simple and low-cost system to monitor power and energy for a runtime FPGA application. This system consists of an external circuit and an internal VHDL core instantiated in the FPGA. The external circuit uses a voltage-to-frequency converter optimized in power consumption and cost. The internal VHDL core completes the system by performing the measurements and storing the data. This method is suitable to know the power and energy of any FPGA-based circuit. It enables a wide range of applications: the FPGA can make decisions according to the consumed energy; it may, for example, turn off part of the system and change the timing of operations or the clock frequency. Additionally, it can be employed as a tool to characterize the consumption of the FPGA, allowing it to obtain detailed power profiles. Further, the internal VHDL core has the capability to trigger the measurement with internal complex conditions, which makes it possible to obtain the consumption of a circuit in a particular state, routine, or processing mode.

The remainder of the paper is organized as follows: The proposed system, including the external measurement unit and the associated VHDL core, is detailed in Section 2. Results are presented and analyzed in Section 3. Finally, main conclusions and future works are summarized in Section 4.

2. THE PROPOSED CURRENT CONSUMPTION MEASUREMENT SYSTEM

This section presents the proposed system. The selected method for the power consumption measurement was based on the following strategies:

- (1) the external circuit power consumption needed to be negligible in comparison with the FPGA consumption;
- (2) the external circuit should be built with inexpensive and off-the-shelf components;
- (3) the FPGA resources consumed by the internal VHDL core should be as low as possible.

In order to fulfill these requirements, a current-to-frequency converter-based system was selected. The system described below is based on a previous work applied

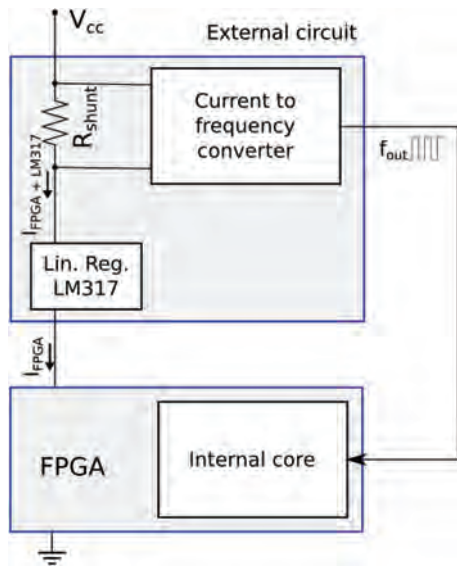


Fig. 1. Block diagram of the proposed circuit.

to microcontrollers.^{41,42} It has two blocks: an external circuit inserted in the power source and a flexible VHDL core integrated in the FPGA, which is responsible for the measurement and storage of the consumption. Figure 1 shows a block diagram of the proposed system.

2.1. External Circuit

The circuit developed to self-measure the consumption of the FPGA consists of two stages (see Fig. 2). The first one is a current-to-current converter that samples the input current, while the second is a current-to-frequency converter.

Thus, the resulting signal has a frequency that varies linearly with the FPGA current consumption.

In the first stage, a shunt resistor R_{shunt} generates a voltage drop at the input of the LM317 regulator, configured to provide an output of 1.2 V which is used to power the FPGA core.

As the I_{LM317} is typically $50 \mu A$ (max $100 \mu A$) is negligible compared with the I_{FPGA} , which for most of the RAM-based FPGAs, is in the range of 5 mA to 100 mA. Then, assuming that $I_{LM317} \ll I_{FPGA}$, the voltage in the non-inverting input of the operational amplifier is

$$V_{in} = V_{cc} - I_{FPGA}R_{shunt} \tag{1}$$

This voltage is converted into a current and scaled down by a factor proportional to the resistor R_{gain} . Thus, the result is an output current equal to

$$I_{out} = I_{FPGA} \frac{R_{shunt}}{R_{gain}} \tag{2}$$

The second stage is a current-to-frequency converter built around a low-power version of a 555 circuit, the CSS555C.⁴³ Ideally, the output current from the previous stage charges the internal capacitor C_T until $V_{CC} \times 2/3$ (the negative reference of the internal comparator Comp1 in the CSS555C). At this point, the transistor is enabled, forcing the discharge of C_T to a voltage equal to $V_{CC}/3$ (the positive reference of Comp2).

