Power Consumption Model for Partial and Dynamic Reconfiguration

Robin Bonamy
CAIRN-IRISA
Université de Rennes1
Lannion
robin.bonamy@irisa.fr

Daniel Chillet
CAIRN-IRISA
Université de Rennes1
Lannion
daniel.chillet@irisa.fr

Sébastien Bilavarn
LEAT
Université de
Nice Sophia Antipolis
sebastien.bilavarn@unice.fr

Olivier Sentieys
CAIRN-IRISA
Université de Rennes1
Lannion
olivier.sentieys@irisa.fr

***Abstract*—In the context of embedded systems development, two important challenges are the efficient use of silicon area and the energy consumption minimization. Hardware accelerated tasks allow to reduce energy consumption of several orders of magnitude, compared to software execution, but these tasks require silicon area and consume power even when they are unused (idle power). Dynamic and Partial Reconfiguration (DPR) brings, to System-on-Chip architectures, an interesting answer by allowing to share a piece of silicon surface between different dedicated accelerators and thus brings the opportunity to reduce power consumption. Nevertheless, many parameters like reconfiguration overhead, accelerator area and performance tradeoff, idle power consumption, etc. make power consumption gain difficult to evaluate. In order to take good implementation choices, it is important to have a precise power and energy consumption estimation of the partial reconfiguration process. In this context, this paper presents a detailed investigation of power consumption of a DPR process using Xilinx ICAP reconfiguration controller. From these results we propose three power models with different complexity/accuracy tradeoffs which helps to analyze the benefits of using accelerated and dynamically reconfigurable tasks in comparison with classical static configuration or full software execution.**

I. Introduction

Run-time reconfiguration, the ability to change a configuration while the rest of the circuit is running, is a research subject since the 90s [1] and is now commonly used in FPGAs, configurable circuits, since Xilinx and Altera provide such circuits. The main advantages of the partial, run-time reconfiguration are to add hardware flexibility and to reuse hardware area, allowing power and production costs reductions. During last decade, much work has been done on tools to exploit the reconfigurable aspects of these configurable circuits. Configurable areas become an usual extension of System-On-Chips (SoC) which are increasingly used in integrated and embedded systems where the energy consideration is substantial. However, there is few work in the characterization and estimation of the power and energy consumption during reconfiguration.

Considering this lack of model for partial and dynamic reconfiguration process, this paper proposes to define an accurate model based on an accurate analysis of the dynamic and partial reconfiguration (DPR) power consumption.

Depending on the application constraints, several choices can be considered, i) implement a static hardware task, ii) implement a dynamic hardware task and reconfigure it when necessary, iii) implement a software task running on the processor.

We claim that through an accurate model of reconfiguration process, it can be possible to exploit dynamic and partial reconfiguration by management of the task scheduling in order to reduce the energy consumption. Defining an accurate model of power consumption is the main contribution of this paper. To build this model, which can be included in the global power/energy estimation of the SoC, we measure the power consumption of the reconfiguration process and extract relevant parameters.

This paper first introduces state of the art on the effects of DPR on power consumption. Then in section III, FPGA reconfigurable architecture is presented. Section IV poses the experimental setup for power measurements which are analyzed in detail in section V. Then section VI exposes models for power estimation and next section discusses the application of models and estimation accuracy. Finally, we conclude this work and present future work.

II. State of The Art

DPR is definitely a key feature to increase the flexibility of hardware devices such as FPGAs. Reconfigurable devices can be self modified following user's constraints and adapted to environmental evolutions [2]. DPR is also widely used to increase area usage [3] and return on static power consumption that may reduce the overall power consumption. Clock configuration modifications are possible involving DPR, association of dynamic frequency adaptation and variation of performances to, on demand, adjust power consumption and energy efficiency [4]. DPR can also disable clock routing to FPGA basic elements thus providing clock gating with interesting efficiency and low overhead [5].

The main drawback of the DPR, in most of applications, is the unavailability of the concerned area during the process. It is possible to reduce the DPR time by using more effective reconfiguration data path [6]. Tips to reduce configuration data size will reduce reconfiguration time, like using differential based configuration, as an example. It is so consequential to model and consider the reconfiguration time cost at the design stage of the system as presented in recent works [7], [8]. DPR

978-1-4673-2921-7/12/$31.00

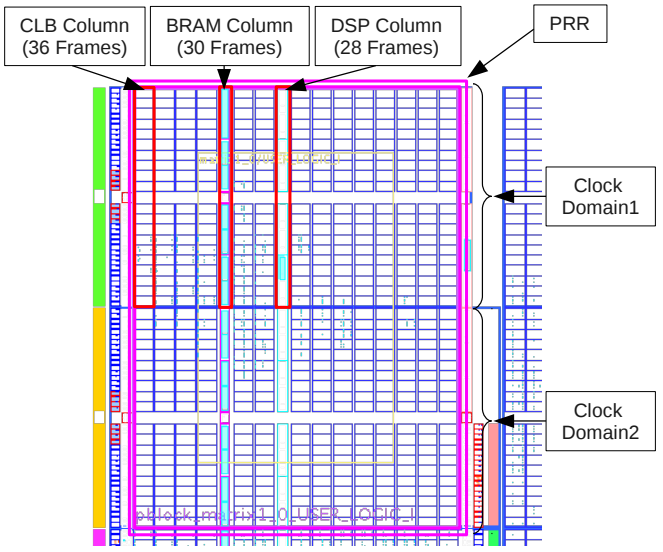

Fig. 1. Top left hand of Virtex-5 XC5VLX50T FPGA layout, approximately one quarter (PlanAhead capture).

time overcost is important to take into account, but in embedded systems, energy savings during execution is a continuous challenge. In the literature, we can find studies on power consumption of the dynamic reconfiguration process [9], [10] and more recently [11] , but there is no detailed information about the power consumption during partial reconfiguration and no power model is presented.

