Efficient Modelling of FPGA-based IP blocks using Neural Networks

Jordane Lorandel, Jean-Christophe Prévotet, Maryline Hélard

Efficient Modelling of FPGA-based IP Blocks using Neural Networks

Jordane Lorandel, Jean-Christophe Prévotet and Maryline Hélard
Institute for Electronics and Telecommunications of Rennes (IETR) - INSA of Rennes
jordane.lorandel@insa-rennes.fr

***Abstract*—Power consumption has become one of the most important concern in the embedded systems' community and being able to accurately and quickly estimate power consumption constitutes a challenging task. In this paper, an innovative and efficient technique for modelling signal activities and power consumption of FPGA-based hardware IP blocks is presented. We use two neural networks to model both the power consumption and the output signal activities of hardware IPs that compose a global system. These models are built according to estimated timing activities, which can be performed by a dedicated low-level tool. Our approach has the same objective as this type of tool while achieving a significant speed-up factor and enabling high-level power estimations. Moreover, we aim at directly estimating an IP-cascaded system's power consumption, at high-level. The effectiveness of the proposed approach is demonstrated through several case studies on specific hardware blocks for FPGA devices.**

I. Introduction

Power consumption constitutes a main issue in system design. Although obtaining the highest level of performance still remains the main purpose of designers, the reduction of the energy consumption has also become a critical problem. The fact is that such systems are usually embedded and often rely on batteries as unique power supply source. This is especially true for Field Programmable Gate Array (FPGAs) devices that have a flexible architecture enabling to implement a myriad of designs. Nowadays, these circuits have become an efficient solution for developing low-cost and high-performance embedded systems. Due to their technology, they constitute a viable alternative to their Application Specific Integrated Circuit (ASIC) counterparts or to processors such as Digital Signal Processors (DSP) and General Purpose Processors (GPP).

FPGA designs processes usually follows a top-down approach from System-level to bitstream generation. From a power consumption point of view, it is clear that decisions taken at system level will have the most important impact on the overall performance. Unfortunately, at this level, accurate implementation details are generally absent and may lead to poor estimation results. It is then difficult to conjugate both accurate power estimations and early decisions at high level.

Today, most tools proposed by FPGA vendors are based on power estimations performed at low abstraction levels e.g. XPower Analyzer from Xilinx. Although these tools are quite accurate, they still require a significant amount of simulation time, especially for large and complex designs. Furthermore, this type of tools is generally used by hardware engineers with a strong experience in hardware design. At this level, expensive re-design costs can occur if the power budget is not met.

In this paper, we focus our study on SRAM-based FPGAs that consist of a matrix of configurable logic cells. In these devices, sources of power consumption may be either static and dynamic. Static power refers to the power consumed in the circuit when it is powered on but without any signal activity (there is no clock activity in hardware). This power is directly related to the leakage currents and then depends on the technology of a specific device, the temperature and the voltage. For this reason, we have decided to only focus our study on dynamic power, since this power may be optimized all along the design process. The dynamic part of the power consumption for a circuit can be expressed as:

$$P_{dyn} = \frac{1}{2}\ V^2\ f\ C_{eff}\ \alpha \tag{1}$$

where V is the power supply voltage, f the operating frequency, C_{eff} the effective circuit capacitance and α represents the average number of signal switches per clock period.

The most difficult metric to evaluate is the switching activity α as it is a pattern-dependent problem [1]. It is clear that the switching activities of control signals and data have a direct impact on the power consumption. Two input patterns may lead to different switching activities and power consumption.

This paper is organized as follows: Section II describes the previous works that deal with power modelling using neural networks. Section III presents the proposed approach. Section IV details some results that have been obtained on several hardware IP, using our approach. Finally, a conclusion and future works are provided.

II. Related works

Neural networks have proved to be very efficient classifiers and estimators in a lot of domains. They enable to model complex non-linear problems into a simple structure that can be computed very fast compared with other algorithms.