The function of the external voltage reference V_{ref} is to take into account possible variations in V_{CC} . With this reference connected to the internal comparator Comp1,

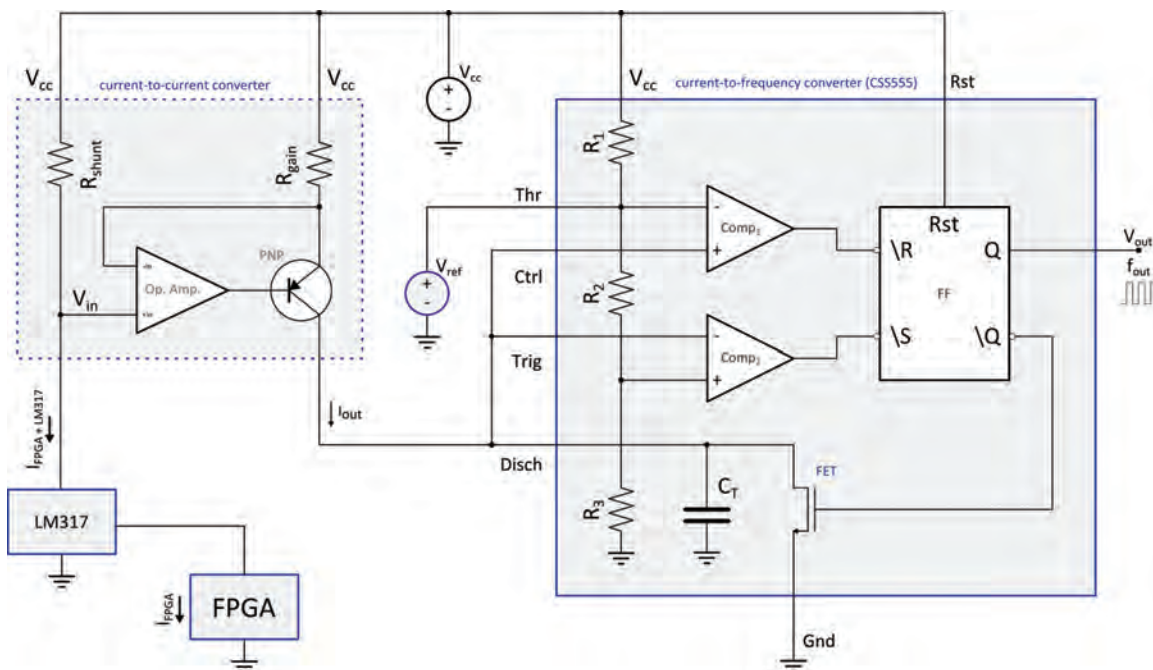


Fig. 2. Block diagram of the external circuit.

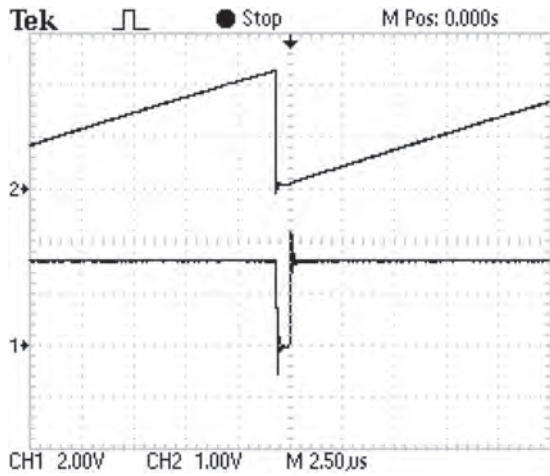


Fig. 3. Voltage in capacitor C_T and output of CDS555C.

the upper and lower limits in the charge–discharge cycle are now V_{ref} and $V_{ref}/2$. Thus, the cycle does not depend on V_{CC} .

Due to the combined effects of the delay in the comparators, the delay and break-before-make system in the flip-flop, as well as the fact that the capacitor C_T is charged and discharged through a low resistance path, the two limits fixed by the voltage reference V_{ref} are exceeded. During the discharge cycle, the capacitor does not stop at $V_{ref}/2$; instead, it reaches 0 V. Moreover, in the charge cycle, the capacitor reaches a voltage greater than V_{ref} .

Figure 3 shows the effect described above. Channel2 (CH2, upper signal) samples the voltage in C_T . In channel1 (CH1, lower signal), the output of the current-to-frequency converter is shown. It can be seen that the voltage in C_T goes over the reference V_{ref} (2.048 V) and after reaching 0 V, it stays there for a certain time.

Naming t_d as the interval between the point where the capacitor C_T reaches V_{ref} and the time in which the voltage in C_T starts increasing from 0 V, the frequency of the output is

$$f(I_{FPGA}) = \frac{1}{(R_{gain} C_T V_{ref} / R_{shunt} I_{FPGA}) + t_d} \quad (3)$$

As can be seen from (3), the linearity of the output frequency is affected by the value of t_d . Therefore, in order to select the components of the circuit properly, this value must be known. According to laboratory measurements, t_d is approximately 1.2 s. This value is also confirmed by the datasheet of the CSS555C for the default settings: micro power and standard power supply of 3 V. The capacitance C_T is fixed during the CSS555C fabrication and has a 1% tolerance and a temperature coefficient of 0.005%/°C. In addition, as the value of the R_{shunt} resistor generates a voltage drop in the input of the regulator LM317, the lowest value must be selected; therefore, a 0.5Ω resistor was used.