In the scope of the Open-PEOPLE project [12], which aim is to provide a complete heterogeneous hardware and software platform for power and energy estimation, optimization and measurement , models of the energy consumption of complete MPSoCs are proposed. In this project, one major contribution concerns the definition of accurate model for reconfigurable devices. In particular, we propose a power/energy model of the dynamic and partial reconfiguration process to help the designer to select the best implementation between static hardware tasks, software execution or dynamic reconfiguration."

III. FPGA Architecture and Partial Reconfiguration

DPR is currently available for Xilinx Virtex circuit family, and would be available soon for Altera devices. In consequence, our work is based on Virtex devices and more specifically on Virtex 5. Virtex 5 FPGA is organized in clock domains, the XC5VLX50T used in our experiments has 6 clock domains. In these clock domains, configuration is organized in frames, organized in columns, containing either CLBs (slices), DSPs, BRAMs or other special blocks as presented in figure 1. One frame configuration requires 41 words of 4 bytes (164 bytes) and the number of frames in one column is dependent on the type of these columns. The minimum addressable reconfiguration area is a frame and the minimum recommended reconfiguration area is a column, so the partial reconfigurable region (PRR) must be a multiple of frames and mainly contains CLBs, DSPs and BRAMs.

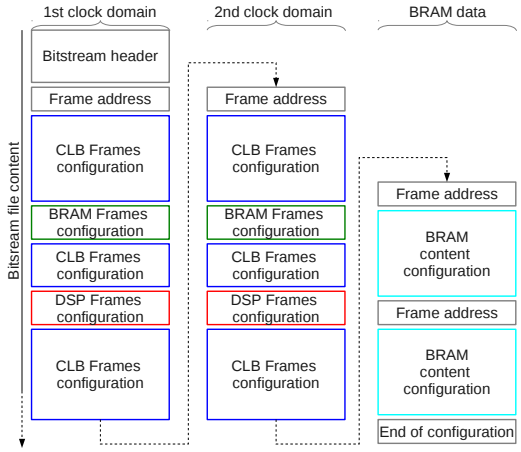

Fig. 2. Bitstream composition to configure our PRR, using Xilinx ISE 12.1 tool suite.

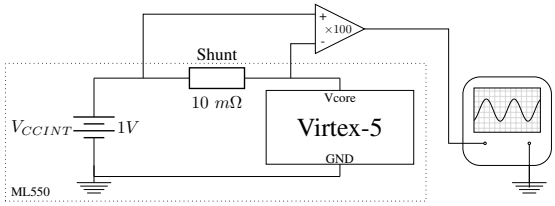

Fig. 3. Current measurement schematics on ML550 Board using a high-precision amplifier and an oscilloscope.

Figure 2 represents the configuration data file (bitstream) organization to configure a PRR example, presented in figure 1. The first words are the header, containing information on the bitstream and configuration starting procedure. The next word contains the address of the first frame to configure. Then the configuration words for the CLB blocks follow. After 4 CLB columns, configuration for one BRAM column is present, following two other CLB columns and one DSP column. Finally CLB configuration data up to the right bound of the PRR are present. Another frame address is pointed, corresponding to the first column of the second clock domain which has the same structure as the previous. Finally the BRAM content is addressed and few words end configuration.

IV. Experimental Setup

A. Power Measurement

To ease the extraction of power measurements, a Xilinx ML550 board is used in the following experiments. Five access points are present on the board to measure the currents consumed by the FPGA and its peripherals. FPGA power values are based on measures made on the power rail of the core. As represented in figure 3, a high-precision amplifier is used to enhance the signal level and the amplified signal is then send to digital oscilloscope. With this procedure, it is possible to measure low currents and power consumption as low as $0.1\ mW$. As it will be shown in the following, this precision is sufficient to clearly identify the different steps of a dynamic reconfiguration process and to define a power model.

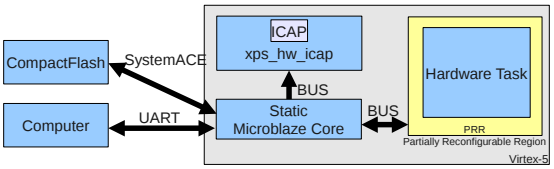

Fig. 4. FPGA content during measurement procedure.

TABLE I
RESOURCES REQUIREMENTS AND IDLE POWER CONSUMPTION OF AVAILABLE CONFIGURATIONS OF THE PRR AND THE FPGA POWER CONSUMPTION.

Task	BRAMs	Slices	DSPs	Idle power(mW)
T_1	8	169	2	26
T_2	8	154	1	24
T_E	0	0	0	0
FPGA without task				402

B. Platform Setup

The ML550 platform is configured using the reference design proposed by Xilinx in [13]. The designed is composed of a `Microblaze` software processor, a CompactFlash memory controller and a Xilinx reconfiguration controller (`xps_hw_icap`) as presented in figure 4. The `Microblaze` processor and the reconfiguration controller operate both at a constant frequency equal to 100 MHz. Reconfigurations requests are managed by the `Microblaze` which reads the configuration bitstreams from the CompactFlash, and sends them to the `xps_hw_icap` which will apply the new configuration to the PRR.