From our point of view, related works can be categorized regarding the level of accuracy of the data that are used to feed the neural networks. First, there exist high-level approaches that feed neural networks based on general quantities such as an approximated number of gates or a clock frequency, without any knowledge on how the system is implemented. Second, low-level approaches require most of the design implementation steps to be completed. In this context, a lot

of hardware details are then provided which enables accurate power and switching activity estimations. Both approaches are described in the following sections.

A. High-level approaches

A common high-level approach is the use of spreadsheets that have been developed by FPGA vendors. This approach allows designers to obtain a first global estimation of the power consumption. These high-level estimators are based on approximated numbers that are provided by the designers prior to any implementation steps. Designers provide parameters values such as the clock frequency, the amount of hardware resources or thermal information. An average power consumption value is then obtained. XPower Estimation (XPE) is an example of spreadsheet for Xilinx FPGA devices [2].

In [3], a neural network is used to model the power consumption from sample data that are provided by XPE, based on Xilinx's spreadsheet. An improved back-propagation algorithm is used to train the network. An average error of 0.12% is shown in comparison to the XPE spreadsheet approach. Analytical power models can also be created to the estimate the IP power consumption in function of parameters of interest [4]. However, such models are usually less accurate but increase the flexibility.

B. Low-level approaches

At low-level, FPGA vendors have also contributed to the development of dedicated tools such as XPower Analyzer (XPA) from Xilinx or PowerPlay (PP) from Altera. Figure 1 illustrates how to obtain accurate power estimation at low level by using XPA or PP. First, it can be noticed that XPA is used after the place and route step where a low-level VHDL model that includes timing information is generated. A timing simulation of this model is then performed. From this simulation, the internal signal activities of the IP is recorded in a dedicated file (.saif or .vcd). These files include specific switching information that will lead to the most accurate power estimation. Finally, XPA estimates the power consumption from the analysis of the activity file, the design netlist (.ncd) and a physical constraint file (.pcf). If activity rates are not provided (by a simulation file), a vector-less estimation algorithm is used. It assigns default activity rates at the inputs of the design (usually 12.5%) and propagates them into the entire design until the outputs are reached. In other words, activity rates of the inputs are propagated throughout the circuit until the outputs are reached in order to obtain an efficient estimation. Under XPA, designers have the possibility to modify the activity values for a specific input signal. The switching activity of a signal is defined by a couple of values: the first is the signal rate (ranging from 0 to 100) that corresponds to the number of switches during a clock period. It is also necessary to provide a specific value that indicates the time during which the signal is at a high logic level (% High).

A lot of techniques have been developed to address this switching activity estimation. Most are based on statistics and require simulations to be performed [5]. Other approaches are based on probabilities [6]. Transition probability of input vectors, spatial correlation of bits within an input pattern [7] represent the information that can be exploited to propose new estimation models. However, such techniques cannot be easily applied to complex IPs and systems. An approach of macro-modelling for digital non-sequential IP has been proposed in [8]. This approach consists in building tables (LUT) that link power to input/output signal statistics. An average error of 1.94% is achieved.

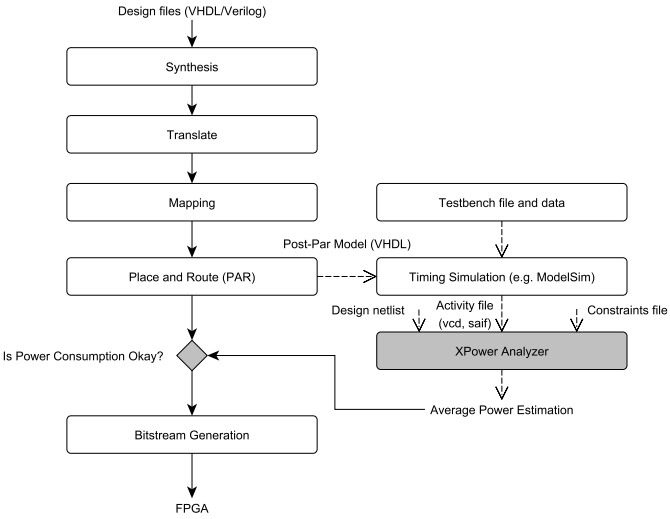

Fig. 1. XPower in a FPGA design flow

From our knowledge, there are currently no other works making use of neural networks to simultaneously model power consumption and switching activities of large and complex FPGA hardware elements such as IP (Intellectual Property). Moreover, switching activity estimation using macro-models or power models has only been applied to fine-grain components such as adders or multipliers.