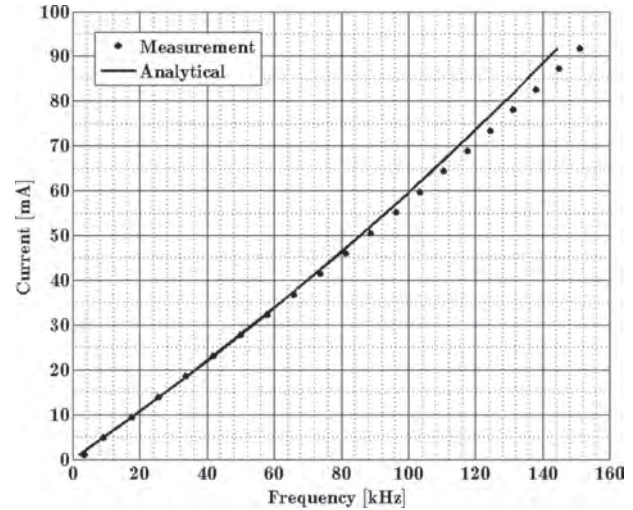


Fig. 4. Current consumption versus frequency of the output signal.

To select the operational amplifier, the key parameter is the input offset V_{off} . As this voltage is added to the one obtained by the R_{shunt} resistor, the V_{off} value must be low enough to obtain a voltage at the non-inverter input of the operational amplifier independent of the offset. According to (1), the condition $V_{off} \ll I_{FPGA} R_{shunt}$ must be assured. As the current to be measured in this work ranges between 5 mA and 100 mA, the maximum input offset must be 250 μ V. To fulfill this requirement, the TLV2460 operational amplifier was selected. It has an input offset voltage of 100 μ V, low-power consumption (500 μ A), and rail-to-rail inputs. Additionally, the chip is compatible with a power source of 3.3 V.

The voltage reference was selected so that its output is similar to the value in the non-inverter input of Comp1. This voltage is $V_{CC}2/3$ and the power source is 3.3 V; therefore, it must be as near as 2.2 V possible to avoid a high current in the output of the reference. Taking this condition into account, the LT6656B-2.048 voltage reference was selected. It has an output of 2.048 V, very good temperature stability (12 ppm/C), and low power consumption (1 μ A).

Finally, the R_{gain} resistor was selected (1.5 kΩ) in order to obtain the best linear fit of (3) in the range of interest. Figure 4 shows the relationship between the input current and the output signal frequency. The solid line graphs this relationship using (3), while the dotted line shows the experimental measurements obtained.

The described circuit can be used to measure the input current in any block, if a shunt resistor can be previously inserted in the voltage power source. In commercial FPGA boards, this action can be done by desoldering the regulator. Another condition is an operating power supply higher than 2.1 V. This voltage is the minimum value at which the reference V_{ref} can work.

Table I summarize the total consumption of the external circuit under different loads.

Table I. External circuit consumption.

I_{FPGA} (mA)	Consumption (mA)
10	1.02
40	0.900
60	0.465
120	0.280

2.2. Internal Core

As presented in the previous section, the external circuit provides the FPGA with a signal whose frequency f_{input} is proportional to its instantaneous power consumption. This section presents the method for measuring and storing this signal. Considering the range of consumption of the FPGA model to be measured, the value of f_{input} can swing in the range of (4). Additionally, a maximum error of 1% was stated as a requirement for the measurement.

$$f_{\text{input}} = [1 \text{ KHz}, 200 \text{ KHz}] \quad (4)$$

There are two basic methods for measuring f_{input} . One method involves using a reference clock with a frequency higher than the corresponding value of the input. Therefore, the number of clock periods that fits inside a period of the input signal is proportional to the input signal's frequency. The maximum error is \pm one clock period. Thus, considering that the highest f_{input} is near 200 kHz, in order to maintain the error under 1%, the reference clock signal must be higher than 20 MHz.

The other method uses a reference clock whose frequency is lower than f_{input} and counts the number of periods of f_{input} that fit inside a period of the clock signal. This implies a measurement of the average value of f_{input} . The maximum error obtained in this case is \pm one f_{input} period. In order to maintain an error below 1%, at least 100 input pulses must be measured. Thus, the reference clock frequency must be at least 100 times lower than the lowest value of f_{input} (1 KHz). In this mode, the frequency of the reference clock signal must be lower than 10 Hz.

The optimal method depends on the application. If the goal is to analyze the power during a short time (ST), for example, in order to measure the energy of a given processor's routine, the first method is preferred. On the other hand, if the aim is to obtain the total consumption or its average value over a long operation time (LT), the second alternative is more suitable.

In this paper both methods were implemented. First, the one used to measure chip energy during long periods is explained in detail. The internal circuit counts the f_{input} and saves this count at a reference frequency of 2.44 Hz. The difference between two consecutive counts is the number of pulses in a clock period. Using this difference, it is possible to obtain the mean frequency, and thus the mean current consumed by the FPGA in that period.

The length of the recording data depends on the number of bits used in the pulse counter and the available

Table II. Resource usage altera cyclone III EP3C16F484.

Circuit	Combinational functions	Registers
Periodic SEM (LT)	263	207
Spaced SEM (LT)	397	300
Spaced SEM (ST)	389	268

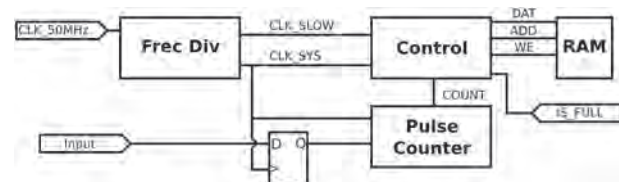
memory. For example, a 32-bit counter of input pulses allows the designer to take samples for more than 5 hours ($2^3/200 \text{ kHz}$) before overflow. As the available memory inside the FPGA is 56 kB, and each measure is 4 bytes long, the measurement time is limited by the internal memory. If 32 kB are used to store the data, then the memory will be full in approximately 1 hour ($32 \text{ kB}/(2.44 \text{ Hz} * 4 \text{ B})$).

Finally, an important characteristic of this method is that the total consumption can be obtained from the reset point of the system because the total number of the input pulses is saved.

Two options of VHDL modules were implemented with this method, and the resource usage of each of them is shown in Table II for the particular FPGA used. The first one does a periodic capture of the accumulated count of input pulses (Periodic SEM LT), while the second one only saves it if an enable signal is activated (Spaced SEM LT). The second version is attached to an asynchronous event so it needs a time stamp in order to save the time of each data saved. This implies an extra counter to be saved, more memory and, as a result, shorter recording of data, but it allows the designer to save consumption values of specific events.

In the periodic implementation, a counter is incremented if a rising edge is detected in the input signal. Additionally, the accumulated count is saved in an internal RAM memory, at a rate of 2.44 Hz. An output signal, named ISFULL, is activated when the memory is full and the circuit stops saving data. The number of bits used for the counter and the memory available for the experiment are configurable and, depending on this setting, the maximum length of the data recording is set.

Figure 5 shows a block diagram of the implemented circuit, where the three main blocks can be seen. The first one, FREQ DIV, uses PLLs from the FPGA to obtain the system clock (CLK SYS 2 MHz) and the slow clock (CLK SLOW 2.44 Hz). The last one is employed to save the accumulated count periodically. The second block, PULSE COUNTER, is a finite-state machine that increments its

**Fig. 5.** Internal block diagram of periodic SEM.

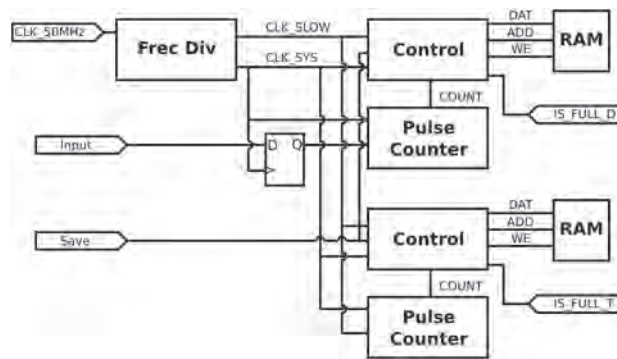


Fig. 6. Internal block diagram of spaced SEM.

counter in each rising edge of the input signal. The actual value of the counter is available at the output of this block. Finally, the control block saves the count in the RAM in each rising edge of CLK SLOW. This block is responsible for saving the count in the RAM correctly and activating the ISFULL output when the assigned memory is full.

The second version, the spaced one, is basically the same as the periodic one but it adds another input signal SAVE. It is activated to allow the control block to save the count (and its corresponding time stamp) in the RAM. Additionally, another pulse counter, control block, and RAM are needed for the time stamp. The time stamp counter stores edges in the CLK SLOW. This value and the input pulse count are saved in the RAM, if a clock edge is detected in signal CLK SLOW and the input SAVE is activated. Figure 6 shows the block diagram of this implementation.

Finally, to obtain detailed power consumptions in a short period of time, a third version was developed (Spaced ST SEM). This version was done only for the spaced implementation and the frequency measurement technique based on counting the number of clock periods that fits inside the input signal. In this case, the fast clock was selected to be 20 MHz in order to obtain an error lower than 1%, since the highest input frequency is 200 KHz. The resource

Table III. Summary of the implemented internal blocks.

