RTL to Transistor Level Power Modeling and Estimation Techniques for FPGA and ASIC: A Survey

Yehya Nasser, Jordane Lorandel, Jean-Christophe Prévotet, and Maryline Hélard

Abstract—**Power consumption constitutes a major challenge for electronics circuits. One possible way to deal with this issue is to consider it very soon in the design process in order to explore various design choices. A typical design flow often starts with a high-level description of a full system, which imposes to provide accurate models. Power modeling techniques can be employed, providing a way to find a relationship between power and other metrics. Furthermore, it is also important to consider efficient power characterization techniques. The role of this article is, first, to provide an overview of the register transfer level to transistor level power modeling and estimation techniques for FPGAs and ASICs devices. Second, it aims at proposing a classification of all approaches according to defined metrics, which should help designers in finding a particular method for their specific situation, even if no common reference is defined among the considered works.**

Index Terms—**Application specific integrated circuit (ASIC), field programmable gate array (FPGA), high-level power estimation, power consumption, power estimation, power modeling, tools.**

I. Introduction

With the imminent arrival of 5G and Internet of Things (IoT), a lot of electronic devices will be able to communicate and share data between them, involving machine to machine or machine to vehicular communications for example. All sectors are concerned, starting from the industries to the agriculture, telecommunication, health, etc. Human activities and technologies have a significant impact of the worldwide carbon footprint. It has been shown that cities cover 2% of Earth's surface but consume up to 78% of the world's energy [1]. In the same time, it has been shown that developing smart and energy-efficient technologies may be an efficient solution to drastically reduce the energy cost and the environmental impact. These electronic devices are generally designed using a very large-scale integration (VLSI) process that consists in building an integrated circuit (IC) by linking millions of transistors on a chip. All complex systems and communication devices are based on VLSI, including analog ICs, such as sensors and operational amplifiers as well as digital ICs, such as microprocessors, digital signal processors (DSPs), and micro-controllers. In addition to this, VLSI design covers the development of application specific ICs (ASIC) and programmable logic (PL) Devices.

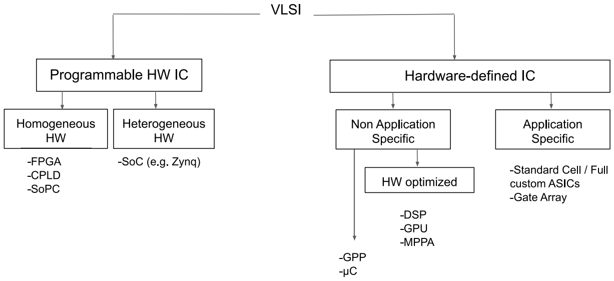

Fig. 1. Taxonomy on VLSI ICs.

In this article, we propose a taxonomy that is illustrated in Fig. 1. This classification tends to facilitate the comparison of many circuits according to their hardware architectures. Two main VLSI categories have been identified: 1) hardware-defined and 2) programmable hardware devices.

Hardware-defined devices can be then divided into application-specific (e.g., ASIC) and nonapplication specific ICs. We further divide the nonapplication specific devices into hardware-optimized for which GPU is an illustrative example and general ICs, including general purpose processor (GPP) or micro-controllers. Regarding programmable hardware ICs, two categories have been defined depending on if the device is built from homogeneous hardware, e.g., field programmable gate array (FPGA) or heterogeneous resources. A system on chip (SoC) is a typical example of heterogeneous hardware ICs that contain a hard processor in combination with logic into the same chip.

Nowadays, FPGAs are used in many applications, from data-centers to low-power smart embedded devices. Depending on the application requirements, such devices can now support complex designs, allowing fast prototyping, and reducing time-to-market. FPGAs are popular in many sectors such as telecommunications, robotics, automotive, etc.

A major feature of FPGAs is reconfiguration. As a consequence, FPGAs consume much more power than their ASIC counterparts as additional transistors are used to maintain the reconfiguration plan. Note that several FPGA technologies

Manuscript received January 6, 2020; revised April 15, 2020; accepted June 6, 2020. Date of publication June 18, 2020; date of current version February 19, 2021. This article was recommended by Associate Editor W. Zhang. *(Corresponding author: Jordane Lorandel.)*

Yehya Nasser, Jean-Christophe Prévotet, and Maryline Hélard are with IETR, INSA Rennes, 35700 Rennes, France (e-mail: yehya.nasser@outlook.com; jean-christophe.prevotet@insa-rennes.fr).

Jordane Lorandel is with ETIS, Cergy-Pontoise Université, ENSEA, 95014 Cergy-Pontoise, France (e-mail: jordane.lorandel@cyu.fr).

Digital Object Identifier 10.1109/TCAD.2020.3003276

exist, such as SRAM or FLASH. The Flash technology is dedicated to low-power applications for specific segments of the market whereas SRAM technology is most common.

In this article, we tackle the issues related to power modeling and estimation of both FPGAs and ASICs. In particular, we focus on CMOS and SRAM technologies, and do not consider the flash technology that is specific. Several surveys have already reviewed associated works [2]–[5] but none of them provides a way to compare power estimation and modeling techniques between them. This prevents designers to rapidly identify the appropriate methods according to their constraints. Moreover, this article focuses on register transfer level (RTL) to transistor level power modeling and estimation techniques, rather than system-level. Interested readers could refer to [6] for higher level techniques as transaction-level power modeling.

The objectives of this survey are as follows.

1) To deliver a comparative study, including power measurement methods, commercial power estimation tools, and power modeling techniques. The comparison is performed based on custom selected metrics.
2) To help designers in identifying the most appropriate technique to estimate/measure the power consumption of their FPGA or ASIC design.

This article is organized as follows: we first propose a background on ASIC and FPGA devices in Section II. Then, related works on power measurement techniques are given in Section VI. Section VI presents various power modeling approaches. We finally present a comparison between them and conclude.

II. Hardware Platforms

Before addressing the power issue for ASICs and FPGAs, the following sections provide an overview of the hardware architecture of each devices.

A. ASIC Architecture

ASICs consist of ICs that are specifically built for predefined functions. They can be classified into two categories: 1) full-Custom and 2) semi-custom ASICs.

Full-Custom ASICs are customized and optimized circuits that offer the highest performance and the smallest die size. The design time is thus very long (up to several years for production) and the development costs are very high. An alternative to full-custom ASICs is denoted semicustom ASICs and helps designers to shorten the design time, to reduce costs, and to automate the design process. This technology is based on a regular organization of homogeneous cells or the integration of standard cells already defined in a library.