III. Proposed approach

In our work, we aim at proposing a high level tool that takes into consideration results that have been obtained at low level (after the place and route steps). This makes it possible to perform simulations of high level models in a very fast way while obtaining a good level of accuracy in terms of power estimation. The model that is proposed in this paper is based on a neural network whose role is to determine the dynamic power consumption of a specific hardware block based on the activity of its input signals. Furthermore, an additional network is proposed to predict the signal activity of the outputs directly from its inputs. This is mainly used to propagate the activities throughout all the models that are representative of the different elements of a circuit.

Our contributions can be summarized as follows:

- coarse grain modelling of entire hardware IP using neural networks,

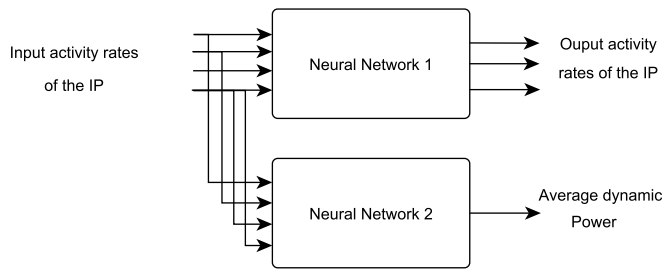

Fig. 2. IP models for dynamic power and signal activity estimation

- propagation of signal activities from model to model to refine power estimation by considering the pattern-dependent problem.
- fast and accurate power and signal activity estimations at high level

In this work, we assume that the considered architectures may be seen as a set of interconnected IPs modules that are implemented in a FPGA. These IPs may be custom built or directly taken from a vendor's library but share the same generic interface allowing them to be easily interchangeable. This interface is fully compatible with the AXI interface (Advanced eXtensible Interface) that enables to share information among several IPs according to a specific protocol. Note that the AXI interface is spread among the main FPGA vendors and compatible with most devices.

The proposed approach described in this paper first aims at providing very fast and accurate power and activities estimations of each hardware IP that constitutes the system. Second, this estimation is exploited at higher level in order to evaluate the overall power of the architecture by taking into account the activity of internal signals. Finally, we aim at directly estimating an IP-cascaded system's power consumption by executing these neural network estimator at high-level.

Our approach is described in Figure 2. Dynamic power consumption and output signal activities are modelled using two distinct neural networks. The first neural network determines the average dynamic power consumption according to the signal rates as well as the percentage of time of its inputs during which they are at a high logic level. The second network estimates the average signal rates of its outputs when applying the same inputs. In a typical data flow architecture, this is mainly used to cascade results to subsequent IPs models while guaranteeing realistic values of the switching activities throughout the circuit.

In our approach, we assume that a dedicated library is made available to designers. This library contains RTL-code of the hardware IPs as well as their power estimates. The neural networks model are also included in the library. This allows designers to built their system very fast and to speed-up modelling and design re-use.

The proposed approach is based on two major stages: an IP characterisation phase and a high-level system modelling phase.

A. IP characterisation

In the characterisation phase, each IP is fully implemented in the targeted FPGA family. Design implementation is performed following the synthesis, mapping and place and route steps. After implementation, a low level power analyzer (XPA or PP), is used to record the average dynamic power consumed by the IP. The results obtained by the tool not only take into account the nature of the circuit (number of LUTs, power supply, type of FPGA, etc.) but also the activity of the IP input signals and their probability of occurrence. Basically, if no timing simulation is performed to record the IP internal activity, a vector-less estimation algorithm is used, as described in section II.