In the following experiments, one PRR fitting the size of the biggest task area is used. We choose a PRR which holds in the top left hand of the FPGA layout, as presented in figure 1, and takes the complete half left area over two clocks domains. It corresponds to a configuration bitstream of a fixed size of 227700 $Bytes$. During a measurement procedure, the FPGA core is powered at 1 V and we check that the die temperature is stable (around 35° C), to focus on power consumption variations only due to the DPR.

In our proposed measurement procedure, we can clearly detect the distinct phases of the reconfiguration process.

C. Reconfiguration Region Setup

As previously defined, our power measurements are done on a PRR which can be configured with three different tasks. The first one, named T_1, is a matrix multiplication. The second task, called T_2, is another version of the matrix multiplication. The RTL descriptions of these dedicated tasks are derived using High Level Synthesis and loop unrolling [14], the corresponding FPGA resources are given in table I. The third task used is called a *blank* bitstream, referred as T_E in the following. A *blank* bitstream corresponds to an empty configuration of a PRR and this can be used to clear the configuration and reduce power consumption [15].

In the following, we analyze the effect of reconfiguring these tasks on the FPGA power consumption.

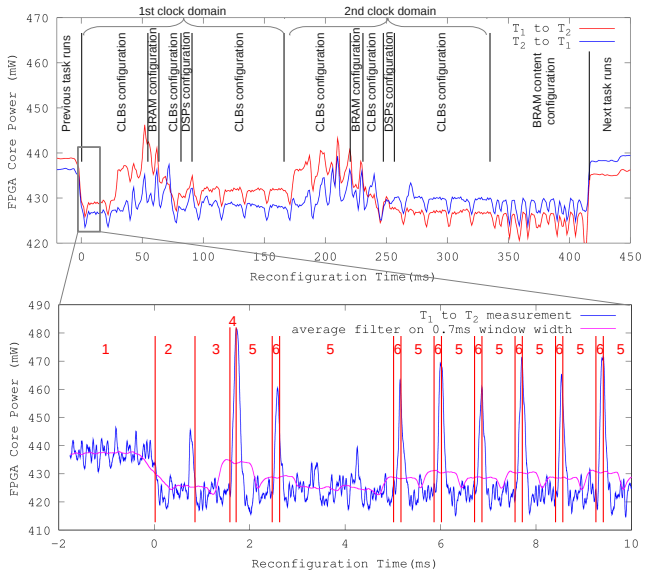

Fig. 5. Top plot is the FPGA core power consumption during DPR from T_2 to T_1 versus time, in blue, and from T_1 to T_2, in red. Bitstream composition is scaled to the reconfiguration time to outline reconfiguration steps. Bottom plot is a zoom on the beginning of the DPR.

V. POWER CONSUMPTION ANALYSIS

The overall DPR process mainly consists of transferring data to the players involved (CompactFlash, `MicroBlaze` memory, reconfiguration controller, configuration memory) before the effective configuration application. The full procedure can be split in the following basic steps, configuration data is:

- loaded from a file stored on CompactFlash memory and buffered in the `MicroBlaze` memory
- written from the `Microblaze` memory to the reconfiguration controller, `xps_hw_icap`
- written from the reconfiguration controller to the Internal Configuration Access Port (ICAP)
- written from the ICAP to the configuration memory
- applied from the configuration memory to the configurable resources (CLB, BRAM, DSP, interconnect)

Under these circumstances, the global reconfiguration power is the result of a combination of two main elements: the activity of bitstream access and management and the actual configuration of FPGA resources. In the following, we examine in detail power activities resulting from configuration data transfer and resource configuration.

A. Data Reading and Writing

Figure 5 (top) reports the power profiles of the FPGA core during DPR from T_2 to T_1 (blue), and from T_1 to T_2 (red). If we inspect closely the full reconfiguration process in this figure, we can clearly observe the power variations due to the DPR procedure.

Second plot of figure 5 is a zoom on the first 10 ms of DPR. To provide even finer power analysis, we have introduced software triggered markers in the reconfiguration controller to highlight each step of the process.

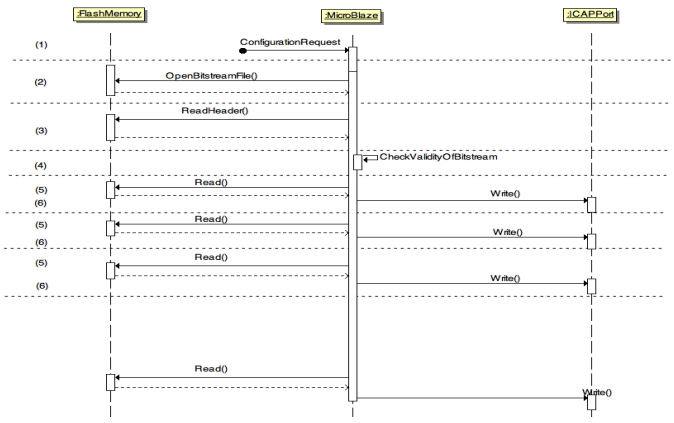

Fig. 6. Sequence diagram of the reconfiguration process.