Circuit	Main properties
Periodic SEM (LT)	<ul style="list-style-type: none"> —Allows to obtain the mean current consumption of the circuit. —Long term consumptions acquisitions. —Continuous measurements, not triggered —Energy counter.
Spaced SEM (LT)	<ul style="list-style-type: none"> —Allows to obtain the mean current consumption of the circuit. —Long term consumptions acquisitions. —External triggered. —Allows to obtain energy consumption between triggers.
Spaced SEM (ST)	<ul style="list-style-type: none"> —Allows to obtain detailed current consumption profiles. —Short term consumptions acquisitions. —External triggered. —Allows to obtain energy consumption between triggers.

usage of this version is also detailed in Table II (Spaced SEM (ST)). Table III summarizes the main properties and usage of each implemented internal VHDL block.

The proposed system fulfill the three objectives pursued at the beginning of this section. The current consumption of the external circuit is less than 1 mA which is negligible compared to the majority of the FPGAs which consume several mA. The external circuit was built using of-the-shelf low-cost components. Finally, Table II shows that the logic elements used by the internal circuit are less than 2% of the area of the FPGA (Cyclone III EP3C16F484).

3. EXPERIMENTAL RESULTS

This section presents the setup of the system and some experimental results. The external circuit is connected between the voltage source and the FPGA, and the output of this block is connected to the FPGA through a general purpose pin (GPIO). At the same time, the VHDL modules that measure this signals frequency are instantiated and incorporated into the device under test (DUT).

In order to show the accuracy of the proposed circuit, the results of each experiment are compared with the values obtained through the measurement of the voltage drop in an external shunt resistor with a multimeter.

After each measurement, the content of the RAM is exported in .HEX format. Then, it is processed to obtain the frequency measurement and the corresponding current consumption. In order to obtain the input signal's frequency, the difference between two consecutive accumulated count values are multiplied by the frequency of the CLK SLOW signal, in accordance with Section 2.1.

The experiments that are presented in this section try to cover a wide range of applications that use FPGAs. The first one studies the self-consumption of both proposed systems for long time experiments, the periodic and the spaced one. No DUT is added in the FPGA. The second experiment uses the periodic LT system to measure the consumption of the OpenMSP430⁴⁴ processor as a DUT. The third one measures the power consumption of a routine running inside the openMSP430 by means of the spaced ST implementation of the self energy measurement. Next, the power consumption of a 32-bit multiplier, a pipeline FFT calculator, an AES encryption circuit and a circuit that implements the physical layer of the 802.15.4 protocol, are analyzed with the periodic LT arrangement.

3.1. Self-Power Consumption

The duration of the experiment was set to 12 s and the results are shown in the first row of Table IV. They correspond to the mean current consumption value. The small difference between the direct measurement using the shunt resistor and the proposed method is caused by the linear frequency-current relationship employed. This difference is not due to the frequency measurement itself, since this was checked using an oscilloscope.

Table IV. Experimental results.

Circuit	SEM (mA)	Direct measurement (mA)	Error %
Periodic LT SEM	11.54	11.45	0.78
Spaced LT SEM	11.54	11.51	0.20
OpenMSP430	32.61	32.85	-0.74
periodic LT SEM			
OpenMSP430	32.61	-	-
spaced ST SEM			
OpenMSP430 during reset	17.49	17.60	-0.63
periodic LT SEM			
32 bit mult active	58.50	58.80	-0.51
periodic LT SEM			
32 bit mult 1 clock enable	14.11	14.05	0.43
periodic LT SEM			
32 bit mult 2 clock gating	11.71	11.65	0.51
periodic LT SEM			
FFT periodic LT SEM	170.57	169.44	0.66
FFT clock enable	12.92	13.00	-0.66
periodic LT SEM			
802.15.4 physical	20.73	20.54	0.89
periodic LT SEM			
802.15.4 physical clock enable	12.55	12.51	0.32
periodic LT SEM			
AES periodic LT SEM	113.9	114.3	-0.37
AES clock enable	12.81	12.84	-0.25
periodic LT SEM			

3.2. OpenMSP430

The second measurement was performed using an OpenMSP430 as the DUT. In this case the periodic LT implementation was used. This experiment shows that the proposed method is suitable to dynamically analyze or control complex processors. Two cases were analyzed: the first one executing a test program, and the second one measuring the power during the reset state of the microcontroller. Both results are presented in Table IV.

3.3. OpenMSP430 Routine

The aim of this experiment was to measure the consumption of a subroutine running inside the OpenMSP430. To this end, an auxiliary circuit was designed that captures the CALL and RET opcodes from the data bus and generates a trigger signal to activate the Spaced ST self energy measurement. This design makes it possible to obtain the consumption of a single subroutine that runs inside the microcontroller.

3.4. 32 Bit Multiplier

In this case, the DUT is a 32-bit multiplier implemented with two types of standby modes: global clock gating and clock enable applied to all the flip-flops.⁴⁵ Figure 7 shows the results of a 12 s experiment. The experiment shows that the technique is also useful to analyze different small design options. Detailed data are shown in Table IV.

3.5. Pipeline FFT 256

In this example, the DUT is a block capable of doing the Fast Fourier Transformation, unidimensional with

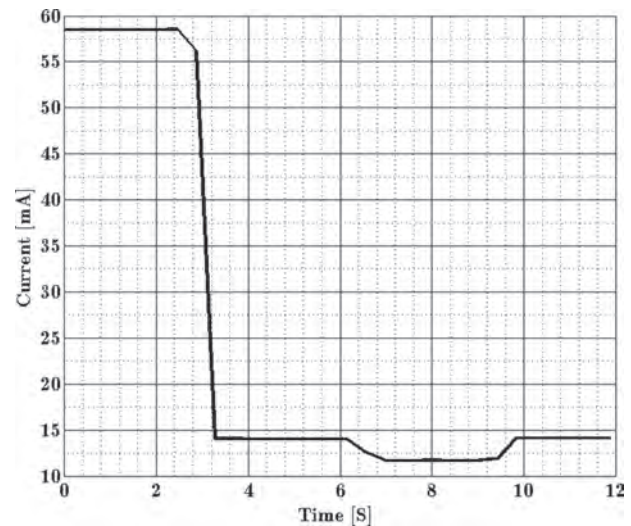


Fig. 7. Current consumption of the 32 bit multiplier with global clock gating and clock enable.

256 points. It has a standby mode: clock enable applied to all the flip-flops. Detailed data is shown in Table IV.

3.6. Physical 802.15.4

Here, the DUT is a block that implement the physical layer of the low-rate wireless personal area networks. It has a standby mode: clock enable applied to all the flip-flops. Detailed data is shown in Table IV.

3.7. AES

In the final example, the DUT is a high throughput and low area AES core that implements the Rijndael algorithm of encryption used in the AES standard. It has a standby mode: clock enable applied to all the flip-flops. Detailed data is shown in Table IV.

We showed that the proposed system can be used to measure the power consumption of a wide range of type of circuits with different consumptions. In all the cases the proposed system has an error less than 1% in comparison with the direct voltage drop measure through a shunt resistor.

4. CONCLUSIONS

A specific methodology to measure power consumption in FPGAs has been presented. The main characteristic of the tool is that it allows self-measurements; thus, an application configured inside FPGA can know-in real time-its own consumption. Simple and low-cost components, which add extra consumption of less than 1 mA, have been employed. The associated internal core uses very few resources inside the FPGA. The error is below 1% and it can be improved by using precision resistors.

The tool has multiple applications and allows the designer to test or add different low-power strategies to FPGA implementations. It is a powerful tool, as it can

be commanded by internal complex trigger conditions to analyze power of a circuit in a particular state, routine, or processing mode. Another interesting application is the self-adaptation option of an FPGA-based circuit, depending on its instantaneous power consumption.

References

1. G. J. García, C. A. Jara, J. Pomares, A. Alabdo, L. M. Poggi, and F. Torres, A survey on fpga-based sensor systems: Towards intelligent and reconfigurable low-power sensors for computer vision, control and signal processing. *Sensors* 14, 6247 (2014).
2. A. De la Piedra, A. Braeken, and A. Touhafi, Sensor systems based on FPGAs and their applications: A survey. *Sensors* 12, 12235 (2012).
3. J. Portilla, A. de Castro, E. de la Torre, and T. Riesgo, A modular architecture for nodes in wireless sensor networks. *Journal of Universal Computer Science* 12, 328 (2006).
4. J. Valverde, A. Otero, M. Lopez, J. Portilla, E. de la Torre, and T. Riesgo, Using SRAM based FPGAs for power-aware high performance wireless sensor networks. *Sensors (Basel, Switzerland)* 12, 2667 (2012).
5. M. Lombardo, J. Camarero, J. Valverde, J. Portilla, E. de la Torre, and T. Riesgo, Power management techniques in an FPGA-based WSN node for high performance applications, *7th International Workshop on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC)*, IEEE, July (2012), pp. 1–8.
6. Y. E. Krasteva, J. Portilla, E. de la Torre, and T. Riesgo, Embedded runtime reconfigurable nodes for wireless sensor networks applications. *IEEE Sensors Journal* 11, 1800 (2011).
7. J.-G. Tong, Z.-X. Zhang, Q.-L. Sun, and Z.-Q. Chen, Design of wireless sensor network node with hyperchaos encryption based on FPGA, *2009 International Workshop on Chaos-Fractals Theories and Applications*, IEEE, November (2009), pp. 190–194.
8. G. Chalivendra, R. Srinivasan, and N. Murthy, FPGA based reconfigurable wireless sensor network protocol, *IEEE International Conference on Electronic Design*, IEEE, December (2008), pp. 1–4.
9. P. Muralidhar and C. R. Rao, Reconfigurable wireless sensor network node based on nios core, *2008 Fourth International Conference on Wireless Communication and Sensor Networks*, IEEE, December (2008), pp. 67–72.
10. Y. Sun, L. Li, and H. Luo, Design of FPGA-based multimedia node for WSN, *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing*, IEEE, September (2011), pp. 1–5.
11. C. H. Zhiyong, L. Y. Pan, Z. Zeng, and M. Q.-H. Meng, A novel FPGA-based wireless vision sensor node, *2009 IEEE International Conference on Automation and Logistics*, IEEE, August (2009), pp. 841–846.
12. F. Philipp and M. Glesner, Mechanisms and architecture for the dynamic reconfiguration of an advanced wireless sensor node, *2011 21st International Conference on Field Programmable Logic and Applications*, IEEE, September (2011), pp. 396–398.
13. R. Garcia, A. Gordon-Ross, and A. D. George, Exploiting partially reconfigurable FPGAs for situation-based reconfiguration in wireless sensor networks, *17th IEEE Symposium on Field Programmable Custom Computing Machines*, IEEE, April (2009), pp. 243–246.
14. J. Oliver, J. Perez Acle, and E. Boemo, Power estimations versus power measurements in spartan devices, *2014 IX Southern Conference on Programmable Logic (SPL)*, Buenos Aires (2014).
15. Xilinx Announces XPower Power Analysis Software for FPGA Design (2000).
16. K. K. W. Poon, A. Yan, and S. J. E. Wilton, A flexible power model for FPGAs, *Proceedings of the Reconfigurable Computing is Going Mainstream, 12th International Conference on Field-Programmable Logic and Applications, FPL'02*, September (2002), pp. 312–321.
17. Altera's Quartus II Version 4.2 Delivers FPGA and CPLD Performance Leadership (2004).
18. E. I. Boemo, G. González de Rivera, S. López-Buedo, and J. M. Meneses, Some notes on power management on FPGA-based systems, *Proceedings of the 5th International Workshop on Field-Programmable Logic and Applications (FPL'95)*, Springer, Berlin, Heidelberg (1995), pp. 149–157.
19. E. Todorovich, G. Sutter, N. Acosta, E. Boemo, and S. Lopez-Buedo, End-user low-power alternatives at topological and physical levels. Some examples on FPGAs, *Proc. DCIS'2000*, Montpellier, France (2000).
20. L. Shang, A. S. Kaviani, and K. Bathala, Dynamic power consumption in Virtex-II FPGA family, *Proceedings of the 2002 Tenth ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA'02)*, Monterey, CA, United States (2002), pp. 157–164.
21. J. Becker, M. Huebner, and M. Ullmann, Power estimation and power measurement of Xilinx Virtex FPGAs: Trade-offs and limitations, *16th Symposium on Integrated Circuits and Systems Design (SBCCI)*, IEEE (2003), pp. 283–288.
22. Altera Corporation, Stratix II versus Virtex-4 Power Comparison and Estimation Accuracy (2005).
23. V. Degalahal and T. Tuan, Methodology for high level estimation of FPGA power consumption, *Proceedings of the Asia and South Pacific Design Automation Conference*, New York, NY, USA, ACM, January (2005), Vol. 1, pp. 657–660.
24. D. Meintanis and I. Papaefstathiou, Power consumption estimations versus measurements for FPGA based security cores, *2008 International Conference on Reconfigurable Computing and FPGAs*, December (2008), pp. 433–437.
25. D. Meidanis, K. Georgopoulos, and I. Papaefstathiou, FPGA power consumption measurements and estimations under different implementation parameters, *2011 International Conference on Field-Programmable Technology (FPT)* (2011), pp. 1–6.
26. H. G. Lee, K. Lee, Y. Choi, and N. Chang, Cycle-accurate energy measurement and characterization of FPGAs. *Analog Integrated Circuits and Signal Processing* 42, 239 (2005).
27. R. Jevtic and C. Carreras, Power estimation of embedded multiplier blocks in FPGAs. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 18, 835 (2010).
28. S. M. Afifi, F. Verdier, and C. Belleudy, Power estimation method based on real measurements for processor-based designs on FPGA, *Computational Science and Computational Intelligence (CSCI)* 2, 260 (2014).
29. B. Jovanovic, R. Jevtic, and C. Carreras, Binary division power models for high-level power estimation of FPGA-based DSP circuits. *IEEE Transactions on Industrial Informatics* 10, 393 (2014).
30. Z. Nakutis, Embedded systems power consumption measurement methods overview. *MATAVIMAI* 2, 29 (2009).
31. A. Borovyi, V. Kochan, A. Sachenko, V. Konstantakos, and V. Yaskilka, Analysis of circuits for measurement of energy of processing units, *4th IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, IEEE, September (2007), pp. 42–46.
32. R. Jevtic and C. Carreras, Power measurement methodology for FPGA devices. *IEEE Transactions on Instrumentation and Measurement* 60, 237 (2011).
33. F. Wolf, J. Kruse, and R. Ernst, Timing and power measurement in static software analysis. *Microelectronics Journal* 33, 91 (2002).
34. I. Haratcherev, G. Halkes, T. Parker, O. Visser, and K. Langendoen, Powerbench: A scalable testbed infrastructure for benchmarking power consumption, *Int. Workshop on Sensor Network Engineering (IWSNE)* (2008), pp. 37–44.
35. X. Jiang, P. Dutta, D. Culler, and I. Stoica, Micro power meter for energy monitoring of wireless sensor networks at scale, *6th International Symposium on Information Processing in Sensor Networks*, IEEE, April (2007), pp. 186–195.

36. T. Laopoulos, P. Neofotistos, C. Kosmatopoulos, and S. Nikolaidis, Measurement of current variations for the estimation of software-related power consumption. *IEEE Transactions on Instrumentation and Measurement* 52, 1206 (2003).
37. V. Konstantakos, K. Kosmatopoulos, S. Nikolaidis, and T. Laopoulos, Measurement of power consumption in digital systems. *IEEE Transactions on Instrumentation and Measurement* 55, 1662 (2006).
38. N. Chang, K. Kim, and H. G. Lee, Cycle-accurate energy measurement and characterization with a case study of the ARM7TDMI [microprocessors]. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 10, 146 (2002).
39. J. Andersen and M. T. Hansen, Energy bucket: A tool for power profiling and debugging of sensor nodes, *Third International Conference on Sensor Technologies and Applications*, IEEE, June (2009), pp. 132–138.
40. Z. Nakutis, A current consumption measurement approach for FPGA-based embedded systems. *IEEE Transactions on Instrumentation and Measurement* 62, 1130 (2013).
41. C. Fernandez, D. Bouvier, J. Villaverde, L. Steinfeld, and J. Oreggioni, Low-power self-energy meter for wireless sensor network, *IEEE International Conference on Distributed Computing in Sensor Systems*, IEEE, May (2013), pp. 315–317.
42. J. Villaverde, L. Steinfeld, J. Oreggioni, D. Bouvier, and C. Fernández, Self-energy meter in duty-cycle battery operated sensor nodes, *Instrumentation and Measurement Technology Conference (I2MTC)* (2014).
43. CSS555C Micropower Timer (with Internal Timing Capacitor), http://www.customsiliconsolutions.com/downloads/Revised_Standard_products/CSS555C_Spec_2.pdf (2012).
44. O. Girard, openMSP430, <http://opencores.org/project.openmsp430/> (2009).
45. J. P. Oliver, J. Curto, D. Bouvier, M. Ramos, and E. Boemo, Clock gating and clock enable for FPGA power reduction, *2012 VIII Southern Conference on Programmable Logic*, IEEE, March (2012), pp. 1–5.

Juan P. Oliver

Juan P. Oliver received a Ph.D. degree in electrical engineering from Universidad de la República, Uruguay. He is a full time Associate Professor in the Electrical Engineering Department at Universidad de la República, Uruguay. His research interests include the design of FPGA-based systems, low-power techniques, embedded systems, and electrical engineering education.

Francisco Veirano

Francisco Veirano received the Electrical Engineering degree from Universidad de la República, Uruguay in 2013. He joined the Electrical Engineering Department of Universidad de la República, Uruguay in 2012 where he is currently working as Research Assistant. Since 2013, he is a Ph.D. student from the same department. His research interests include ultra low-power analog and digital integrated circuits design, in particular subthreshold digital circuits and DC–DC converters.

Diego Bouvier

Diego Bouvier received the Electrical Engineering degree from Universidad de la República, Uruguay in 2015. He was a R&T Assistant with the Electronics Department at the Facultad de Ingeniería, Universidad de la República. Since 2000 works in ANTEL, a state telecommunications company.

Eduardo Boemo

Eduardo Boemo received the Electrical Engineering degree from the Universidad Nacional de Mar del Plata, Argentine, and the Ph.D. degree in Telecommunication Engineering from the Universidad Politécnica de Madrid, Spain, in 1985 and 1996, respectively. Currently, he is Titular Professor at the School of Computer Engineering, Universidad Autónoma de Madrid, Spain. His current research interests include the design of FPGA-based systems, low-power techniques, computer arithmetics, self-timed circuits, and electrical engineering education.