Under XPA, it is possible to directly define the switching activity of the inputs signals using a setting file (.xpa). Consequently, we automatically generated a set of 10000 files in which input signal rates of the IP have been randomly defined. Then, a vector-less estimation algorithm has been run and results have been stored in dedicated files. This is automatically performed to determine 1) the average activity rates of the IP outputs, 2) the average dynamic power. The obtained results are then used to build representative examples to elaborate our models.

At the end of this phase, we obtained the power consumption and activity rates related to the outputs of the considered IP, with respect to the randomly-defined activity rates of the inputs. Note that this characterisation phase is only performed once for each IP and for each device of a FPGA family. A complete library of IPs is made available that may be used to build global systems very rapidly.

B. Neural Architecture

The two neural networks that have been described in Figure 2 consist of two independent multi-layer perceptron (MLP) neural networks.

The structure of these networks is depicted in Figure 3. In the forward phase, the hidden layer weight matrix is multiplied by the input vector $X = (x_1, x_2, x_3, \ldots, x_n)^T$ to compute the hidden layer output:

$$y_{h,j} = f(\sum_{i=1}^{N_i} w_{h,ji} x_i - \theta)$$

where $w_{h,ji}$ is the weight connecting input i to unit j in the hidden neuron layer. θ is an offset termed bias that is also connected to each neuron. The function f is a non linear activation function. In this work, the classic S-shaped sigmoid function is used for neurons in the hidden layer:

$$f(\alpha) = \frac{1}{1 + e^{-\alpha}}$$

For the output layer neurons, classic linear activation functions are used. The inputs of both models represent the activity rates of the IP inputs, which are the same inputs as in XPA.

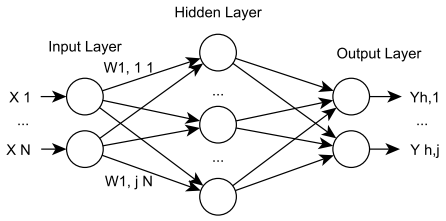

Fig. 3. Structure of a Multi-Layer Perceptron

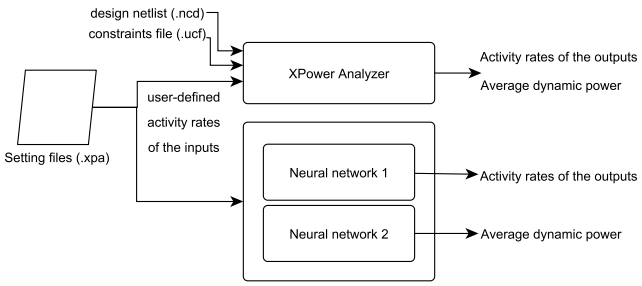

Fig. 4. Comparison of our approach against XPA

Several structures of MLPs have been studied but a simple network with one hidden layer has been retained (for both networks) since it provides a good level of performance (fast convergence) and reduces the computing time. In the power model, only one output has been considered which represents the estimate of the IP dynamic power. In the activity model, the number of outputs corresponds to the switching activity rates of each IP output signal and then differs for each model.

C. Training, tests and validation

In order to build the different bases that are required to train, test and validate the network, we have used the 10000 random activity patterns representing the activity rates of all inputs bits. Classically, 70 % of these patterns were used to train the networks. 15 % were used to test the network and 15 % were used to constitute the validation set. A simple back-propagation algorithm has been used to train the network and determine the optimal values of the weights. As in [3], we used Matlab to easily train, test and validate the neural networks.

As described in Figure 4, a comparison between the results that are provided in XPA and our approach is realised. This comparison is performed on the 10000 random activity patterns.