B. FPGA Architecture

An FPGA architecture consists of a 2-D array of configurable logic cells, inputs/outputs, and interconnection between them, as illustrated by Fig. 2. The basic internal structure of a logic cell generally consists of look-up tables (LUTs) implemented as memories. Each n-input LUT can implement any function with up to n variables and consists of 2^n SRAM bits that store the required boolean truth table. In addition, there is at least a multiplexer and a flip-flop within each cell. Each FPGA vendor proposes its own implementation of logic cells. They also may differ regarding the layout of their interconnect and the architecture of their configurable elements. FPGA devices can also integrate specific elements, such as DSPs blocks, RAM memories, and frequency synthesizers (PLL), as well as more sophisticated elements, such as PCIe hardware controller, high speed communication transceivers etc.

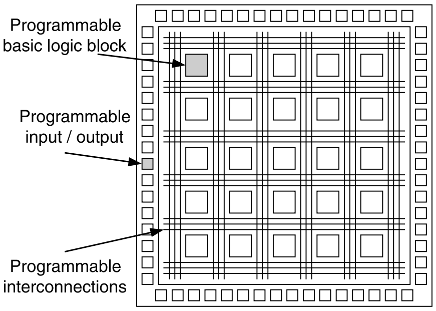

Fig. 2. Basic FPGA architecture [7].

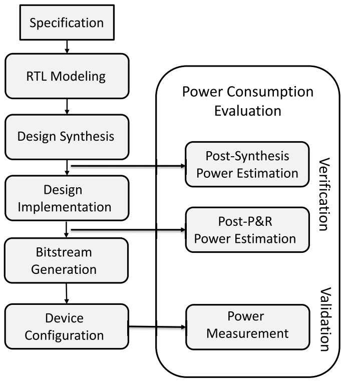

Fig. 3. FPGA design flow.

C. Power Considerations in the Hardware Design Flow

Power consumption can be estimated using dedicated tools or simulations at different steps along the design flow as indicated by Fig. 3. Right after design synthesis, power can be estimated using the resource number information coming from the synthesis tool and by taking into account the estimated timing. The Post-P&R power estimation takes into account the physical implementation details, including routing delays, [8] so that timing information is more realistic. Finally, power measurements can be realized either on FPGAs or ASICs after implementation.

In this regard, the gap between FPGA and ASIC in terms of power consumption becomes narrow [9]. More specifically, it is possible to deal with the dynamic power consumption on both platforms using the same methods, as demonstrated in [10]. As for the static power in ASICs and in FPGAs, it is design dependent, as partially shown in [11] for the FPGA case.

III. Power Consumption in Digital Circuits

A. Static Power

Static or leakage power consumption corresponds to the power that is consumed when the circuit is powered but not active, meaning that transistors are not switching. As technology advances, this power is becoming non negligible due to the shrinking of transistors' size as well as the thickness of the oxides. Three fundamental currents contribute to static power. The first one is the gate-oxide leakage current that occurs between the transistor's channel and gate. The second is the subthreshold leakage current that occurs between the transistor's source and drain. The last contributor is the reverse bias current, which is located between the transistor's drain and the substrate. When considering high-k dielectric devices, the main source of static power remains the subthreshold leakage current I_{leakage} [12] and it can be expressed as

$$I_{\text{leakage}} = I_0 e^{\frac{-qV_{\text{th}}}{\beta k_B T}} \tag{1}$$

where:

- I_0 denotes a constant that depends on the dimension and the fabrication technology of the transistor;
- β is a technology-dependent factor;
- k_B stands for the *Boltzmann* constant;
- T represents the temperature;
- q represents the charge of the electrical carrier;
- V_{th} represents the threshold voltage.

Depending on a given technology, temperature, and threshold voltage have an exponential impact on leakage currents and thus on static power since $P_{\text{static}} = V_{dd} I_{\text{leakage}}$, where V_{dd} represents the supply voltage.

B. Dynamic Power

In current digital circuits, dynamic power remains the main source of power consumption. This source of power consumption is generated by the transistors' switching activity when the circuit is active. When considering a basic CMOS cell, each logic gate is made of two types of transistors, i.e., N-MOS and P-MOS. The total power P_t consumed for charging and discharging the load capacitance C_L within a cycle can be computed using $P_t = [(C_L V_{dd}^2)/t]$ where C_L denotes the load capacitance, V_{dd} represents the supply voltage, t denotes the total time, which equals to $k.T_{\text{clock}}$, k corresponds to the number of cycles, $T_{\text{clock}} = 1/F_{\text{clock}}$. For a transistor that toggles N times over a time interval t, the corresponding power consumption can be hence modeled as in (2)

$$P_{\text{dynamic}} = NP_t = N\frac{C_L V_{dd}^2}{t} = N\frac{C_L V_{dd}^2}{k.T_{\text{clock}}} = \alpha C_L V_{dd}^2 F_{\text{clock}} \tag{2}$$

where α denotes the average number of switching per cycle. It is also called the activity factor, and can be expressed as $\alpha = (N/k)$.

A small amount of P_{dynamic} is due to a short circuit current that appears during the switching of both P-MOS and N-MOS transistors, due to the fact that input data stimuli signals typically are not sharp signals. Consequently, during a small period of time, both N-MOS and P-MOS transistors are turned on simultaneously. This current flows between the supply voltage and the ground, leading to a short-circuit, that can contribute to up to 10% of the total dynamic power consumption [13]. This short circuit power, P_{sc}, can be evaluated using

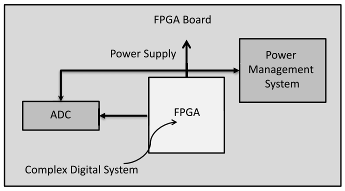

Fig. 4. On-board power measurement system.

$$P_{\text{sc}} = K\tau F_{\text{clock}}(V_{dd} - 2V_{\text{th}})^3 \tag{3}$$

with K, a technology-dependent parameter, τ the charge/discharge duration, F_{clock} the clock frequency, V_{dd} the voltage supply, and V_{th} the threshold voltage [14].

Finally, the total dynamic power, P_{Dyn}, can be estimated using the following equation:

$$P_{\text{Dyn}} = P_{sw} + P_{\text{sc}}. \tag{4}$$

As a summary, the main parameters of both static and dynamic powers are now known. In the next section, we review the existing methods allowing to physically measure power using either on-board measurements or an external setup.