Seven steps are identified and presented in the sequence diagram of figure 6:

- 1) reconfiguration order arrives
- 2) open bitstream file on the CompactFlash
- 3) read bitstream file header
- 4) check validity of bitstream file header
- 5) read file fragment on the CompactFlash
- 6) write data in `xps_hw_icap`
- 7) repeat steps 5 and 6 until the end of file.

Power consumption during CompactFlash reading (phases 3 and 5) is low. Due to CompactFlash slowness, the `MicroBlaze` is waiting for data and does not generate a lot of activity. However writing (phase 6) implies much more work, involving bus traffic, reconfiguration control and FPGA resource configuration. The corresponding power overheads are about $45\ mW$ which is more than $10\ \%$ of the overall FPGA power consumption. There is another power overhead during bitstream validity check (phase 4). This overconsumption is even higher because it is processed by the `MicroBlaze` CPU which has the largest part of power. Note that these power peaks are not visible on figure 5 because of an averaging filter used for display convenience.

B. Configuration Application

Previously, bottom of figure 5 clearly showed the influence of data transfer on power consumption. However, the average power consumption over the seven first read and write cycles should be the same during the whole reconfiguration phase, as we can see on the magenta curve which is a sliding window average over $0.7\ ms$. Nevertheless, both power curves shown on top of figure 5 (which have been filtered) are not as flat as predicted from the single read and write cycles. First, power overconsumption are noticeable and look like two "waves" located around 50 and $200\ ms$. Secondly, power consumption at the beginning and at the end of the reconfiguration is not the same. This power seems linked to the currently configured task, even when this task doesn't run. The shape seems to be linked to some other parameters than the reconfiguration controller.

We assume that power consumption variations are due to bitstream content and we suppose this consumption can be separated in two parts: power surges and power steps. A reconfiguration can cause power overconsumption due to activity generated by the configuration modification (power surges). Then reconfiguration may enable or disable signals and blocks that can cause power consumption breaks (power steps).

The following sections present the analysis of power consumption curves which justifies our assumptions.

1) Power Surges: Figure 5 presents the power consumption during the reconfiguration of task T_2 where T_1 was previously configured and vice versa. Both curves have the same overall shape, which is mainly driven by the write operation in the reconfiguration memory. Indeed, when a task is configured over a previous task, the reconfiguration process consists in writing data on the configuration memory. If both n^{th} words of the bitstreams of tasks T_1 and T_2 are the same, in this case the memory cells don't change and the power consumption is low. Inversely, if the two words of the two bistreams are exactly complementary, in this case each bit of the memory cell changes and this leads to a larger power consumption. From this observation, we assume that the power consumption during the reconfiguration process is linked to the difference between the bitstream of task before the reconfiguration and the bitstream of task after the reconfiguration. These difference can be quantified by a metric like the Hamming distance.

On the curves of figure 5, two remarkable zones are present: the first one is located around $50\ ms$ and the second one around $200\ ms$. The amplitude of overconsumptions is about $15\ mW$ in this case. Further analysis of the bitstream content shows that these two zones correspond to the configuration of two BRAM frames. The BRAM frames belong in two distinct clock domains which explains the presence of two of overconsumption zones. By relating this information with the implemented design view in Xilinx floorplanning tool (PlanAhead), we can notice that the occupation rate of slices close to the BRAM frames is higher compared to more distant slices. This placement of logic around the BRAM blocks can be explained by place and route algorithms which probably try to group the resources used in order to limit the interconnect cost.

Top of figure 7 represents the power consumption of the FPGA core during the reconfiguration of a task T_2 on a PRR where a previous task T_1 was configured. Bottom of this figure shows the Hamming distance (word by word of $32\ bits$) between the bitstreams of the two concerned tasks. Abscissa values are adjusted to tie in the reconfiguration time (ms). We notice on these curves that the shape of Hamming distance between the bistream of the previously implemented task and the new task is close to the shape of the power consumption profile. The Hamming distance peaks at the same time when the overconsumptions are present, around $50\ ms$ and $200-250\ ms$. This strengthens the assumption of the link between configuration differences and overconsumptions.

Moreover another experiment is set up. This experience consists in stopping the clock during the reconfiguration (and thus

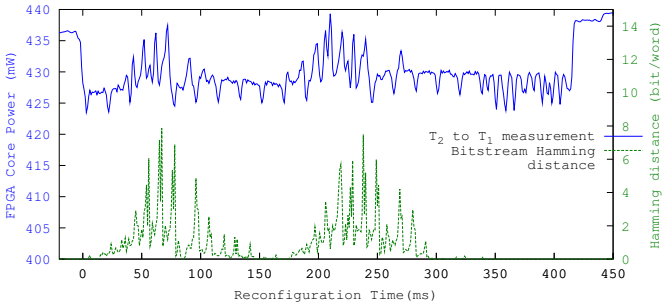

Fig. 7. FPGA core power consumption during DPR from T_2 to T_1 versus time, in blue, and Hamming distance per configuration word between T_1 and T_2 bitstreams versus reconfiguration time, dotted green line.

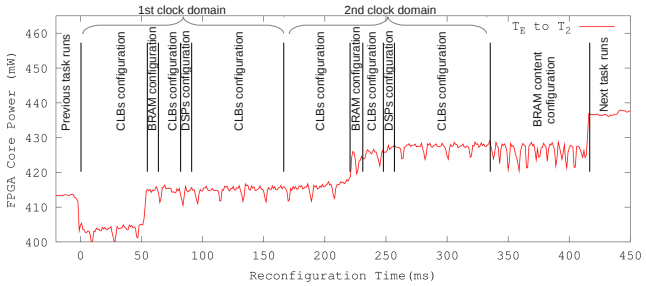

Fig. 8. FPGA core power consumption during DPR of a PRR containing T_E to T_2 versus time and bitstream composition is presented.

stopping the reconfiguration for a while), especially before and during an overconsumption zone to compare static power. This experience highlighted that power overconsumptions are, partially, independent of the configuration write process, and composed of static power consumption. Linking this experience and figure 7 reveals that configuration differences cause static power consumption during partial reconfiguration.

Previous statement suggests that overconsumptions result from activating and deactivating interconnect between FPGA resources. This may cause unwanted connections and perhaps small short circuits while the PRR is not fully configured as mentionned in [10], considering an ATMEL device. Modifications of the FPGA interconnect are indicated by the bitstream and the number of the interconnections modified are reflected in the Hamming distance of the configuration words. At this stage and for this platform, it is difficult to determine which are the interconnection configuration bits in the bitstream file that should be taken on to refine the power model.

2) Power Steps: Looking more closely to figure 5 describes that power level during, and especially, at the beginning and the end of the reconfiguration are different according to the tasks concerned by the process. This difference is concordant with tasks power consumption presented in table I. Both T_1 and T_2 don't have the same resource usage and the same idle power consumption.

This idle power is linked to the task's size and resource occupation. So power consumption analysis during DPR of two very different tasks' idle power consumption will highlight the power behavior to switch from the previous task to the next task's idle power.

As an illustration, figure 8 presents the FPGA core power profile during the partial dynamic reconfiguration of a PRR from T_E to T_1, where T_E is an empty task. This figure explicitly shows two power steps during reconfiguration process. These steps bring the power consumption from the idle power level of the former task T_E to the idle power level of the latter task T_1.

The advanced decomposition of the bitstream data, presented in figure 8, shows that one power consumption step appears just before the configuration of BRAM interconnections on a Xilinx Virtex-5. This behavior is typical of activating or deactivating elements, there is probably a link with the power consumption of BRAM memory, where the consumption is different depending on the enable state, or clock routing to CLBs.

This section has presented the analysis of power consumption during DPR. It showed that there are three main components involved in partial reconfiguration power consumption. First there is an overhead due to the reconfiguration controller to transfer bitstream. Then, there are power surges due to configuration differences between the previous and the new configuration. Finally, there are power steps related to the establishment of the resources of the newly configured task. From these detailed measures and analysis, the following section defines three power models of DPR with different complexity/accuracy tradeoffs.

VI. Power Consumption Estimation

A. Coarse Grained Model

The easiest way to estimate the power of reconfiguring a PRR is to record measurements of multiple reconfigurations and to consider the average value. This average value is subtracted to the idle power consumption before the reconfiguration to only keep the reconfiguration power value. It is considered as the reconfiguration controller power consumption and defined by $P_{controller}$ in the following. Under the experimental conditions, the FPGA idle power consumption before the reconfiguration can be defined as the FPGA idle power consumption with an empty PRR, P_{FPGA}, and the PRR's idle power before the reconfiguration, P_{before}.

The coarse grained power model is defined as:

$$P_{CG}(\tau) = P_{FPGA} + P_{before} + P_{controller} \quad (1)$$

With τ is a time period corresponding to the configuration time of one word (32 $bits$ of the bitstream).

B. Medium Grained Model

An improved method is based on interpolating between the FPGA idle power before and after the configuration, by drawing a straight line between them. This approach requires idle power of the tasks configured before and after in addition to the power of the reconfiguration controller. The corresponding model is represented by the following equation:

$$\begin{aligned} P_{MG}(\tau) &= P_{FPGA} + P_{before} + P_{controller} \\ &+ (P_{after} - P_{before}) \times \frac{\tau}{BS_{size} \times T_{word}} \end{aligned} \quad (2)$$

with BS_{size} is the bitstream size in Bytes, T_{word} is the time required to configure a word (32 $bits$). P_{after} is the idle power consumption of the PRR with the newly configured task.

This medium grained model considers that idle power consumption of the PRR can change using DPR. PRR's power is interpolated between the beginning and the end of the reconfiguration which is more accurate than the coarse grained model. However, section V showed that PRR's power consumption during the reconfiguration is composed of power surges and power steps. The following model considers these parameters.

C. Fine Grained Model

In this section, we present a finer model of reconfiguration power based on the parameters used in previous section. But, contrary to the medium grained model based on interpolation where the power profile varies linearly from P_{before} to P_{after}, here, the profile evolves non linearly, by steps which are dependent on the bitstream's contents. Then power surges are also considered in this estimation with the Hamming distance reflecting the configuration words difference between both bitstreams.

The resulting model is defined by the following equation:

$$\begin{aligned} P_{FG}(\tau) &= P_{FPGA} + P_{before} + P_{controller} \\ &+ steps(\tau) \times (P_{after} - P_{before}) \\ &+ \alpha \times d_{Hamming}(\tau) \end{aligned} \quad (3)$$

where $steps(\tau)$ is a function which result is between 0 and 1 and depends on the technology and PRR size which, in our measures, seems to be linked to the BRAM position. This value is computed from the bitstream size and reconfiguration speed. Multiple intermediate values are possible depending on the PRR's size and content.

$d_{Hamming}(\tau)$ is the function to compute Hamming distance between the previous and the next configuration of the PRR, performed on $32 - bits$ word with a floating window average of 100 words. This average is required because differences in all words in a frame may not cause the same overconsumption. Hamming distance average allows a better estimation of the real power consumption which is required during frames reconfiguration.

Finally α is the coefficient adjusted for a given technology, calibrated by a minimization algorithm to minimize average error.

This section has presented three power models with different complexities. The following section sets up these models for our experiment and discusses the results and the precision of the estimation.

VII. Results

A. Model Implementation

1) Corse Grained Model: Under the experimental conditions presented in section IV, this section presents the values of parameters measured or computed to enable power estimation.

P_{FPGA}, the FPGA idle power consumption with an empty PRR, P_{before} and P_{after} the PRR's idle power before and after the reconfiguration are measured and presented previously in table I. P_{FPGA} is $402\ mW$. P_{before} and P_{after} are selected following the configuration case.

$P_{controller}$ is the average extra power required for the reconfiguration controller to perform reconfiguration. This power has been measured and is $P_{controller} = 20\ mW$.

2) Medium Grained Model: Considering the medium grained model, three other parameters are required. The bitstream's size to reconfigure the PRR is $BS_{size} = 227700\ B$ and during the experiments the reconfiguration takes $T_{Reconfiguration} = 422\ ms$. Finally, the one word configuration time, T_{word}, is computed with the following equation:

$$T_{word} = \frac{T_{Reconfiguration}}{BS_{size}} \times 4 \quad (4)$$

so $T_{word} = 7.4\ ms$ here.

3) Fine Grained Model: Fine grained model requires three more parameters, $d_{Hamming}(\tau)$, α and $steps(\tau)$.

First $d_{Hamming}(\tau)$ is the function that returns the Hamming difference computed word per word between the previous configuration and the new configuration data. This value is filtered with a floating window average over 100 words. Exception is done on the end of the bitstream, the part corresponding to BRAM data. $d_{Hamming}(\tau)$ returns 0 in this case since BRAM content configuration doesn't have the same power behavior as the rest of the configuration.

Then α is the weighting factor of the Hamming distance. This factor is determined with an iterative optimization algorithm to minimize the average absolute power error between the estimated power and the measurement. Considering T_2 to T_1 configuration, we find that $\alpha = 0.003$ is the optimized Hamming distance power factor.

Finally, the formalization of power steps, $steps(\tau)$, is derived from the FPGA layout. Presented in section IV-B, considered PRR takes place on two clock domains with one BRAM column in each. As seen in section , power steps occur before the BRAM interconnect configuration, so there are two equal steps here. In the bistream, before reaching the first BRAM configuration frame, four CLB columns are present. According to Xilinx documentation [16], a CLB column requires 36 frames and one frame is 41 words. So the first step is at $41 \times 36 \times 4 = 5904$ words. And the second is at the same position in the second clock domain, ie 31898 words, without considering addressing commands and NOPs that hold a few words in the bitstream, and is maintained up to the end of the bitstream: 56925 words. Then

$steps(\tau) = 0 \quad \forall \tau \in [0 - 5903]$
$steps(\tau) = 0.5 \quad \forall \tau \in [5904 - 31897]$
$steps(\tau) = 1 \quad \forall \tau \in [31898 - 56925]$.

The power profiles resulting from these models are represented on figures 9, 10, 11 and 12. Red curve represents the power measured during the experiment, in black the power estimated by the coarse grained model. In green, the medium grained power estimation is presented, finally, blue curve shows the fine grained model result. These result and curves are discussed in the following.

B. Energy Accuracy

Four cases are considered to cover the different reconfiguration scenario: reconfiguration from T_1 to T_2 and T_2 to T_1

TABLE II
ENERGY MEASURED (M) DURING DPR AND ENERGY ERROR USING COARSE GRAINED (CG), MEDIUM GRAINED (MG) AND FINE GRAINED (FG) MODELS

Config	M (mJ)	CG error (mJ)	MG err. (mJ)	FG err. (mJ)
T_2 to T_1	9.12	-1.3 (-13.7%)	-0.84 (-9.2%)	0.5 (5.5%)
T_1 to T_2	9.58	-0.89 (-9.2%)	-1.3 (-13.5%)	-0.22 (-2.3%)
T_1 to T_E	7.97	6.2 (77%)	0.59 (7.4%)	-0.13 (-1.7%)
T_E to T_2	10.41	-7.1 (-67.9%)	-2.1 (-19.7%)	0.5 (5%)
Average	**9.27**	**-0.77 (-8.3%)**	**-0.9 (-9.8%)**	**-0.16 (1.8%)**

which results in very similar curves and configuration from T_1 to T_E and T_E to T_2 are studied.

The accuracy of each power model is evaluated by the average error and standard deviation compared to measured power curve. Average power is directly related, by the reconfiguration time, to energy. Since reconfiguration time is the same in the experimental conditions, energy is comparable in the four considered cases.

Table II reports energy computed from measurements and estimated energy errors for the three models. As visible in figures 9 and 10, which relate power consumption and estimation for the reconfiguration from T_1 to T_2 and vice versa, coarse grained and medium grained models provide a good energy estimation with less than 14% error. This small error is limited by the limited idle power differences between T_1 ans T_2.

Looking at reconfigurations including T_E which idle power consumption is very different from the two other tasks, figures 11 and 12, coarse grained model prediction is bad, 77% and 68% of energy errors. This is caused by the fact that this model does not consider the idle power consumption differences between configured tasks and increases estimation error. Contrary to the coarse grained model, the medium grained model limits its energy error to 20% thanks to the consideration of both previous and next PRR's task's idle power consumption.

Fine grained model provides a very good energy estimation in all cases, less than 6% energy error, thanks to the consideration of power surges and power steps. These characteristics are viewable on the figures and the estimated power, blue curve, have globally the same shape as the measured power, in red.

Considering the average energy error, both coarse grained model and medium grained model are good, respectively 8.3% and 9% error. However coarse grained model has huge estimation errors that compensate on average. Fine grained model proposes a very accurate estimation with an error of 1.8%.

C. Power Accuracy

With the average power error, related to the energy presented in previous section, an important accuracy criteria is the power error standard deviation. It gives an information on the model's precision to predict the good power behavior.

Power error standard deviation values are presented in table III for the three models and four configuration cases. Average values done on all reconfiguration cases show that the coarse grained model has a high power error standard deviation,

TABLE III
POWER ERROR STANDARD DEVIATION USING COARSE GRAINED (CG), MEDIUM GRAINED (MG) AND FINE GRAINED (FG) MODELS, COMPARED TO MEASUREMENTS

Config	CG (mW)	MG (mW)	FG (mW)
T_2 to T_1	2.7	2.7	3.2
T_1 to T_2	4.9	4.6	3.8
T_1 to T_E	10	5	4.3
T_E to T_2	8.2	3.9	3.7
Average	**6.5**	**4.05**	**3.75**

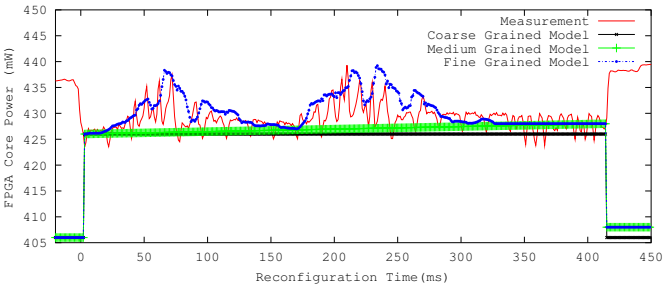

Fig. 9. Measurements and estimation results for FPGA core power consumption during DPR from T_2 to T_1 versus time.

$6.5\ mW$, which is 32.5% of the reconfiguration controller average power. This error is due to its inability to evaluate power surges and power steps.

As predictable, medium grained model, with a standard deviation of $4.05\ mW$, is better than the former, thanks to its capacity to roughly follow power steps.

Finally, fine grained model is slightly better, with a standard deviation of $3.75\ mW$ which is 18.8% of the reconfiguration controller average power. This accuracy parameter shows the limitations of the raw Hamming distance computation. Power surges are mainly overestimated when T_E comes into play, probably because of Hamming distance symmetry which is not the case of these power surges as we can see in measurements between figures 11 and 12. Moreover, power slight and fast variations are not estimated which increases the deviation. But, obviously, the power envelope is mainly well estimated and most of power peaks are estimated.

D. Model Conclusion

The three proposed models are interesting in power and energy modeling of reconfigurable devices and addresses different goals. Coarse grained model is used for a fast estimation. It's good energy accuracy, in average, is suitable to estimate a global energy consumption of a reconfigurable system. Medium grained model also provides a good energy

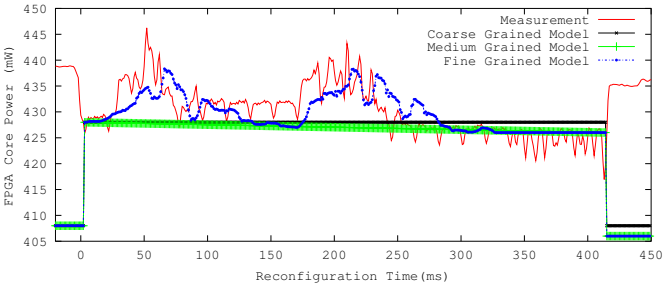

Fig. 10. Measurements and estimation results for FPGA core power consumption during DPR from T_1 to T_2 versus time.

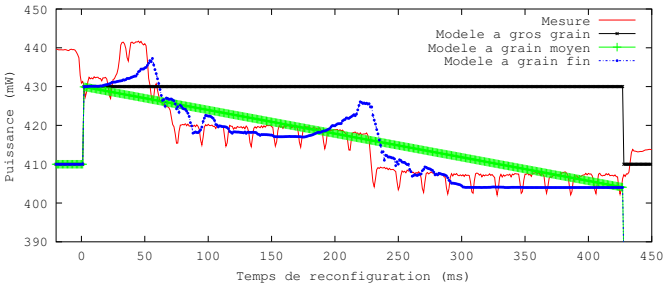

Fig. 11. Measurements and estimation results for FPGA core power consumption during DPR from T_1 to T_E versus time.

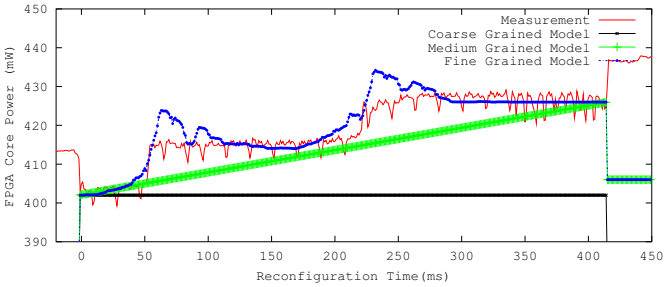

Fig. 12. Measurements and estimation results for FPGA core power consumption during DPR from T_E to T_2 versus time.

accuracy, but this accuracy is reasonable in all reconfiguration cases and not only in average contrary to the coarse grained model. This model is appropriate for a good power and energy estimation of each reconfiguration process. Finally, the fine grained model provides a very accurate energy and power profile which can be useful to analyze more sensitive parameters like thermal behavior or battery lifetime. It's good power peaks estimation is useful in systems where these peak may cause malfunctions or reduce battery recoverability.

It is important to stress that the efficiency of the reconfiguration controller has a very large impact on power. In the proposed experimental setup (based on standard Xilinx procedure), it is very dependent on the use of a CompactFlash with the `xps_hw_icap` and `MicroBlaze`. Recent works like [17] propose a more efficient framework based on optimized reconfiguration control where the `MicroBlaze` is not needed and DDR memory is used instead of CompactFlash to increase throughput. In [11], energy performances are 45 times better but power variations are much more difficult to measure because of the level of currents and the current probe bandwidth in this case. However, these effects are also present when performing reconfiguration at a higher throughput and using a higher energy efficient reconfiguration controller, effects of reconfiguration and bitstream differences are proportionally more important and should be considered in power and energy estimators.

VIII. Conclusion

Basing ourselves on a reference procedure to perform dynamic partial reconfiguration in FPGAs, this paper presents different measurements of the core power consumption during partial reconfiguration. The measurements show that the power consumption is not as simple as we might expect, it is dependent on the previous configuration and the reconfigurable region contents. Based on a detailed bitstream and power consumption analysis, we proposed three models at different grain to estimate power or energy consumption during partial reconfiguration from an simple estimation to a highly detailed equation considering tasks configuration differences and bitstream composition.

This power characterization and estimation work will be extended to be included in a system on chip energy estimator and optimizer using dynamic partial reconfiguration.

References

[1] E. Lemoine and D. Merceron. Run time reconfiguration of fpga for scanning genomic databases. In *FPGAs for Custom Computing Machines, 1995. Proceedings. IEEE Symposium on*, 1995.

[2] V. Tadigotla, L. Sliger, and S. Commuri. FPGA implementation of dynamic run-time behavior reconfiguration in robots. In *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, 2006.

[3] James G. Eldredge and Brad L. Hutchings. Run-Time Reconfiguration: A Method for Enhancing the Functional Density of SRAM-based FPGAs. *The Journal of VLSI Signal Processing*, 12:67–86, 1996.

[4] Xun Zhang, H. Rabah, and S. Weber. Dynamic slowdown and partial reconfiguration to optimize energy in fpga based auto-adaptive sopc. In *Electronic Design, Test and Applications, 2008. DELTA 2008. 4th IEEE International Symposium on*, 2008.

[5] L. Sterpone, L. Carro, D. Matos, S. Wong, and F. Fakhar. A new reconfigurable clock-gating technique for low power SRAM-based FPGAs. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2011*, 2011.

[6] C. Claus, B. Zhang, W. Stechele, L. Braun, M. Hubner, and J. Becker. A multi-platform controller allowing for maximum dynamic partial reconfiguration throughput. In *Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on*, 2008.

[7] M. Rullmann and R. Merker. A cost model for partial dynamic reconfiguration. In *Embedded Computer Systems: Architectures, Modeling, and Simulation, 2008. SAMOS 2008. International Conference on*, 2008.

[8] F. Duhem, F. Muller, and P. Lorenzini. Reconfiguration time overhead on field programmable gate arrays: reduction and cost model. *Computers & Digital Techniques, IET*, 6:105–113, 2012.

[9] J. Becker, M. Huebner, and M. Ullmann. Power estimation and power measurement of xilinx virtex fpgas: trade-offs and limitations. In *Integrated Circuits and Systems Design, 2003. SBCCI 2003. Proceedings. 16th Symposium on*, 2003.

[10] Michael Lorenz, Luis Mengibar, Mario Valderas, and Luis Entrena. Power consumption reduction through dynamic reconfiguration. In *Field Programmable Logic and Application*. Springer Berlin / Heidelberg, 2004.

[11] R. Bonamy, H-M. Pham, S. Pillement, and D. Chillet. UPaRC - Ultra Fast Power aware Reconfiguration Controller. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2012.

[12] Open-PEOPLE - Open-Power and Energy Optimization PLatform and Estimator, http://www.open-people.fr/.

[13] Xilinx. UG744 - PlanAhead Software Tutorial: Partial Reconfiguration of a Processor Peripheral.

[14] Robin Bonamy, Daniel Chillet, Olivier Sentieys, and Sebastien Bilavarn. Parallelism Level Impact on Energy Consumption in Reconfigurable Devices. *ACM SIGARCH Computer Architecture News*, 39:104–105, 2011.

[15] Shaoshan Liu, Richard Neil Pittman, and Alessandro Forin. Energy reduction with run-time partial reconfiguration. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, 2010.

[16] Xilinx. UG191 - Virtex-5 FPGA Configuration User Guide.

[17] Duhem, F. and Muller, F. and Lorenzini, P. FaRM: Fast Reconfiguration Manager for Reducing Reconfiguration Time Overhead on FPGA. In *Reconfigurable Computing: Architectures, Tools and Applications*. Springer Berlin / Heidelberg, 2011.