IV. Results

We have applied the characterisation process on several hardware IPs. In this paper, we mainly focus on IPs that constitute the baseband processing of wireless communication applications. To illustrate the validity of our approach, we have modelled several configurations of an Inverse Fast Fourier Transform (IFFT) hardware block. This block is widely used to realize OFDM modulation in current wireless systems. IFFT configurations that were implemented are detailed in Table I. We have implemented the xfft_v7_1 core from Xilinx on a Virtex-6 LX240T FPGA using ISE 14.4. It can be noted that data quantization is changed from 8 bits to 18 bits. As a consequence, the complexity in terms of the number of used resources for the considered design evolves as a function of the data quantization.

TABLE I
IFFT CONFIGURATIONS

Parameter	Value
Transform size	256 pts
Architecture	Pipelined streaming
Data quantization (input data and phase factor widths)	[8,10,12,14,16,18]bits
Scaling / Rounding	Scaled / Truncation
Output ordering	Natural / Cyclic Prefix insertion

For each characterisation phase, the number of neurons that constitutes the hidden layer was arbitrary set to 20.

A. Model Accuracy

Table II shows the accuracy of the models that have been obtained using both neural networks. The model accuracy has been evaluated by computing the Mean Square Error (MSE) and the average absolute error between the target outputs and the actual outputs. According to this table, it may be noticed that the power estimation error is very low (in the mW range) for IPs that generally consume from few to hundreds of milliwatts. In fact, the range of power consumption values directly depends on the choice of the FPGA target and the IP complexity.

The MSE (that has been computed for the activity rates) is also very low (it ranges from 3.1e-2 to 0.15) which leads to very accurate estimations of the signals' activity. This means that our approach achieves nearly the same accuracy that the low-level XPA tool. From this consideration, it makes it possible to propagate these results to subsequent IP modules without degrading the overall estimation. Regarding the power estimations that have been obtained, the MSE goes from 5.7e-2 up to 0.46. The average absolute error is also very low as it is lower than 0.46 mW and 0.17 for the power and the activity respectively.

B. Simulation Acceleration Factor

The main interest of the proposed approach is to considerably accelerate the simulation time as compared to classic approaches. In fact, the objective is to allow designers to built a complete system simply by interconnecting several neural networks at high level. This point underlines our choice of interface and the need for the signal activities propagation through all models. Moreover, the use of models based on neural networks avoid the development of the entire system and saves time.

To clarify the interest of our approach, let us imagine that a designer wants to replace the first block with a custom

TABLE II
MODEL ACCURACY AGAINST XPOWER ANALYZER

Configuration		Mean Square Error	Mean Absolute Error	Range
8 bits	Power*	5.07e-2	0.14	[24-82] mW
	Activity	3.1e-2	0.1	[0-100]
10 bits	Power*	0.109	0.201	[26-95] mW
	Activity	7.76e-2	0.14	[0-100]
12 bits	Power*	0.34	0.39	[28-123] mW
	Activity	4.9e-2	0.13	[0-100]
14 bits	Power*	0.15	0.25	[29-145] mW
	Activity	5e-2	1.3e-3	[0-100]
16 bits	Power*	0.38	0.4	[40-185] mW
	Activity	0.14	0.16	[0-100]
18 bits	Power*	0.46	0.46	[35-203] mW
	Activity	0.15	0.17	[0-100]

* Dynamic power without inputs/outputs (IOs) power

TABLE III
ESTIMATION SPEED COMPARISON

IFFT config.		XPower Results	Proposed Model	Speed up factor
8 bits	Power	23s	20ms	11500
	Activity			
10 bits	Power	24s		12000
	Activity			
12 bits	Power	28s		14000
	Activity			
14 bits	Power	32s		16000
	Activity			
16 bits	Power	35s		17500
	Activity			
18 bits	Power	41s		20500
	Activity			

hardware IP in a system that is made of 4 consecutive blocks. Consequently, activity rates are going to be modified due to the change of the first hardware IP. Moreover, the 3 subsequent blocks also have to be re-evaluated separately in order to consider the modification of the activity rates at the output of the first block (the power consumption as well). Using the classic XPA approach, a complete evaluation has to be performed for every subsequent IP (inputs activity has to be provided in XPA and a simulation has to be run to generate new output activities). Once all the input/output activity rates of every IPs are known, global power consumption can be evaluated. This requires several steps and an additional simulation time that is often prohibitive.