IV. Power Measurement Issues

The most intuitive way to evaluate power of devices consists in performing real measurements on the circuit directly. However, this requires to perform all design steps before obtaining any power consumption profiles [15]. Either on-board or on-chip sensors may be used to monitor metrics, such as the supply voltage and the current drawn by the FPGA or ASIC circuit. An external instrumentation setup has then be defined to properly evaluate power consumption.

A. On-Board and On-Chip Solutions

The on-board power measurement solution is a portable solution that can help the digital designers to evaluate their designs as shown in Fig. 4. This can be achieved using the available on-board components, like current sensors, and voltage regulators. In the following, we review the existing works that are related to this solution.

Recently, some development boards permit to perform voltage and current measurements at specific circuits locations. FPGAs manufacturers like Xilinx and Intel offered such solutions for real power measurements. For instance, Xilinx provides a solution with Texas Instruments (TIs) that helps designers in performing real power measurements on Xilinx boards [16]. This solution was based on several elements: 1) the built-in current sensors and 2) the power regulators with a JTAG to USB adapter that connects the board to the host PC to monitor the current and the supply voltage. However, this solution was considered as limited due to some hardware noise caused by the built-in regulators and sensors, by the

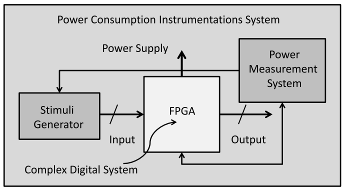

Fig. 5. External solution for power measurement.

sampling frequency limitation and by the low bit resolution of the on-board analog to digital converters (ADCs).

This leads to real problems in terms of measurement accuracy and measurement bench flexibility. These limitations are presented in the works of [17] and [18] that study the power consumption overhead of dynamic partial reconfiguration.

B. External Measurement Setup

To obtain more accurate power measures, an external power measurement setup is often considered as depicted in Fig. 5. A stable low-noise power source is usually required to supply the FPGA core power pins. External instrumentation is also needed to measure the current across a shunt resistor connected in series with the FPGA core. Some approaches use an additional control board that provides custom data to the FPGA so that users may have a full control of the input stimuli, and thus the input switching activity. This is used to perform an efficient characterization of power according to given data as in [19]. In this work, the authors isolate different power sources of the FPGAs (logic, clock, and interconnect) and perform measurements for 10 000 different vectors applied to the module under characterization. Unfortunately, measures are performed during a small time window due to the limited memory size of the FPGA platform used as a data source.

Oliver *et al.* [20] proposed a system that is based on a current-frequency conversion block that measures the power consumption of a running application in real-time. However, the authors present the power measurement methodology without mentioning the different types of noise that can be induced by the different electronic blocks and their effects on the accuracy and the accuracy of the physical measurements.

The work in [21] studied the switching activities of a set of internal signals and the corresponding power values to obtain a model with an external runtime computation. With the same idea, the work in [22] applied an online adjustment process and obtained higher accuracy. The hardware implementation of the signal monitoring in these works led to a resource overhead of 7% as well as an additional workload of 5% of CPU time.

Recently, Lin *et al.* [23] introduced a novel and specialized ensemble model for runtime dynamic power monitoring in FPGAs. Their design provides accurate dynamic power within an error margin of 1.90% of a commercial gate-level power estimation tool.

In addition, Oliver *et al.* [24] presented a framework to compare power values either given by a power estimation tool or by real measurements in a formal way. This work helps in identifying the power estimation tool accuracy rather than comparing the estimation techniques themselves.

V. Power Estimation Techniques

Power estimation techniques are very efficient alternatives to measurement-based methods as they do not necessary need a characterization phase allowing a fast design exploration. In fact, such techniques can be considered at different levels of abstraction of the design, leveraging the time needed before having power estimates. Three methods can be identified: 1) simulation-based; 2) probabilistic-based; and 3) statistical-based. In this article, we focus on low level power modeling and estimation techniques, ranging from RTL to lower implementation levels.

A. Probabilistic-Based Methods

Historically, many proposals have contributed to the probabilistic analysis of digital circuits. Probabilistic power estimation methods use data characteristics instead of the real data. They generally rely on the static probability and the transition probability of given signals. The static probability $P(s_i)$ of a signal s_i can be defined as the probability of this signal to have a HIGH logic value. The transition probability $TP(s_i)$ of a signal represents the probability that this signal changes its state from a logic HIGH to a logic LOW or *vice versa*. Probabilistic-based methods propagate these values throughout the nodes and gates of a given circuit (or netlist) to obtain a global power estimation. This bypasses the use of the simulator.

Tsui *et al.* [25], Marculescu *et al.* [26], Monteiro *et al.* [27], and Najm [28] make use of these probabilistic methods. These are characterized by a good estimation speed, which makes them faster than simulation-based methods. Although the estimation speed is very important, the estimation accuracy is also crucial.

Tsui *et al.* [29] and Marculescu *et al.* [30] proposed additional metrics to be taken into account in order to improve the accuracy of the probabilistic techniques, such as the temporal and spatial signal correlations. The signal spatial correlation takes place when a bit value of an input depends on another input bit. A signal temporal correlation occurs when the bit value of an input bit depends on the previous bit value of the same input. Schneider and Krishnamoorthy [31] showed that the signal correlations may affect estimation results. For instance, neglecting the temporal correlation increases the estimation error from 15% to 50%. Discarding spatial correlation also degrades the error from 8% to 120%. Note that power estimation focuses on the propagation of transition density and static probability. In this regard, the discussed techniques assume that there is no more than one transition at the same time. In other words, glitches are not taken into account. Generally, these techniques only use deterministic delay models, meaning that gates are modeled with simple constant delays. This is a severe limitation since delay fluctuations and uncertainties may occur and have a significant impact on power consumption.

To counteract this issue, Garg *et al.* [32] suggested an improvement to propagate the transition density of data signals and to take into account the uncertainty of the delay models. The probability of the data signals is then described as a continuous function of time, which provides more accurate results compared to the fixed delay models.

Although, there are no current work dealing with probabilistic-based approaches, these solutions are still in use and integrated in significant power estimation tools [33].

B. Simulation-Based Methods

Simulation-based power estimation is used by most of the computer-aided design tools. This type of technique consists in applying data stimuli to the inputs of the design under test and to perform a simulation to determine the corresponding outputs. Depending on the abstraction level, the type of information that is required to obtain power estimation is different, going from current and voltage values, capacitance, and clock frequency to the switching activities of all signals.

Many simulators can be used to obtain the required information. A famous transistor-level simulator is a SPICE. It uses large matrix solutions of Kirchhoff current law (KCL) equations to determine nodal currents at transistor level [34]. In this work, basic elements, such as resistors, capacitors, inductors, current sources, voltage sources, and higher-level diode and transistor device designs are used to correctly predict the current and voltage drop. Although highly precise, these tools become quickly unpractical as the size of the circuits increases.

Another transistor level simulator called PowerMill [35] uses linear piece-by-piece transistor modeling to store transistor characteristics in lookup tables. It also uses an event-driven timing algorithm to reach speeds comparable to logic simulators. The difference with other approaches is that it does not consider logic transitions but rather changes in node voltages. Using lookup tables leads to inaccuracy, but results are provided 2 to 3 times faster compared to SPICE. Although, PowerMill was introduced more than 2 decades ago, there are still some works that make use of it [36].

Gate level simulation includes the use of logic parts, such as NAND/NOR gates, latches, flip flops, and interconnection networks. The most popular technique of assessment includes an event-driven model [37]. When an event occurs at a gate input, it may generate an output event after a simulated time delay. Power consumption is predicted by computing the charging/discharging capacitance at the gate and by evaluating the activity of this node [38].

The main advantages of simulation-based power estimation techniques are their precision and their genericity. However, these methods also have some drawbacks. First, they generally require large memory resources to store all signals' information. Second, they often need a very long simulation time. Third, it is a size-dependent technique since the estimation depends on the size of the simulated circuit (number of gates, inputs, outputs etc.).

C. Statistical-Based Methods

Statistical power estimation techniques are used to obtain the power consumption of a given design, after defining random input stimuli that are applied to the primary inputs of a given circuit. Then, the design is simulated using a power simulator until a desired precision is achieved.

One of the first study can be found in [39]. Authors present McPower, a Monte Carlo power estimator, in which the simulation is stopped when sufficient precision is achieved according to a specific level of confidence. Note that this technique is also time consuming but delivers faster results in comparison to simulation-based methods.

Xakellis and Najm [40] introduced an effective statistical sampling technique to estimate individual node transition densities. They also classify nodes into two categories: 1) regular and 2) low transition density node. Regular-density nodes are certified with user-specified percentage error and confidence levels whereas low-density nodes are certified with an absolute error, with the same confidence. This speeds convergence while sacrificing accuracy only on nodes that have a small contribution to power. This technique has been enhanced in [41] by using distinct error values for distinct nodes. For nodes that often switch, error levels are evaluated more accurately.

More recently, in [42], the meta-modeling approach is adopted. In this work, two different statistical models are extracted to estimate power dissipation for individual intellectual property (IP) core and bus in the design. For an entire SoC, the average power is extracted by a simple addition of all power estimation results of these two models. In experiments, the average error is 11.42%. Verma *et al.* [43] applied statistical methods to estimate power of low-power embedded systems. Around 30 digital circuits are synthesized using Xilinx Synthesis Tool and power is visualized using Xpower Analyzer.

D. Power Estimation Tools

Power estimation tools are built on methods that have been described in the previous sections. For instance, Synopsys offers tools such as PrimePower [44] that aims at accurately analyzing the power of a full-chip of cell-based designs, at various stages of the design process.

Cadence proposes Genus as a power estimation tool working at both RTL and gate-level [45]. The Genus RTL power tool provides time-based RTL power profiles with a system-level runtime, along with high-quality estimates of gates and wires. Note that both tools offer vector-free (or also called probabilistic-based) peak power and average power analysis. They also both deal with vector-based (or also called simulation-based) analysis. Power estimation is then based on a detailed power profile of the design that takes the interconnect, the signals' switching activity, the nets' capacitance, and the cell-level power behavior data into account.

In FPGAs, power estimation is usually evaluated at high level using spreadsheets which are usually specific to a device. An example of a vendor's tool is the Xilinx power estimator (XPE) [46]. These tools typically aim at providing power and thermal estimates at an early stage of the design flow. In addition, Xilinx has developed the Xpower analyzer [47] to analyze power consumption at different levels of abstraction. Vector-free and vector-based power estimation are also supported for these devices [48]. In parallel, Intel provides PowerPlay [49] that include early power estimators. The Intel Quartus Prime software power analyzer also gives the opportunity to estimate power consumption at various design stages.

TABLE I
ESTIMATION TECHNIQUES ADVANTAGES AND LIMITATIONS

Estimation Techniques	State of the art	Advantages	Limitations
Probabilistic-based	[25], [26], [27], [28], [29], [30], [31], [32].	high estimation speed.	low accuracy.
Simulation-based	[34], [35], [36], [37], [38].	1) high accuracy; 2) generic.	1) large amount of memory resources; 2) low estimation speed.
Statistical-based	[39], [40], [41], [42], [43]	moderate accuracy.	moderate estimation speed.

A typical FPGA power estimation flow requires simulation at RTL using a dedicated simulator. The outputs generally consist of a (VCD) file to be provided to the power estimation tool. If vectors are not available, switching activities may be assigned to the inputs and propagated using an activity estimator such as ACE [50]. Academic FPGA tool flows such as VTR also has its own power estimator (VersaPower) [51] which relies on activities generated by ACE to perform power estimation.

Recently, a SPICE-based power estimation tool called FPGA-SPICE which is integrated with the tool versatile placement and routing (VPR) framework was presented by [52]. This tool can run SPICE level simulations for a given design mapped to an FPGA and provides cell-level power values or total full chip power as specified by users. The power values obtained using FPGA SPICE are more accurate as compared to VersaPower but the runtime is significantly longer and it is not scalable for large designs.

E. Summary

In this section, a summary of different power estimation techniques is proposed. Table I classifies the works of the state-of-the-art and discusses their advantages and limitations. First, simulation-based techniques have two important advantages: 1) high accuracy [2] and 2) generality [53]. Nevertheless, the simulation time is an important limitation of such technique since power estimation needs to wait until all current node waveforms are generated. In addition to this, significant memory resources are also required.

Probabilistic-based approaches deliver fast estimation. On one hand, they do not require waveforms generation since only signals and transition probabilities are used to estimate power. On the other hand, lower accurate results are usually obtained due to the use of simplified delay models for circuit components [2] and average signal probabilities (as compared to real input stimulus using simulations).

The statistical-based power estimation technique consists of a tradeoff between the accuracy of the simulation-based approach and the estimation speed of the probabilistic-based techniques [54]. To this purpose the estimation speed and accuracy is considered as moderate in the table.

VI. POWER MODELING TECHNIQUES

In this section, we review the power modeling approaches and classify them into four categories. These categories, illustrated in Fig. 6, can be summarized as follows: 1) analytical; 2) table-based; 3) polynomial-based; and 4) Neural Networks.

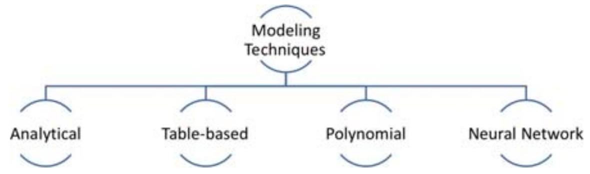

Fig. 6. Modeling techniques literature review.

From our analysis, some parameters that are common to all modeling techniques were identified and selected as important qualitative metrics for modeling.

1) The *modeling level* represents the level at which the model is intended to be used. More specifically, power consumption can be modeled for either a circuit or a component within a circuit.
2) The *modeling effort* evaluates the effort that is needed to build the model. Three levels are proposed for this metric: a) low; b) moderate; and c) high, represented as $*$, $**$, and $***$, respectively. A high modeling effort typically requires multiple iterations of the design flow whereas a low modeling effort could only require few simulations with abstract information to build the power model.
3) The *model granularity* represents the information level that is used to build the model. Two granularity models can be defined: a) fine grain and b) coarse grain. The first includes power models that need information at bit-level, such as the transition rate and the static probability, whereas the second includes more abstract information, e.g., number of operations, data width, etc.
4) The *characterization technique* represents power modeling methods are based on either estimated or measured power values and thus require a power characterization step. Table-based, regression-based, and neural networks approaches necessitate power characterization.

A. Analytic Modeling

Analytic techniques attempt to relate power consumption to the switching activity and the capacitance of a design [5]. More specifically, these techniques are based on the theoretical equation of the power dissipation for a CMOS transistor expressed in (2). Landman [3] divided this modeling type into activity-based and complexity-based techniques.

Complexity-based techniques tend to roughly estimate the capacitance from the design architecture. The major drawback of these techniques is that input patterns are not considered. Nevertheless, it is clear that these patterns have a strong effect on dynamic power since they are directly related to the number of transitions per clock cycle.

Concerning the estimation of the total capacitance C_L of a design, the Rent's rule [55] has been massively used. The rule expresses a relationship between the number of pin/signals and the number of hardware blocks. When going up in abstraction, i.e., at the gate-level, another solution consists in evaluating the hardware complexity through the number of equivalent gates used in a design [3]. An estimation of the overall power is then

TABLE II
CLASSIFICATION OF THE MAIN ANALYTICAL MODELING TECHNIQUES

References	Metrics							
	Inputs	Outputs	modelling level	Error (%)	modelling effort	Model granularity	Year	Labels
[57]	r_{xx0}, r_{yy0}, r_{xx1}, r_{xy1}, r_{yx1}, r_{yy1}	Power	Component	10%(Avg)	**	Fine	2005	(3,2)
[63]	$C(y)$,$D(y)$,f, $Vsupply$,$Vswing$	Power	Circuit	20-4.8%(Avg)	**	Fine	2005	(3,2)
[60]	$C, Vmax_{in}, Vmax_{out}, \tau$	Power	Component	5%(Avg)	**	Fine	2012	(3,1)
[64]	$F_{C_{in}}$,$F_{C_{out}}$,F_s,L,W,I,K	Static P.	Component	15%(Avg)	**	Fine	2013	(3,1)

possible if the average power consumption of an equivalent gate (e.g., 2-input NAND) is known. However, such techniques suffer from poor accuracy because switching activity is not generally considered.

Activity-based models address the power modeling issue analytically from the entropy concept. This concept, borrowed from information theory, is used to evaluate the average activity of a circuit. It consists in finding a relationship between power and the amount of computational work that is performed. Note here that this technique does not consider any timing information, which is a significant limitation.

In [2], a relationship between capacitance and activity has been proposed. In this work, the area is used as a metric to estimate the physical capacitance. This work shows that the power consumption of a circuit heavily depends on the primary input probabilities and activities.

Probabilistic techniques are usually preferred for switching activity and power estimation because of their computational efficiency. In [28] and [40], a transition density model (TDM) is proposed to estimate the switching activity of a node by considering the number of its signals' transitions.

It was demonstrated in [56] that, in order to improve accuracy, spatial, and temporal correlations should be taken into account when estimating switching activity using probabilities. This approach considers signal statistics, such as transition probabilities to model the transition densities of the outputs. Based on this, word-level signal statistics are used to model the power consumption of several operators (e.g., adders). An accuracy of around 10% against XPA (the Xilinx power estimation tool) has been reached [57]. A limitation of TDM is that it does not consider glitches which significantly contributes to dynamic power.

Another interesting approach was proposed in [58] and [59], in which an effective power model based on Markov chains is used to accurately estimate the power sensitivity to the primary inputs. This power sensitivity represents the variations of power dissipation induced by signal inputs. In this work, an average error less than 5% is achieved.

Some works focused on the power modeling of specific FPGA elements (reconfigurable routing resources) by using the equations which are related to the charge/discharge capacitance [60]. Their fine-grain models achieve an accuracy of about 5%. Another work focusing on the modeling of dividers was also proposed in [61]. The authors estimate the power from the divider structure and input signal statistics, such as mean, variance, and auto-correlation. Their model achieves a mean relative error lower than 10% against real measurements.

In addition, the analytical model proposed in [62] allows to evaluate the area-efficiency and logic depth of designs implemented on FPGAs by determining the relationship between the logic blocks and the cluster parameters. By combining such models with a delay model, this approach can be used to quickly evaluate a wide variety of lookup-table/cluster architectures, but still without taking power into consideration.

So far, many approaches have focused on dynamic power modeling, which has represented the main power dissipation source for the last decades. Poon *et al.* [63] presented additional models of short-circuit and leakage powers for FPGA devices. Whereas switching activity is estimated using transition density of every node, dynamic power is also estimated. Their models achieved an error of 4.8% for routing segments up to 20% for other resources.

More recently, the analytical approach models area, delay, and power, allowing both static and dynamic power evaluation during design exploration [64], [65]. This allows designers to explore the impact of FPGA architecture parameters, including the number of logic cells and the associated switch boxes, wire lengths, and clock frequency [66]. For instance, Leow *et al.* [64] improved Poon's model [63] accuracy by integrating the width (W) and length (L) of wires, delivering a static energy model that takes into account logic and routing architecture parameters.

As a summary, Table II presents the main aforementioned analytic models along with their corresponding inputs that enable power estimation. The error and modeling effort are also indicated as well as the model granularity and the modeling level.

Many advantages can be offered by analytical power models, which achieve a relatively good accuracy against low-level simulation tools. However, most of the approaches models power for single components. Since analytical power models are generally used at high-level of abstraction, they make it possible to obtain results very fast, and to enable fast design exploration, especially if the number of hardware resources is a parameter of the power model. Regarding FPGA or even ASIC, technology keeps on evolving by modifying the size of the logic cells, making the generalization of the power model very difficult.

The analytical power modeling technique presents also many disadvantages. More specifically, it is very hard to take into account the effect of glitches. This is of further importance when the model is developed for a component (and not for an entire circuit), destined to be connected to other elements. It is complicated to analytically derive the power consumption of more complex digital systems, and to take efficiently low level

TABLE III
CLASSIFICATION OF TABLE-BASED MODELING TECHNIQUES

References	Metrics							
	Inputs	Characterization Technique	modelling level	Error (%)	modelling effort	Model granularity	Year	Label
[67]	$A(t)$, $A(t-1)$, $B(t)$, $B(t-1)$...,	Statistical	Component	7% (Avg)	***	Fine	1996	(2,3)
[68]	P_{in}, D_{in} & D_{out}	Statistical	Circuit	6% (RMS)	**	Fine	1997	(2,3)
[69]	[70]	Simulation	Component	$<$ 10% (Avg)	**	Fine	1999	(2,4)
[70]	[70] + SC_{in}	Statistical	Circuit	4% (RMS) or 6% (Avg)	***	Fine	2000	(2,3)
[71] & [72]	[72] + T_{in}, P_{out}, D_{out}, S_{out} & T_{out}	Statistical	Component/Circuit	1.84% (Avg) /15% (Avg)	***	Fine	2006/2014	(2,3)

information into consideration. Here, appears the importance of adopting new techniques, like the approaches detailed in the following sections.

B. Table-Based Power Modeling Technique

The LUT-based or table-based power modeling technique is the tabulation process of power values [67]. Each cell of the table is addressed from inputs that have to be carefully chosen, depending on the characterization process. In addition, when power values are missing in the table, an interpolation method can be used. Note here that it does not require any mathematical model compared to the analytical approach. The LUT-based power modeling technique gained a lot of attention from researchers.

Raghunathan *et al.* [67] presented a modeling technique to estimate the switching activity and the power consumption of components at RTL. In this work, glitches are taken into account, in order to increase the accuracy of the estimated power. This done using piece-wise linear models that consider the fluctuation of the output glitching activity and power consumption according to word-level parameters. These parameters may be the mean, the standard deviation, the spatial and temporal correlations, and the glitching activity at the component's inputs. The authors consider more than six factors as input entries that correspond to the present and previous input values of the component (see Table III). The authors claim that the obtained power estimation accuracy is around 7%.

Gupta and Najm [68] presented a modeling method that evaluates the power consumption of a combination circuit. This method quantifies the effect of I/O signals' switching activity. The studied parameters are the average input signal probability, the average input transition density, and the average output zero-delay transition density. They use the resulting power values to build a 3-D LUT for any given I/O signal statistics. This method has been implemented and verified for many benchmark circuits and achieves an accuracy around 6%. Contrary to [67], this work deals with a table dimension that is independent of the number of inputs component. This constitutes an advantage to reduce the table size.

Moreover, Barocci *et al.* [69] analyzed the work proposed in [68], that can be used in the behavioral simulation, and recognized its drawbacks. They propose a new solution based on the interpolation method, which can be helpful if a given entry is not available in the LUT. This method is based on the two closest neighboring entries of the missed entry value, and the power is calculated by linear interpolation between the respective closest power values;

Furthermore, Gupta and Najm [70] added a new attribute to represent the average spatial correlation coefficient. Although, this method showed an RMS error of about 4% and an average error of about 6%, it still ignores the glitch power, which make this model not accurate.

Finally, Durrani *et al.* [71] also proposed a power estimation technique at the RTL. The proposed approach enables designers to estimate the power dissipation of IP components. To model power dissipation, several metrics are used, such as the average input signal probability *Pin*, the average input transition density *D*in, the input spatial correlation *Sin*, the input temporal correlation *T*in, the average output signal probability *P*out, the average output transition density *D*out, the output spatial correlation *S*out, and the output temporal correlation *T*out. The results show an average error of 1.84%. Although this is a very good improvement in terms of estimation accuracy, additional attributes, and computations are required. Durrani and Riesgo [72] extended the work of [71]. They demonstrated the use of the table-based method for a full system power estimation and proved the scalability of the method. The method allows system-level assessment of the power consumption based on earlier characterized components' models.

Table III summarizes the works that use LUTs as a modeling method. It is possible to compare the different approaches according to four metrics: 1) accuracy; 2) modeling effort; 3) modeling level; and 4) modeling granularity. Among all these works, the most significant improvement in terms of accuracy at component level appears in [71].

Although these techniques have demonstrated their feasibility and performance, they also present some limitations. First, they often require to store a lot of data, which is memory consuming. Second, the modeling effort is considered as moderate since it depends on the number of attributes to consider, which may be significant. Finally, the computational effort increases as tables grow, because of the search performed to get the proper value for a given input entry.

C. Polynomial-Based Power Models

Long simulations and lots of parameters often limit design space exploration. To overcome this problem, regression-based power modeling may be used to predict power. It is possible to define linear regression analysis as a statistical inference method, where the relationship between dependent variables (power consumption) and one or more independent variables (i.e., the design parameters) is established [5].

To overcome this problem, polynomial models can be developed to determine the linear relationship between power and one or more independent variables (i.e., the design

TABLE IV
CLASSIFICATION OF POLYNOMIAL-BASED TECHNIQUES

References	Metrics							
	Fitting Parameters	Characterization Technique	modelling level	Error (%)	modelling effort	Model granularity	Year	Label
[73]	P_{in}, D_{in}, SC_{in}, D_{out}	Statistical	Component	20%(Avg)	**	Fine	1999	(4,3)
[74]	[75]+S_{in}, T_{in}	Simulation	Component	1.8%(Avg)	***	Fine	1999	(4,4)
[76]	I/Os switching activities	Simulation	Component	6.1% (Avg)	**	Fine	2000	(4,4)
[75]	same as in [75]	Simulation	Component	3.1%(Avg)	**	Fine	2001	(4,4)
[79]	SC_{in},D_{in}, bitwidth	Simulation	Component	2% (Avg)	**	Fine	2004	(4,4)
[80]	T_{Slice},T_{Mult},T_{BRAM}	Statistical	Component	7% (Avg)	*	Coarse	2008	(4,3)
[78]	$SWCVf$	Measure	Component	3%(Avg)	**	Coarse	2010	(4,5)
[77]	SWC_{eff}	Statistical	Component	6%(Avg)	**	Coarse	2011	(4,3)
[81]	$F, \rho, w, SW, N_{I/O}, N_{Slices}$	Simulation	Component	10% (Avg)	**	Coarse	2012	(4,4)
[88]	$bitwidth, N_{I/O}, N_{LE}$	Statistical	Component	4%(Avg)	**	Coarse	2016	(4,3)
[87]	$Temp, channel\_length, etc.$	Simulation	Component	2.1-6.8%(Avg)	***	Fine	2017	(4,4)
[85]	$n_{FF,LUT,BRAM,DSP}, C_{FF,LUT,BRAM,DSP}$	Measure	Component	5% (Avg)	*	Coarse	2018	(4,5)
[82]	same as in [82]	Statistical	Component	3% (Avg)	**	Coarse	2019	(4,3)

parameters) using a fitting analysis [5]. This category does not include any nonlinear approach that will be discussed later. These techniques deal with power values that are obtained from simulations or measurements according to specific parameters (e.g., capacitance, switching activity, and clock frequency). They are generally bottom-up approaches that usually require a characterization phase performed at low level.

In [73], power models for DSP blocks were developed and achieved an accuracy of 20% against a gate-level simulator. In [74], the work of [73] is extended by considering spatial and temporal correlations of input signals when estimating switching activity using probabilities. Their regression-based approach demonstrated an improvement of the models, with an error lower than 2%.

Besides, Shang and Jha [75] achieved a better accuracy by defining subsets of signals depending on their nature, e.g., control or data signals. An adaptive regression method was employed to build a model considering statistical parameters. This provides a good tradeoff between accuracy and simulation time. Their approach for FPGA and CPLD achieved an average relative error of 3.1%.

Bogliolo *et al.* [76] proposed an advanced regression technique for VLSI circuits. This technique is presented to boost the accuracy of linear models using regression trees algorithm. In fact, if the golden model is strongly nonlinear, a linear approximation may lead to unacceptable large errors. As a consequence, control variables are defined to choose the most appropriate regression equations among different ones. An online power characterization is also proposed to improve the power modeling accuracy of small combination circuits from 34.6% to 6.1%.

As previously mentioned in this article, FPGAs integrate multiple elements, such as embedded DSP blocks, RAM blocks, look-up-tables, and D flip-flops. Some approaches go further by taking into account the number of specific hardware resources in their modeling approach. In [77], a linear relationship between the amount of hardware resources, capacitance, I/Os switching activity, and power is determined to build a general model of IPs. Low-level simulations are performed to obtain signal activities whereas XPower Analyzer delivers power estimates. For a given number of IPs and training sequences (for building the model), the average error is about 6%. When introducing new IPs and patterns, the power estimation error increases up to 35%. Jevtic and Carreras [78] proposed a power model focuses on embedded DSP blocks of FPGAs. It takes into account various signal statistics and multiplier sizes. The model is built using a multivariable regression over different power measurements, achieving an accuracy of 7.9%.

Rather than considering the architectural elements of FPGA devices, another solution consists in creating power models for basic arithmetic operators, such as adders or multipliers [79]–[81]. By taking into account switching activity, operating frequency, auto-correlation coefficients, and words length, a power model can be elaborated as shown in [81]. This model leads to an average of 10% on a Virtex-2 Pro FPGA. However, such approach does not consider the interconnection between elements when estimating the power consumption of complex designs.

Deng's [80] area and power models were recently improved in [82] by considering power optimization technique such as clock gating. The average error obtained for a set of several IPs is decreased to around 3%.

As complexity of hardware systems is growing, a solution could be to automatically identify signals that are the main contributor to power consumption and guide power estimation tools [83], [84]. Another solution could be to directly make use of high-level synthesis to create power models based on resource utilization and real measurements, in order to explore impact of HLS directives on both power and area [85].

A complete framework for FPGA is presented in [86]. This methodology, called functional level power analysis (FLPA), aims at decomposing the system into functional blocks. The components, that are activated in the same function, are clustered and real power measurements are performed. Then power models are computed using regression according to both system and architectural parameters.

As for dynamic power, leakage power models can also be developed using regression. In [87], more than one hundred parameters are used to describe the static power, including channel length and doping, temperature, etc. The accuracy of the macro-model for CMOS technology is very good, up to 2.1% for the 16-nm technology node.

Finally, we summarize the main power modeling approaches based on polynomial linear regression in Table IV.

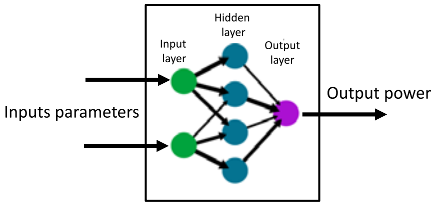

Fig. 7. Neural power model.

According to the table, developing power models with a coarse granularity produces good results with an error lower than 13%. We may note that most of the approaches propose power models for specific hardware component, such as operators (dividers, adder, and DSP) or interconnection elements (multiplexers). Unfortunately, such approaches do not consider complex digital circuits made of different components. Hence, the exploration of the design space of possible configurations is not always feasible.

Moreover, polynomial-based approaches are limited regarding the number of input variables to consider during modeling. They allow to find a linear relationship between a few number of variables but are not adapted for solving nonlinear problems.

D. Neural Networks-Based Techniques

Approaches based on polynomials have a much simpler form than neural networks, and can be characterized in particular as a linear function of features. Nonetheless, model based on neural networks can perform linear and nonlinear regression. This approach is also much more effective because of its hierarchical architecture, which makes it much more efficient in model generalization.

Artificial neural networks (ANNs) are based on connected neurons that propagate information among them, similarly to synapses in the biological brain. As illustrated in Fig. 7 and observed in the literature, ANNs can be used to model power of digital circuits. Existing works have proved the capability of ANNs to approximate generic classes of functions [89].

Hsieh *et al.* [90] proposed a new power modeling technique for CMOS sequential circuits based on recurrent neural networks (RNNs). The main goal consists in learning the relationship between I/O signal statistics and the corresponding power consumption. The work in [90] also considers nonlinear characteristics of power consumption distributions, as well as the temporal correlation of input data. The results show that estimations are accurate with an error range of 4.19%. In fact, this work is limited by two constraints. First, the number of parameters that are needed to estimate the power consumption is large (about 8). Second, the approach is not scalable.

Other works model power consumption of CMOS digital circuits using other types of neural networks such as the back-propagation neural network (BPNN), as in [91]. In this work, the authors model the relationship between power consumption and the circuit's primary I/O statistics. The main difference with [90] is that it does not require a behavioral simulation to obtain output features. The experiments conducted on the ISCAS-85 circuits showed an average absolute relative error below 5.0% for most circuits. Both previous works show the same estimation accuracy, but [91] requires less modeling effort and provides faster estimation.

The importance of using neural networks was also demonstrated in [92], especially for the high-level power estimation of logic circuits. In this work, a simple BPNN is used and a comparison is provided with other modeling methods. In comparison to [90] and [91], the proposed neural network performs better. However, the main limitation of this work is related to the number of the inputs of the model, since it is highly dependent on every input's width of the components to the model.

Neural networks were also used to model the power consumption of a chip as presented in [93]. In this work, the power consumption was modeled using different parameters, such as the frequency and the flash, ROM, and RAM capacity. However, this work has several limitations. First, it does not take into account the inputs' activity. Second, it is only valid for a given chip and cannot be generalized easily. Finally, there is no information regarding the estimation time.

Other works consider other types of neural networks. For instance, Sagahyroon and Abdalla [94] presented a radial base function (RBF) neural network to estimate the energy and the leakage power in standard cell register files. This NN model uses the number of words in the file (D), the number of bits in one word (W), and the total number of read, and write ports (P). However, this limits the power model to specific registers and to a specific technology.

Ramanathan *et al.* [95] presented a method for power estimation of ISCAS'89 Benchmark circuits that exploit both BPNN and RBF. The number of inputs and outputs and the number of logic gates are used as predictors in VLSI circuits. There is no need of detailed architecture description and no interconnection information is required to deliver power consumption as an output. However, the use of neural networks is not well motivated in this work. Moreover, the presented example ignores the fact that power consumption is data dependent.

Lorandel *et al.* [96] used neural networks as a powerful modeling tool to perform both power and signal activities modeling of an IP FPGA-based component. They use simulated data that can be obtained from low-level simulations and evaluate the estimation time. The results showed that the minimum speedup factor achieved by neural models is about 11 500. This demonstrates that neural networks are able to estimate power consumption very fast and with a good accuracy.

In the same context, Tripathi and Rajawat [97] provided power estimate using ANNs. As a result, an average relative error of 3.97% is obtained. Also, a remarkable increase in the estimation speed is observed since its magnitude order is greater than the one for the commercial power estimation tools. Despite these important results, the use of the neural networks is not well derived in this work. In addition, the data-set used to train the neural network is limited (about 74 samples only).

In the literature, more accurate results are achieved when data switching activities are modeled at the circuit level. The work presented in [98] delivers very accurate results that are less than 1%. The main difference between [98] and [99] is

TABLE V
CLASSIFICATION OF NEURAL-BASED TECHNIQUES

References	Metrics							
	Fitting Parameters	Characterization Technique	modelling level	Error (%)	modelling effort	Model granularity	Year	Label
[90]	$TI_{00}, TI_{01}, TI_{10}, TI_{11}, TO_{00}, TO_{01}, TO_{10}, TO_{11}$	Simulation	Circuit	4.19% (Avg)	***	Fine	2005	(1,4)
[91]	$P_{in}, D_{in}, S_{in}, T_{in}$	Simulation	Circuit	2% (RMS) 5% (Avg)	**	Fine	2005	(1,4)
[92]	$TN_I, TN_O, N1_I, N1_O$	Simulation	Component	2.25% (Avg)	**	Fine	2007	(1,4)
[101]	V_{in}, $F_{sampling}$, & $F_{operating}$,	Measure	Circuit	1.53% (Avg)	*	Coarse	2010	(1,5)
[93]	I/O number, Frequency, & Flash depth	-	Circuit	-	*	Coarse	2010	-
[94]	Depth, Width, & Port	-	Component	10.94%(-)	*	Coarse	2012	-
[95]	Number of I/O & Number of gates	Statistical	Component	8.5% (Avg)	*	Coarse	2013	(1,3)
[96]	SW_{in} & P_1	Probabilistic	Component	1.31% (Avg)	**	Fine	2016	(1,2)
[97]	Number of binary, memory and conversion instructions & DSP blocks	Probabilistic	Circuit	3.97% (Avg)	**	Coarse	2018	(1,2)
[99]	Switching activities of input pattern	Simulation	Circuit	3.17% (Avg)	**	Fine	2019	(1,4)
[98]	Register & I/O switching activities	Simulation	Circuit	1% (Avg)	**	Fine	2019	(1,4)
[100]	Frequency, Switching activities & Static probabilities of input pattern	Simulation	Component	9% (Avg)	**	Fine	2020	(1,4)

that convolutional neural networks (CNNs) are used in the former, and the multilayer perceptron (MLP) in the latter.

Finally, Nasser *et al.* [100] presented NeuPow, a neural-based power estimation method that counts the signals characteristics propagation throughout connected neural models. This method also considers the switching activities of the input patterns but operating frequencies as well. It is thus possible to estimate power consumption for various circuits at different frequencies. However, this comes at the cost of an average relative error of 9%, which is relatively big.

A summary of all the aforementioned works that adopt neural networks as a power modeling technique can be found in Table V. This table shows that the works presented in [92], [96], and [101] (performed at component level) exhibit the most accurate results with an estimation error less than 3%. Note here that these works consider MLP neural networks, but more promising results can be achieved using CNNs, as demonstrated in [98].

E. Analysis and Discussion

1) Problem Statement: In this section, we provide a detailed analysis and discussion on all aforementioned power modeling techniques. The main issue when comparing several works in terms of estimation accuracy is the lack of a common reference. Moreover, many factors have to be considered, such as the power measurement setup, the design and power estimation tools, etc. To circumvent this issue, we propose dedicated metrics as detailed in the next section.

2) Metrics: The six following metrics were proposed to compare modeling techniques.

Modeling Effort: It corresponds to the quantity of information and time that is required to build the model. For example, a high modeling effort may require either a long characterization phase or a significant time to obtain data. For moderate and low modeling effort, less data points and time are needed to build the model. Regarding a table-based techniques, the modeling effort is considered as moderate in average. Recall that this metric depends on two factors: 1) the number of used attributes/predictors to build the table/model and 2) the number of points to prepare the table. According to the literature (see Section VI-B), the number of predictors depends on the number of inputs of the component/circuit. Polynomial-based power models may be built with a moderate modeling effort, depending on the amount of data to gather. