

# Power Measurement Methodology for FPGA devices

Ruzica Jevtic and Carlos Carreras

## Abstract—

The efficiency of power optimization tools depends on information on design power provided by the power estimation models. Power models targeting different power groups can enable fast identification of the most power consuming parts of design and their properties. The accuracy of these estimation models is highly dependent on the accuracy of the method used for their characterization. The highest precision is achieved by using physical on-board measurements. In this work, we present a measurement methodology that is primarily aimed at calibrating and validating high-level dynamic power estimation models. The measurements have been carefully designed to enable the separation of the interconnect power from the logic power and the power of the clock circuitry, so that each of these power groups can be used for the corresponding model validation. The standard measurement uncertainty is lower than 2% of the measured value even with a very small number of repeated measurements. Additionally, the accuracy of a commercial low-level power estimation tool has been also assessed for comparison purposes. The results indicate that, the tool is not suitable for power estimation of data path oriented designs.

*Index Terms*—FPGA, power, measurements, high-level.

## I. INTRODUCTION

POWER estimation models serve to accelerate power optimization process by providing power estimates without a need of implementing the design and measuring its power.

The following equation is used for estimating the dynamic power of a gate or a connection line:

$$P = \alpha \cdot C_l \cdot V_{dd}^2 \cdot f \quad (1)$$

where  $\alpha$  (referred to as switching activity) is the average number of  $0 \rightarrow 1$  transitions in one clock-cycle,  $C_l$  is the load capacitance,  $V_{dd}$  is the power supply voltage, and  $f$  is the clock frequency.

The value of the power supply is usually fixed and constant, and the clock frequency is defined for each specific design. The switching activity can be determined from design simulations. Therefore, only the load capacitance remains unknown for power estimation. According to the features of this parameter, dynamic power can be divided into three components: the power of the clock circuitry (with dedicated routing resources), the logic power consumed in the functional units and memories (where the load capacitances correspond to the loads driven by the outputs of the logic gates), and the power of the interconnects between the units (where the load capacitance depends on the type and length of the wires).

In this work we consider the power estimation of Field-Programmable Gate Arrays (FPGAs). FPGAs and appli-

cation specific hardware (ASICs) are the most commonly used implementation mediums that can achieve circuits with high processing rates. In particular, FPGAs have become an attractive solution for various embedded designs due to their ability for reconfiguration and significantly lower cost compared to ASICs.

Since FPGAs are configured at the transistor level, power estimation is based on a gate-level approach rather than the instruction-level approach that is often used in microprocessors ([1], [2], [3]). FPGAs do not have a fixed structure as microprocessors do (processor, memory and signal bus), as the design architecture depends on each application. This indicates a need for different power measurement techniques aimed at measuring the power of logic components and the connections between them.

FPGAs are available only in a closed form to the users, (i.e. in a device package). This means that their electrical structure is hidden from the outside world. The only way to separate the power of different elements inside the chip is to know their capacitances. There are two different ways for obtaining these values: from the low-level tools provided by the chip vendors, or through a methodology based on on-board power measurements.

There are a few low-level tools designed for commercial FPGAs, and the most widely used are XPower from Xilinx [4] and PowerPlay from Altera [5]. These tools provide a detailed power breakdown of a design based on the resource capacitance, resource utilization and data switching activity [6]. However, a significant difference has been reported when the estimates obtained from these tools have been compared to physical measurements [7], [8]. Additional problems are encountered when complex designs with many signals are to be modelled, as these tools require large amounts of memory and long execution times. As a result, it is preferred that the power estimation models are characterized by on-board measurements.

In this work, we present a measurement system that is designed for FPGAs in order to facilitate the separation of the static power, the clock power, the power of the global interconnects and the power consumed in the logic. The separation of different power groups is important since it allows for the optimization techniques to localize the most consuming parts of the design, determine the nature of their power (whether they belong to the logic, clock or interconnect group) and apply the corresponding optimization steps. The methodology is adapted to the special features of FPGAs such as the existence of several different types of wires, programmable switch matrices used for establishing connections between the wires, limited number of routing resources, etc. These features lead to a more difficult power separation when compared to the power mea-

R. Jevtic and C. Carreras are with the Department of Electrical Engineering, ETSI Telecomunicacion, Ciudad Universitaria s/n, 28040 Madrid, Spain e-mail: ({ruzica, carreras}@die.upm.es)

measurements in ASICs. The power of interconnects is more accessible in ASICs as it can be directly related to the wire length and the number of routing resources is unlimited.

The Xilinx Virtex II Pro device XC2VP30 that first appeared on the market in 2002 has been selected as target platform. The methodology presented here can be easily extended in order to consider the most recently released high-speed FPGA devices, as their structure is built upon the Virtex II Pro device architecture.

The main features of this work are the following:

- The measurement system is designed so as to eliminate input vector generation power.
- A tool developed in C++ is capable of extracting the exact number and type of the wires used for design interconnections from design files.
- The effective wire capacitances are obtained through measuring power of simple designs in many scenarios.
- The measurement system provides precise measured power values. The standard measurement uncertainty is found to be below 2% of the measured value.
- A thorough analysis is performed in order to explore the accuracy of XPower over all different power groups. Although the tool is more user-friendly, it lacks the accuracy required for model validation.

The paper is organized as follows. Section II highlights the previous work on physical power measurements. Section III describes the measurement methodology, followed by its implementation in Section IV. Experimental results are presented in Section V and conclusions in Section VI.

## II. RELATED WORK

On-board power measurements of FPGAs have been used for many different purposes, from the analysis of power distribution over different elements [6], [9], to the influence of the design architecture features on power [8], [10], and the characterization and verification of the power estimation models [7], [11].

Dynamic power consumption is analyzed in Virtex II [9] and Virtex 4 [6] devices by extracting the effective capacitance of all the resources through simulations and measurements. The aim of the work in [9] is to better understand where power is consumed in FPGAs, while the work in [6] presents a methodology for pre-silicon dynamic power estimation of FPGA-based designs. These approaches have some similarity to the approach presented here regarding the extraction of the effective wire capacitances. However, as they have access to the proprietary information on the chip layout, they rely on transistor level simulations to obtain the effective capacitances of all the resources. On the other hand, the approach presented here is available to any user as the power values are measured directly. Also, the input vectors are loaded from another board. Furthermore, the description of the tool used to extract resource usage after place-and-route is confidential in [6], [9], whereas a detailed description of our tool is included in Section IV-C.

A high-level power estimation model of Xilinx FPGA embedded memories is presented in [7], [11]. The model uses a set of architectural and algorithmic parameters,

where the coefficients standing by the parameters are obtained through curve fitting over measured power values. Significantly, a maximum error of 132% is reported in the estimates provided by XPower, for the implementation of a FIR filter in Virtex II Pro and Virtex E devices.

The work in [10] analyzes the impact of pipelining on power consumption in both Xilinx and Altera FPGAs by varying the number of pipeline stages and detecting the power difference. Still, as they measure the total board consumption, they are not able to isolate the dynamic consumption of the FPGA in order to guide other architectural decisions, apart from the number of pipeline stages.

Cycle-by-cycle energy measurement in FPGAs is presented in [8]. The measurement is based on switched capacitors, which allow determining the static and dynamic energy per cycle. Additionally, the authors compute the average power value and report high overestimation errors when these values are compared to XPower estimates. Still, the logic and interconnect power components cannot be separated from the whole FPGA core consumption.

Some accurate measurement methods have been proposed for microprocessor systems [1], [2], [3]. A current mirror circuit with bipolar transistors is used for measuring instantaneous current drawn by the processor in [2], [3]. The technology is primarily applied to small microprocessors. The work in [1] presents mathematical criteria to keep the measurement uncertainty associated to software-related current drain measurements below a given target value. The results point out that the accuracy of the measurements can be improved by choosing an integration time much longer than the waveform period.

Unlike any other reported previous work, the work presented here is used for separate validation of different power groups in FPGA high-level power estimation models. The measurement methodology is used to determine the average dynamic power value. It separates the interconnect power from the rest of the dynamic power by using a methodology similar to [6], [9], and also the logic power from the other power components by loading the input vectors from other board, as in [7], [11].

## III. MEASUREMENT METHODOLOGY

Due to different load capacitance characteristics of interconnections and logic, their power estimation is achieved through different models [12], [13], [14]. Consequently, the models need to be characterized separately, so separate logic and interconnect measured power values are required. The chips are enclosed and the power supply of different design elements cannot be accessed separately. Hence, the total power of the chip has to be measured and then, the different power components can be separated by carefully designing the circuits to be measured and by post-processing the results afterwards.

Three steps are used to separate the power groups:

- The static and the clock power are obtained through measurements with different input vector sets.

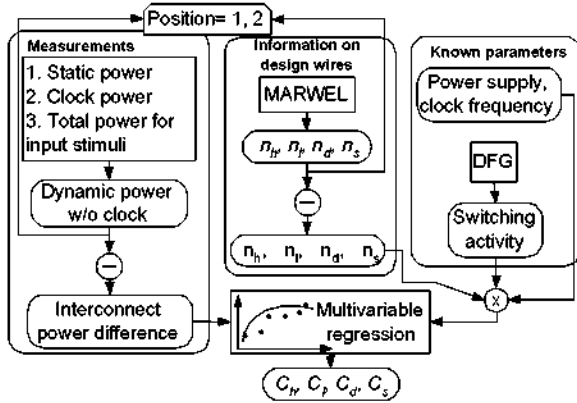


Fig. 1: Methodology for effective capacitance extraction

- The interconnect power is obtained after computing the effective wire capacitances through power measurements. The measurements are performed for circuits specially designed for this purpose.
- The logic power is obtained by subtracting the other three power components from the total power.

The complete methodology is presented in Fig. 1. In the following, each of the steps is described in more detail.

#### A. Power measurements

The circuits used in our measurements contain multiplier or adder components. The components are replicated between one and four times on the board, as to improve the accuracy of the measurements. Hence, each design consists of several identical modules and the lines that connect the module's pins to the I/O pins (see Fig. 2). This facilitates the separation of power components as it is explained later.

First, we measure the static power of the designs when no input vectors and no clock signal is injected to the board. It is known that the static power varies with the state of logic signals during design operation, and also with the way a design utilizes the FPGA hardware. The activity of the logic signals increases the chip temperature, which in turn, increases the static power as well. However, the designs we have used are extremely small, so it is assumed that the static power increase would be negligible.

Second, we measure the clock power together with the static power when the design is stimulated only with the clock signal, while the inputs are set to '0'. As the circuits contain only synchronized combinatorial logic without any feedback loops, it can be considered that there is no toggling on logic signals when all the inputs are set to '0'. Hence, the clock power is also measured properly.

Finally, we measure the total power of the design by performing various measurements for sets of 10000 input signal vectors with gaussian distributions and different autocorrelation coefficients. The power of the clock circuitry together with the static power is subtracted from the total power, as to isolate the dynamic power of the logic and the interconnects for each input stimuli set.

In order to confirm the assumption that the static power is measured properly, we have repeated the three above mentioned steps at two different frequencies (50MHz and

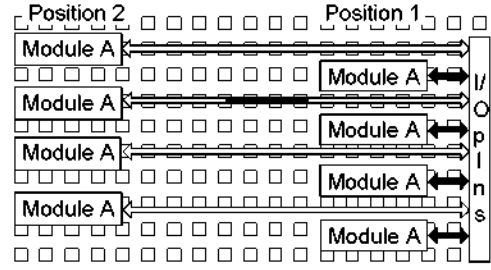


Fig. 2: Two different module positions in FPGA

100MHz) for several of the most power-consuming designs in the set (containing multipliers implemented in LUTs). Indeed, after subtracting the static power from the total design power, the relationship between the two isolated dynamic power values for each design corresponded to the relationship between these two frequencies (i.e. two).

The next step in the measurement procedure includes the separation of the power of a component from the power of the global routing. This is done by first, eliminating the logic power, and then, computing the effective capacitances of all the global routing resources through measurements.

First, we repeat the set of measurements (1 - 3 in Fig. 1) for two different positions of the modules on the chip: one where the modules are placed very close to the I/O pins, and the other where they are placed far from them (see Fig. 2). We use area constraints in order to accomplish the wanted module positions. By subtracting the two values obtained for the dynamic power consumption in the two positions, we are able to obtain the value of the power consumption of the interconnect difference between them.

It is important to note that the modules considered here have registered inputs and outputs. Inserting registers at the inputs and outputs is necessary in order to eliminate the glitching that might occur inside the module due to the different paths from the I/O pins. Thus, we ensure that, as a result of the subtraction, the module power is completely cancelled.

#### B. Wire capacitance extraction

In commercial FPGAs, routing is accomplished through a hierarchy of segmented routing resources in order to achieve high speed. The most power consuming are the long lines, followed by the hex and double lines, while the least consuming are the single lines [9], [15], [16].

We model the effective capacitance of each resource (long, hex, double and single) as the capacitance of the routing wire together with the programmable switch that drives the wire, as in [6]. In the continuation, the number of hex, long, double and single wires used for routing of each interconnect  $i$  that goes from or to IO pins in the design is marked as  $n_{hi}$ ,  $n_{li}$ ,  $n_{di}$  and  $n_{si}$ .

As the inputs and outputs are registered, there is no glitching in the wires that connect I/O pins with the inputs/outputs of the modules. So we are able to obtain the switching activities,  $sw_i$ , of the routing wires from simple data flow graph simulations. The value of the switching activity for each interconnect is then multiplied by the corresponding number of each wire type used for its routing.

According to (1), we need four parameters in order to calculate the power of the interconnects. Two of them are known, as the power supply has a value of 1.5V for Virtex-II Pro devices, and the clock frequency is fixed to the value used in our measurements (50MHz or 100MHz).

As mentioned before, by subtracting the obtained dynamic power for two different module positions, we eliminate the logic power and obtain the interconnect power difference. By using equation (1), we can express the power difference of a design in the two measured positions as:

$$\begin{aligned}
P_1 - P_2 = & V_{dd}^2 \cdot f \cdot (C_h \cdot \sum_{i=1}^{I_1+I_2+O} [(n_{hi}^1 - n_{hi}^2) * sw_i] \\
& + C_l \cdot \sum_{i=1}^{I_1+I_2+O} [(n_{li}^1 - n_{li}^2) * sw_i] \\
& + C_d \cdot \sum_{i=1}^{I_1+I_2+O} [(n_{di}^1 - n_{di}^2) * sw_i] \\
& + C_s \cdot \sum_{i=1}^{I_1+I_2+O} [(n_{si}^1 - n_{si}^2) * sw_i])
\end{aligned} \quad (2)$$

where  $P_1$  and  $P_2$  are the measured dynamic power values of the design with the modules in the positions far from and near to the I/O pins respectively,  $C_h$ ,  $C_l$ ,  $C_d$ ,  $C_s$  are the variables representing the effective capacitance of the hex, long, double and single wires respectively,  $I_1$ ,  $I_2$  are the word-lengths of the two input operands and  $O$  is the word-length of the output. The design position is identified through superscripts 1 (far) and 2 (near). We have two sets of unknown variables: the number of different wire types used for routing each interconnect in the design ( $n_{hi}$ ,  $n_{li}$ ,  $n_{di}$ ,  $n_{si}$ ) and the effective wire capacitances ( $C_h$ ,  $C_l$ ,  $C_d$ ,  $C_s$ ). The first set is obtained by extracting routing information from the design files. In particular, we use MARWEL, C++ tool specially designed for this purpose. It is described in more detail in the next section. Then, a multivariable regression over a number of measurements for modules with various operand word-lengths is applied, as to obtain the effective capacitance for all types of wires.

Once we have these values, we can obtain the power consumption of any interconnect by using the information about the number of different wire types used for its routing. Hence, the power of a single interconnect is:

$$P = V_{dd}^2 \cdot f \cdot sw \cdot (n_h \cdot C_h + n_l \cdot C_l + n_d \cdot C_d + n_s \cdot C_s) \quad (3)$$

Finally, the total interconnect power is obtained by summing the power of all global interconnects in the design.

### C. Logic and Input buffer power

Input buffers are also power supplied by the FPGA core voltage. Hence, the remaining power obtained by subtracting the interconnect power from the design dynamic power contains two power components: the module power and the input buffer power. In order to obtain the module power, we need to compute the effective capacitance of the input buffers as well. This capacitance is computed by measuring the power of two designs: one containing three multipliers implemented in LUTs, and the other containing only one multiplier implemented in LUTs. First, we

subtract the corresponding interconnect power from each of the designs. Thus, we obtain the following logic power values:

$$P_{log,1} = 3 * P_{mult} + P_{in\_buf} \quad (4)$$

$$P_{log,2} = P_{mult} + P_{in\_buf} \quad (5)$$

where  $P_{log,1}$  and  $P_{log,2}$  are the logic power values of the first and the second design, respectively,  $P_{mult}$  is the logic power of the multiplier and  $P_{in\_buf}$  is the power of the input buffers. From these two equations we are able to extract the power of the input buffers. The effective capacitance of a single input buffer is then obtained by dividing  $P_{in\_buf}$  by the sum of the switching activities of the inputs, the square of the power supply and the design frequency:

$$C_{in\_buf} = \frac{P_{in\_buf}}{V_{dd}^2 \cdot f \cdot \sum_{i=1}^{N_{in\_buf}} sw_{in_i}} \quad (6)$$

where  $N_{in\_buf}$  is the total number of inputs, and  $sw_{in_i}$  is the switching activity of the  $i$ -th input. As a result, the measured effective capacitance of a single input buffer is found to be 3.52 pF for Virtex II Pro devices.

The module power can now be easily obtained by subtracting the interconnect power and the power of the input buffers from the total dynamic power of the design.

## IV. MEASUREMENT IMPLEMENTATION

### A. Measurement setup

The measurement setup is presented in Fig. 3a. The development of the measurement setup was inspired by the work presented in [7] and was also partially described in [13]. The system contains two FPGA boards: a XUP board from Xilinx and a Stratix DSP Development board from Altera (see Fig. 3a). The board from Altera is used for loading the simulation vectors to the XUP board. The XUP board serves for measuring the power of a specific design. As the power supplies for the core, I/O pins and auxiliary power supply are separated on the XUP board, we measure directly only the core power of the FPGA. The 1.5V power supply for the core voltage is provided by a synchronous buck-switching regulator connected to the 4.5V-5.5V external power input [17]. A simplified diagram is provided in Fig. 3b.

We use a resistance at the entrance of the core power supply to the chip, and for each test design, we measure the voltage over this resistor. This enables us to calculate the current provided by the supply. The resistance value is chosen so as to ensure the correct functionality of the power supply regulator on the XUP board as it is explained next.

The 1.5V power supply for the core voltage is created by the synchronous buck-switching regulator. The regulator has a feedback loop in order to maintain a fixed value of the output voltage. The feedback controlling input to the regulator is taken directly from the core power supply pin on the XC2VP30 device and is marked as point B in Fig. 3a.

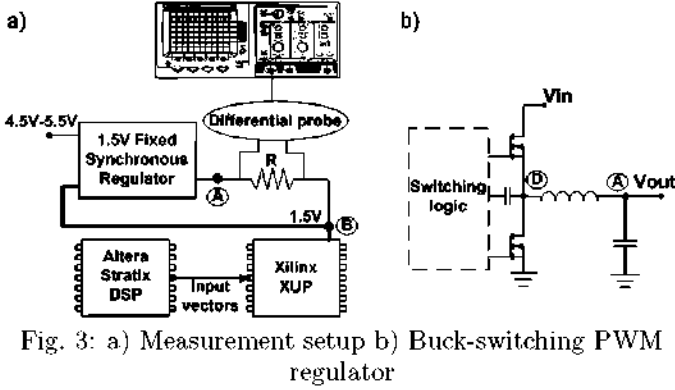


Fig. 3: a) Measurement setup b) Buck-switching PWM regulator

This connection is integrated on the XUP board and is marked with a thicker line in Fig. 3a. Therefore, the voltage at the input of the chip,  $V_B$ , is maintained at 1.5V meaning that the functionality of the chip itself is guaranteed. As feedback is obtained through point B in Fig. 3a, the voltage value at the output of the regulator (marked as A in both figures) has the value:

$$V_A = V_B + R * i \quad (7)$$

The regulator is buck-switching, so it is important to avoid the saturation of the internal coil of the regulator. The saturation will occur when the average voltage value at the output of the buck converter exceeds the value of the average voltage on the other coil end [18] (marked as D). The average voltage value at point D equals to:

$$V_D = d * V_{in} \quad (8)$$

where  $d$  is the duty cycle of the buck converter. Consequently, the saturation of the coil will not occur as long as voltage  $V_A$  is smaller or equal to the maximum voltage  $V_D$ . From equations (7) and (8), we obtain the condition that has to be fulfilled:

$$V_B + R * i \leq d_{max} * V_{in} \quad (9)$$

Since the value  $d_{max}$  is not provided in the regulator's data sheet, we have obtained it experimentally and it equals to 0.5. Therefore, we obtained that the average voltage over resistance  $V_D - V_A$  should not surpass the value of 1V. We have measured power using several different resistance values, starting from 1 ohm, in order to find the largest one that would fulfill the condition of the maximum voltage value. The circuits used in our measurements contained only one to four multiplier or adder components as the number of their replications was limited by the number of available I/O pins on the Xilinx board. The modules had operand sizes smaller than 17 bits (limitation due to the connection of the boards), connected directly to the I/O pins. Thus, their power consumption was always small enough to allow a 10 ohms resistance value. For larger designs, this value should be reduced in order to satisfy (9).

The voltage over the resistance is measured by using a differential probe Tektronix P6248. In order to account for the inherent probe offset we measure both, the voltage between A and B ( $V_{A \rightarrow B}$ ) and the voltage between B

and A ( $V_{B \rightarrow A}$ ). The measured value for each probe position is the average of 750.000 voltage values recorded in the oscilloscope (75 values for each of the 10000 loaded input vector pairs). The final voltage value is obtained as the mean of absolute voltage values:  $(|V_{A \rightarrow B}| + |V_{B \rightarrow A}|)/2$ . An additional signal is generated on the Altera board that indicates the beginning and the end of the loaded input vector sequence. The power is then obtained as the product of the power supply voltage and the average current going through the resistance.

As the designs are stimulated externally from the Altera board, they do not contain extra blocks like memory arrays, control logic etc., that would contribute to the total power. As a result, it is much easier to separate the module power from the global interconnect power.

### B. Measurement uncertainty

First, we consider the systematic error due to the resistance tolerance. The tolerance of the resistance is  $\pm 5\%$ . The real value, which is measured by using an Agilent multimeter 34410A, is 9.84 ohms. So we apply the corresponding correction factor according to the Guide to the Uncertainty in Measurements [19].

Next, we analyze the measurement uncertainty, by using the methodology presented in [1]. The standard uncertainty can be computed as:

$$u = \sqrt{u_{T1}^2 + u_{T2}^2} \quad (10)$$

where  $u_{T1}$  is type A standard uncertainty that is evaluated from a statistics of  $N$  uncorrelated measurements, and  $u_{T2}$  is type B standard uncertainty that is mainly due to the instrumental contributions and can be evaluated from instrumentation specifications. Type A uncertainty can be further divided into two independent contributions: the lack of coherence in sampling the voltage waveform and the superimposed wideband noise.

We will make the uncertainty analysis for the total measured power of a  $16 \times 16$  adder. The overall uncertainty has to be smaller than a threshold  $u_d$ . As a rule of thumb, it is recommended to take the value higher than  $u_{BM}$ , which represents the maximum type B standard uncertainty associated with a certain range of the oscilloscope [1] (in this case 1.5%). The following condition that is used to analyze the measurement uncertainty for the periodic current signal in [1], will also be used here:

$$u^2 \leq \frac{\sigma_a^2}{8N} \left( \frac{T_p}{T_o} \right)^2 + \frac{\sigma_r^2}{N \cdot B \cdot T_o} + u_{BM}^2 < u_d^2 \quad (11)$$

where  $T_p$  was the period of the current signal, and  $T_o$  was the time interval of the oscilloscope. As we measure the voltage when 10000 different vectors are applied to the modules inputs,  $T_p$  in our case corresponds to the total time duration of 10000 input vectors. On the other hand, since this time also corresponds to the time interval of the oscilloscope during which we record measured voltage values, it can be considered that  $T_p \approx T_o$ . The chosen frequency for the design is 50MHz, so  $T_o = \frac{10000}{50MHz} = 0.2ms$ .

Furthermore, we consider the worst case for the number of measurements and set the value  $N$  to one. The two-side equivalent noise bandwidth of the instrument is marked with  $B$  and is approximately twice the bandwidth reported in the instrument's specifications. The bandwidth of the differential probe is 400 MHz, resulting in  $B \approx 800\text{MHz}$ .

The lack of coherence in sampling the voltage waveform which occurs whenever the starting measure time of the oscilloscope is not synchronized with the beginning of the waveform is represented as the variance  $\sigma_a$ . We have repeated five different measurements, and they resulted in  $\sigma_a = 3.65 \cdot 10^{-4}$ . We can see that this variation is very small. This is due to a specific signal that marks the beginning and the end of the input sequence and to the fact that the measurements are processed afterwards by software. Hence, only the measured voltage values after the rise and before the fall edge of the specific signal are accounted for. Consequently, the measurement time interval is synchronized with this signal, meaning that the sampling of the voltage waveform is coherent for each repeated measurement. As a result, the uncertainty due to this effect is driven to its minimum.

The wideband noise superimposed on the voltage values is represented as variance  $\sigma_r$ . It is assumed to have Gaussian distribution and thus, it is usually obtained as a 1/6 part of the peak-to-peak amplitude of the measured signal. As we measure the voltage over the resistance twice, one for the positive polarity of the voltage and the other for its inverse polarity, for each recorded voltage point  $i$  we compute the average absolute value of these two voltages,  $av_i$ . Then, the peak-to-peak amplitude  $r_{pp}$  is equal to  $\max(av_i) - \min(av_i)$  which leads to  $\sigma_r = 0.0424$ .

By using inequality (11) we obtain that the measurement uncertainty equals to 1.5% and is completely determined by the last term representing the instrument uncertainty. The first two terms are several of orders of magnitude smaller and can be neglected.

### C. MARWEL tool

MARWEL (Measurement of ARchitectural WirE Lengths) is a C++ tool designed to extract the exact number and type of the wires used for design interconnections. After placement and routing of a design, the Xilinx synthesis tool ISE creates a native circuit description file (.ncd) which represents the physical circuit description of the input design [4]. XDL file is the text version of the placed and routed circuit description (.ncd) and is created by the Xilinx Design Language (XDL) tool. First, we will give an overview of the .xdl file structure, as this information is essential for MARWEL. Then, the structure of MARWEL will be described.

The .xdl file is obtained through the `ncd2xdl` command of the Xilinx ISE framework. It consists of two parts. In the first part, there is a list of all the design instances together with their configuration and location on the FPGA board. Instances belong to one of the following groups: logic blocks, I/O pins, DCMs, and multiplexers, and their description begins with the word "inst" (see Fig. 4).

```

Instance   Logic   Instance CLB   Instance slice
name       or I/O  position      position
inst "m/N1479" "SLICE",placed R42C41 SLICE_X81Y77
cfg " BXINV:::#OFF BXOUTUSED:::#OFF BYINV:::#OFF
BYINVOUTUSED:::#OFF BYOUTUSED:::#OFF...
```

Fig. 4: XDL file syntax: part I

		Line number
Net name	→ net "operand1<1>",	1
Net end	→ inpin "m/N214" BX,	2
Net beginning	→ outpin "input1<1>" IQ1,	3
Switch matrix	→ pip R10IR38 L_Q10 -> OMUX5,	4
	pip R39C46 OMUX_SW5 -> S6BEG9,	5
CLB position	→ pip R45C46 S6END9 -> W2BEG8,	6
	pip BRAMR44C8 W2END_N8 -> W2BEG0,	7
	...	

Fig. 5: XDL file syntax: part II

The basic instance description is followed by its configuration details. Since MARWEL uses only the first line of each instance description, most of the configuration data is irrelevant for the extraction of design routing properties.

The second part of the XDL file contains a list of all the nets in the design. An example of a net's description is given in Fig. 5. A net always begins with the word "net", followed by the net's name. Next, the name of the pin where the net begins and the names of the pins where it ends are listed. These names correspond to some of the instance names given in the first part of the .xdl file.

The identifier "pip" is used to describe a connection inside a switch matrix. It is followed by the position of the switch matrix. Finally, a description of the wires that are connected inside that particular matrix is provided. The positions of the switch matrices as well as the wire description are essential information for extracting design routing properties. Hence, we give a more detailed explanation.

An FPGA is an array of Configurable Logic Blocks (CLBs), where each CLB position is defined by its row and column numbers. For example, the CLB position marked in Fig. 5, begins with the letter *R*, followed by a number which represents the row coordinate. A similar notation is used to express the column coordinate.

The four types of global wires are described as follows:

- Direct line: it starts with a notation *OMUX* which is then followed by a track number and/or a direction. For example, line 4 in Fig. 5 marks the beginning of the direct line in track 5, and in line 5 this direct line ends with a direction south-west (*SW*).
- Double line: it starts with a direction, followed by a number 2 (for double), the abbreviation *BEG* (begin), *MID* (middle), *END* (end) which mark the current wire part, and a track number. For example, the end of line 6 in Fig. 5 marks the beginning of a double line in track 8 that has a direction towards west (*W*).
- Hex line: it has the same notation as the double line, except for the number 6 after the direction.
- Long line: there are two notations for a long line: *LV* for vertical or *LH* for horizontal direction.

MARWEL represents nets as graphs, where nodes corre-

spond to CLBs and edges correspond to wires connecting two CLBs. Functions provided in the Graph Template Library [20] are used in order to describe the design nets as graphs. This facilitates the circuit description and the search algorithms applied in order to find the specific design information.

MARWEL operates in three stages. First, it parses the first part of the .xdl file and gathers the information about the names and the positions of the logic and I/O pins of the design. It separates the list containing all I/O pins from the list that contains logic blocks. This is necessary for the purpose of the work presented here, since we need to identify the connections that go to or from I/O pins separately from the local connections between the CLBs inside an arithmetic component.

Second, each net is transformed into a graph where nodes represent switch matrices and edges represent wires. There is no available published documentation on the XDL tool, so this task is extremely difficult. Failing to identify only one connection, leads to an unfinished graph, as each wire has a single particular predecessor.

Finally, the third stage applies a large number of functions designed for user purposes, including:

- Net functions: for each net there are functions that can compute the total number of all wires, hex wires, double wires, direct wires, long wires, local wires inside a CLB, and identify the switch matrices used for the routing of a net.
- Path finder functions: these functions can find how many routing resources and of which type have been used for routing a part of a multi-terminal net between two specified logic blocks, as well as between a logic block and an I/O pin.
- Clock functions: a clock net is routed via special-purpose wires, so some special functions are included for analysing the routing properties of these nets.

## V. EXPERIMENTAL RESULTS

### A. Effective wire capacitances

In this section, first we give the values obtained for the wire capacitances. They are followed by the errors obtained when the measured power difference is compared to the estimated interconnect power difference. Interconnect power difference is computed by using wire capacitance values and the information obtained from MARWEL.

In order to ensure correct values, some measurements were repeated several times under different temperature conditions in the laboratory. The static power changed as a function of the alteration of the temperature.

The experiments were performed on four different size multipliers implemented in LUTs, four different size embedded multipliers and five different size adders with operand sizes of 8, 12 and 16 bits. The module input signals had a zero-mean gaussian distribution with autocorrelation coefficients that varied between 0 and 0.9995. The characterization set used for the multivariable regression considered the power values corresponding to the input signals with autocorrelation coefficient equal to 0 (i.e. sw-

Table I: Effective capacitances for different wire types.

Wire type	Long	Hex	Double	Direct
Capacitance per unit-length [fF]	182.2	88.1	73.2	$\approx 0$

Table II: Error for the interconnect power computed with the effective capacitance values.

Module types	Error [%]				
	$\rho = 0$	$\rho = 0.5$	$\rho = 0.9$	$\rho = 0.99$	$\rho = 0.9995$
mult16x16	4.23	-6.05	10.56	-2.01	-0.81
mult12x12	9.75	6.16	11.10	1.61	6.05
mult8x8	-1.04	-0.34	-0.63	-11.72	-1.81
mult12x8	-3.66	-9.41	-4.18	-11.13	-8.92
emb12x12	7.61	-3.34	-4.42	-12.48	-9.19
emb16x12	15.95	10.06	7.4	6.99	6.41
emb16x8	-6.86	-6.44	-1.4	-6.95	-6.62
emb12x8	-2.45	-12.65	-0.16	-7.96	-4.63
add16x16	-2.02	1.01	1.38	-0.06	-2.67
add12x12	3.16	-0.24	3.66	-7.01	-2.01
add8x8	6.67	5.42	0.96	2.23	11.23
add16x8	4.93	4.75	5.33	3.46	10.96
add12x8	-4.87	-3.21	2.85	2.82	4.63

itching activity of 0.5), as they provided the largest consumption and thus, the best accuracy. Additionally, we have also used the power value of the  $16 \times 16$  multiplier and the  $16 \times 16$  adder with autocorrelation coefficients of 0.5, 0.9 and 0.99 as they are the largest components used for the measurements, and thus, have the largest consumption. Although an adder consumes less power than a multiplier, we have replicated each adder core three times in order to improve the measurement precision.

The measured capacitance values for the different wire types are given in Table I. These values correspond to the capacitances spanning the distance between two neighbouring CLBs. The total wire capacitance for each wire type is obtained when the corresponding capacitance per unit-length is multiplied by the number of wire segments.

However, the presented capacitance model does not take into account wire parasitics. For example, the number of possible connections between different lines inside the switch matrix is quite limited. Hence, many interconnections pass through multiple switches of the same switch matrix before reaching the connection towards the desired line. As a result, the wire capacitance increases. In order to evaluate the impact of wire parasitics, we obtain two values for each module and each autocorrelation coefficient:  $\delta P$ , that corresponds to the power difference between the module positions 1 and 2, in the left-hand side of equation (2), and  $P_{cap}$ , that corresponds to the right-hand side of the same equation, computed from the obtained effective capacitance values. Table II shows the relative errors when the computed  $P_{cap}$  is compared to the measured  $\delta P$ .

It can be observed that the resulting discrepancy is 5.36% on average and always smaller than 15.95%. Thus, it is confirmed that capacitance model is accurate enough to be used for purposes of validation and characterization

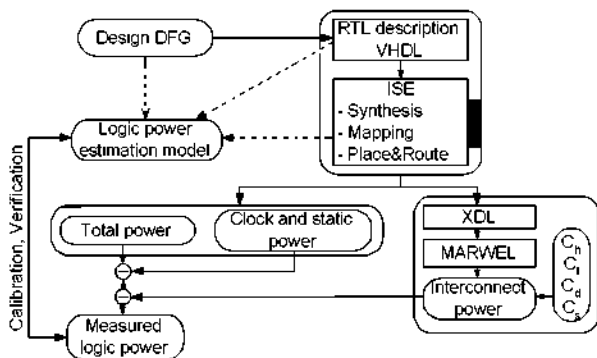


Fig. 6: Estimation flow

of high-level power estimation models. We are only able to compare these results to the work in [6], as they use a similar methodology for obtaining the wire capacitances. The mean error reported in their work is around 12%, with maximum error of 27%. However, these errors refer to the low-level estimation of the whole design power, while we focus only on the interconnection error.

### B. Estimation flow

The following example (see Fig. 6) demonstrates how to use the measurement system for verification and calibration of high-level estimation models.

Suppose that we have high-level logic power estimation models that we want to calibrate first, and then to test their accuracy. High-level logic models can be represented as  $P_{log} = f(K_{set}, V_{set})$ , where  $K_{set}$  are the constant coefficients that are obtained through the calibration and stand beside the variables in  $V_{set}$ . The variables in  $V_{set}$  can be input signal statistics, operand’s word-lengths, clock frequency, etc. First, we choose a set of variable values that are to be used for model calibration (for example, input signals with  $\rho=0$ , operand word-lengths of 16 bits and a frequency of 50MHz). We generate the corresponding input signal file that will be loaded to the Altera board which will stimulate the designs implemented in the XUP board. Then, we design the circuit at the RTL level by using VHDL. After that, the circuit is implemented by using ISE, Xilinx synthesis software.

Next, there are two separate steps we have to take. On the one hand, we load the design to the board and measure the clock power together with the static power and the total power of the design as described in Section III. On the other hand, we generate the XDL file from the description of the implemented design, and apply MARWEL to extract the number of hex, double, direct and long lines used for routing the design. With this information and the wire capacitances given in Table I, we apply equation (3) in order to compute the design interconnect power. Finally, we subtract the static, clock and interconnect power from the total design power and thus, obtain the measured logic power. The whole process is repeated as many times as different combinations of input variable values are considered in the characterization set. With the obtained measured values of logic power, we use the multivariable regression in order to obtain coefficients in  $K_{set}$ . At this point, the

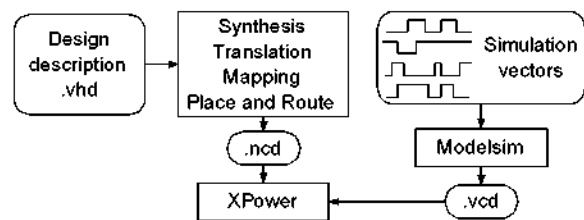


Fig. 7: XPower design flow

models are calibrated, and we can use them for any other values of variables belonging to  $V_{set}$ .

In order to test the accuracy of the models, we choose some variables that have not been used in the characterization set and we repeat the measurement procedure for these values. At the same time, we also apply them to the power models, so we can compare each high-level power estimate with the measured logic power. The same methodology can be used for other power models developed at any level of abstraction (RTL, gate, transistor, etc.) where the variables in  $V_{set}$  may differ from the ones we used in this example (such as the number of LUTs, registers, etc.).

Without this methodology, the calibration of the logic power models based on on-board measurements is not possible. Thus, the most common approach in the literature is to avoid using estimation models. Instead, the relative difference in total power after applying optimization techniques is detected and the optimization step is discarded or adopted accordingly. However, the drawback of this approach lies in the fact that the part of the design that has caused the power increase/decrease cannot be identified. It remains unclear whether this power variation should be contributed to logic, interconnect or clock power. When the methodology presented here is used, the optimization techniques can easily localize the hot spots in the circuit and guide the optimization process as to reduce their power.

### C. Measurements vs. low-level estimations

In the following, we analyze the accuracy of a low-level estimation tool. In particular, we use XPower, a Xilinx low-level tool, for comparison. XPower allows a user to analyze the total dynamic power, and the power per-net, of routed, partially routed or unrouted designs.

The typical XPower design flow is given in Fig. 7. First, a gate-level timing simulation of the placed-and-routed design is run, and as a result, a VCD file is obtained. The VCD file contains detailed information on the toggling rates and frequencies of all the signals in the design, and it is used as the input simulation file for XPower. The output file of the tool is a power report. The report option that provides the most detailed information on design power is the "Advanced" report, and we have always used this option in our experiments. Information about the power of each individual element in the design is listed and sorted by type into the following four groups:

- 1) The power of the clock tree including both, the power of clock nets and the power of all clock buffers, except the input clock buffer (Clock power group).



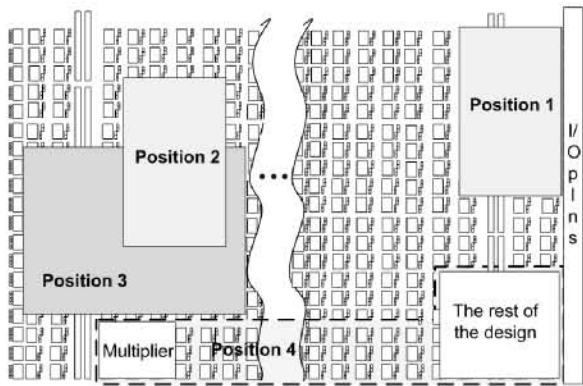


Fig. 8: Design positions for  $DSP_1$

2) The power of logic considering only the power inside CLBs and embedded blocks (Logic power group).

3) The power of signals including both, local connections inside a component, like the connections between the CLBs that form a component, and global connections used between I/O pins and component input and output registers (Signals power group).

4) The power of input buffers (Inputs power group).

The evaluation set consists of three DSP designs that implement the following arithmetic expressions:

$$\begin{aligned}
 DSP_1 &= (x_1x_2 + 1)x_3x_4 + (256x_1 + x_2) \\
 DSP_2 &= (x_2x_3)x_2 + (x_1 + x_3)x_2 \\
 DSP_3 &= ((x_1 + x_2)(x_3 + x_4) + x_1x_2)x_2(x_3 + x_4)
 \end{aligned} \tag{12}$$

Furthermore, four different placements for design  $DSP_1$  are also considered (see Fig. 8). The first placement (position 1) is achieved without using any area constraints. For the position 2, the relative positions of the modules are kept as in the first placement, but all the modules are placed far from the I/O pins. In position 3, a bounding box with the size of a quarter of the FPGA surface is applied as an area constraint, and it is placed on the opposite side of the pins. In position 4, an area constraint for only one of the multipliers is created by placing it far from the I/O pins and the rest of the design.

Evaluating power in four different positions also enabled us to confirm that the interconnect power values computed using MARWEL and the effective capacitances could serve as a fair substitute for direct power measurements. After measuring the dynamic power consumption of the design in the four positions, we subtracted the computed interconnect power from the measured dynamic power for each design position. The results, which represent the logic power, should be the same in all four positions. Indeed, the maximum relative difference between these logic power values was found to be 2.05%.

We have also tested some larger benchmarks (approx. 10 times larger than  $DSP_1$ ). However, the frequency of these designs was lowered (16MHz, 20MHz) in order to avoid the increase in the static power, and smaller resistance values (4.7ohms, 8.2ohms) were chosen so as to satisfy (9). The accuracy of these measurements was confirmed to be the same as the accuracy reported in Section IV-B. However,

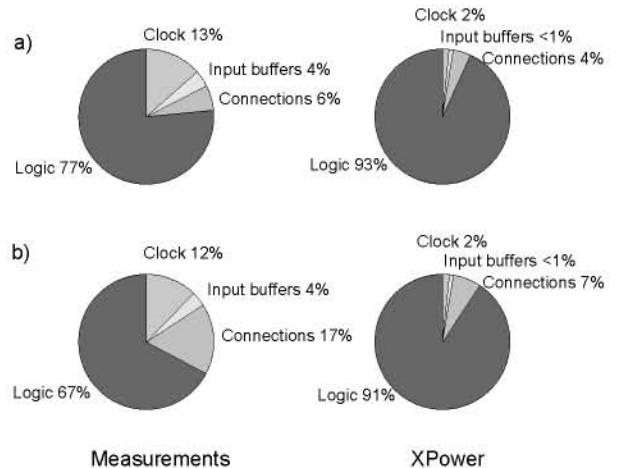


Fig. 9: Power distribution for  $DSP_1$  design in a) position 1, and b) position 2

since we have used different measurement parameters, the results have not been included here.

In order to understand better the error distribution among different power groups provided by XPower, we present a power distribution pie charts for both, measurements and XPower, in Fig. 9. Fig. 9a corresponds to the power distribution of the benchmark  $DSP_1$  when it is located in position 1 (i.e. near the I/O pins), while Fig. 9b corresponds to the power distribution of the same benchmark located in position 2 (i.e. far from the I/O pins). It can be seen that in all cases the percentage of a power group obtained from the measurements compared to the percentage of the same power group obtained from XPower does not match in any of the design positions. Furthermore, XPower fails to account properly for the significant increase in the interconnection power when placing the design further away from the pins.

In both cases, the dominant power component is the logic power. This is in contrast with the expected interconnect power dominance reported in [9]. The reason for this is that the main goal of the designs used here is to achieve the highest measurement precision as to obtain accurate effective capacitance values. Small data-path oriented designs result to be ideal candidates for this purpose, as well as the use of multipliers implemented in LUTs which consume a lot of power (approx. 3-4 times more than embedded multipliers). Additionally, there is no congestion in the interconnections between the components due to the size of the designs, whereas the benchmarks used in [9] are designed to fit the whole FPGA. High resource occupancy in FPGAs leads to a significant increase in the interconnection length and a dominance of the interconnect power.

Fig. 10 represents the error distribution for XPower estimates of the design dynamic power once the static power has been subtracted from the total design power. In the right column we give the errors for each of the presented power components separately: logic, interconnect and clock. The results are given for four different autocorrelation coefficients in order to see the impact of different amounts of glitching generated in logic on the power esti-

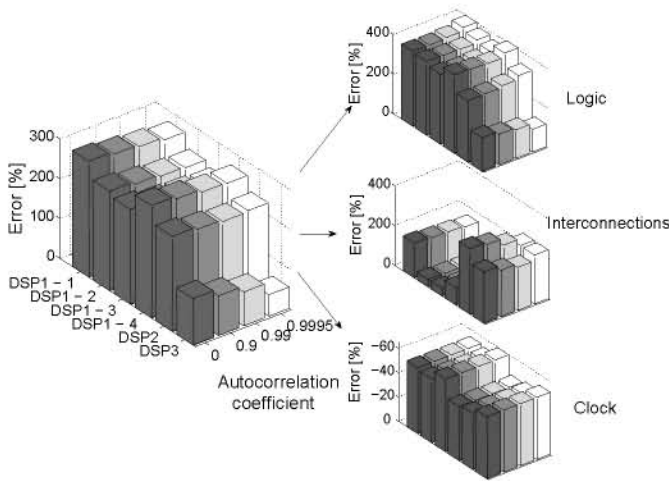


Fig. 10: Error distribution for XPower considering total, logic, interconnect and clock power

mates. We have omitted the input buffer error, as the XPower error was found to be negligible (approx. 3.5%).

It can be seen that XPower has large overestimates (over 300%) for all power components except for the clock power. Furthermore, the interconnect power errors tend to decrease for longer interconnection lengths. For example, when considering the  $DSP_1$  test design in positions 2 and 3 (modules far from I/O pins), the errors drop drastically from 200%, obtained in position 1, to below 50%. It seems that XPower tends to overestimate particularly the consumption in the short connections.

We believe that the large errors of XPower are due to the fact that the reported static power is a constant for the Virtex II Pro device, and that the tool is calibrated to estimate the power of large designs. The power values for interconnects are greater than their real values in order to compensate for the increase in static power due to the higher temperature generated by the activity in large designs. Indeed, the results in [21] demonstrate that the XPower errors are below 30% for larger benchmarks implemented on the same platform. Consequently, it seems that low-level tools are suitable for coarse architecture optimization (order of watts), but they are not suitable for power model validation. A methodology based on on-board measurements should be used instead.

## VI. CONCLUSION

In this work we have presented a measurement system aimed at measuring separate values of static, clock, interconnect and logic power in FPGAs. For this purpose, we have used two FPGA boards, one for measuring power and the other for loading the input vectors into the first one. A tool in C++ has been developed for extracting the lengths of the different wire types used for routing the design. This has allowed for the separation of interconnect power from the rest of the dynamic power. Static and clock power have been obtained from power measurements in different scenarios. The results show that the system is capable of obtaining accurate power values that can be further used for calibration and validation of power estimation models.

Additionally, we have explored the accuracy of XPower. According to the results, XPower provides large overestimates. This could be due to the small size of the designs used in the experiments, as XPower has to compensate for the assumption that the static power does not vary with the activity of the design, and it tends to overestimate all the other dynamic power components.

## ACKNOWLEDGMENT

This work was supported in part by the Spanish Ministry of Education and Science under project TEC2009-14219-C03-02.

## REFERENCES

- [1] D. Macii, and D. Petri. Accurate Software-Related Average Current Drain Measurements in Embedded Systems. *IEEE Trans. on Instrumentation and Measurement*, 56(3):723–730, June, 2007.
- [2] T. Laopoulos, P. Neofotistos, C.A. Kosmatopoulos, and S. Nikolaidis. Measurement of current variations for the estimation of software-related power consumption. *IEEE Trans. on Instrumentation and Measurement*, 52(4):1206–1212, Aug., 2003.
- [3] N. Kavvadias, P. Neofotistos, S. Nikolaidis, C.A. Kosmatopoulos, and T. Laopoulos. Measurements analysis of the software-related power consumption in Microprocessors. *IEEE Trans. on Instrumentation and Measurement*, 53(4):1106–1112, Aug., 2004.
- [4] Xilinx. [www.xilinx.com](http://www.xilinx.com).
- [5] Altera. [www.altera.com](http://www.altera.com).
- [6] V. Degalahal and T. Tuan. Methodology for High Level Estimation of FPGA Power Consumption. In *Proc. of ASP-DAC'05*, pages 657–660, Jan. 2005.
- [7] D. Elléouet, Y. Savary, and N. Julien. An FPGA Power Aware Design Flow. In *Proc. on PATMOS*, pages 415–424, Sept. 2006.
- [8] H. G. Lee, K. Lee, Y. Choi, and N. Chang. Cycle-Accurate Energy Measurement and Characterization of FPGAs. *Analog Integr. Circ. and Signal Proc.*, 42(3):239–251, Mar. 2005.
- [9] L. Shang, A. S. Kaviani, and K. Bathala. Dynamic Power Consumption in Virtex-II FPGA Family. In *Proc. of FPGA'02*, pages 157–164, Feb. 2002.
- [10] S. J. E. Wilton, S. Ang, and W. Luk. The Impact of Pipelining on Energy per Operation in Field-Programmable Gate Arrays. In *Proc. of the FPL'04*, pages 719–728, Aug. 2004.
- [11] D. Elléouet, N. Julien, D. Houzet, J. G. Cousin, and E. Martin. Power Consumption Characterization and Modeling of Embedded Memories in Xilinx Virtex 400E FPGA. In *Euromicro Symposium on DSD*, pages 394–401, Sept. 2004.
- [12] R. Jevtic and C. Carreras. Power Estimation of Embedded Multipliers in FPGAs. *IEEE Trans. on VLSI*, 2009.
- [13] R. Jevtic, C. Carreras, and V. Pejovic. Floorplan-based FPGA Interconnect Power Estimation in DSP Circuits. In *Proc. of SLIP'09*, pages 53–60, July 2009.
- [14] R. Jevtic, C. Carreras, and G. Caffarena. Fast and Accurate Power Estimation of FPGA DSP Components Based on High-level Switching Activity Models. *Int. Journ. of Electronics*, 95(7):653–668, July 2008.
- [15] J. H. Anderson and F. N. Najm. Power Estimation Techniques for FPGAs. *IEEE Trans. On VLSI*, 10(12):1015–1027, Oct. 2004.
- [16] K. Paulsson, M. Huebner, and J. Becker. On-line Optimization of FPGA Power Dissipation by Exploiting Run-time Adaption of Communication Primitives. In *Proc. of the 16th SBCCI*, pages 283–288, Sept. 2006.
- [17] Xilinx. Xilinx University Program Virtex-II Pro Development System. Hardware Reference Manual, v1.0, Mar. 2005.
- [18] R. Erickson and D. Maksimovic. *Fundamentals of Power Electronics*. Springer, 2001.
- [19] Evaluation of measurement data - Guide to the expression of uncertainty in measurements. JCGM, Sept. 2008.
- [20] GTL. <http://www.infosun.fim.uni-passau.de/gtl/>.
- [21] I. Herrera-Alzu, M.A. Sanchez, M. Lopez-Vallejo and P. Echeverria. Experimental Methodology for Power Characterization of FPGAs. In *Proc. on ICECS 2008*, Sept. 2008.