In our approach, input and output activity rates are directly propagated from one neural network to the subsequent ones thanks to a common interface. Only one computation is required to evaluate both output activity rates and the related power consumption of the overall system. Designers only have to know the switching activity rates of the input pattern. This makes it possible to obtain much faster results.

These results are even more significant when designers want to test several configurations of a complete scheme (for example, when different FFT sizes are explored or several data quantizations need to be tested). In this case, our approach only consists in replacing the corresponding IPs models and running a full simulation.

It can also be noticed in Table III that the time that is required to estimate power or activity with XPA is increasing as the complexity of the IP grows. For the different configurations of the IFFT, XPA requires an estimation time ranging from 23s to 41s. In our approach, since neural networks have a simple structure, they can easily be computed in software. As a consequence, the time to estimate the power or the activity of each IP is nearly constant (around 20 ms), regardless of the IP complexity. Finally, it can be seen that a speed-up factor between 11500 and 20500 has been obtained.

V. CONCLUSION

In this paper, a new FPGA-based approach for IP power consumption and signal activity modelling has been presented. The approach is based on neural networks that aim at modelling the relationship between outputs signal activities (or power consumption) and input signal activities. Our technique achieves the same accuracy as a low-level power estimation tool but with a huge acceleration factor in terms of computation speed.

We have shown that our approach enables to explore the design space very efficiently and fast. Designers will only have to build their own architecture by connecting a set of predefined models (where each model consists of two neural networks) and perform a global simulation. A simple parameter change is almost instantaneous and does not imply to re-evaluate each IP that builds the systems, as in classic approaches.Therefore, using our approach makes it possible to test thousands of configurations in a minimal time.

As future works, we will continue to enrich the existing library with new IP power estimates and target other applications domains as well as other architectures (like other FPGA families or ASICs). We will also try to investigate some more generic models that can reduce the learning process and thus facilitate the building of libraries.

REFERENCES

[1] F. N. Najm and M. G. Xakellis, "Statistical Estimation of the, Switching Activity in VLSI Circuits," *VLSI Design*, vol. 7, no. 3, pp. 243–254, 1998.

[2] Xilinx Inc., "XPower Estimator," Tech. Rep., April 2014, uG440 (v2014.1). [Online]. Available: {www.xilinx.com/support/documentation/sw_manuals/xilinx2014_1/ug440-xilinx-power-estimator.pdf}

[3] G. Zhou, B. Guo, X. Gao, J. Ma, H. He, and Y. Yan, "A FPGA Power Estimation Method Based on an Improved BP Neural Network," in *2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, Sept 2015, pp. 251–254.

[4] D. Elleouet, N. Julien, and D. Houzet, "A high level SoC power estimation based on IP modeling," in *Proceedings 20th IEEE International Parallel Distributed Processing Symposium*, April 2006, pp. 4 pp.–.

[5] E. Todorovich, E. Boemo, F. Angarita, and J. Vails, "Statistical power estimation for FPGAs," in *Field Programmable Logic and Applications, 2005. International Conference on*. IEEE, 2005, pp. 515–518.

[6] H. Hassan, M. Anis, and M. Elmasry, "Total Power Modeling in FPGAs Under Spatial Correlation," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 17, no. 4, pp. 578–582, April 2009.

[7] J. Lamoureux and S. J. Wilton, "Activity estimation for field-programmable gate arrays," in *Field Programmable Logic and Applications, 2006. FPL'06. International Conference on*. IEEE, 2006, pp. 1–8.

[8] Y. Durrani, T. Riesgo *et al.*, "Power macromodeling for IP modules," in *Electronics, Circuits and Systems, 2006. ICECS'06. 13th IEEE International Conference on*. IEEE, 2006, pp. 1172–1175.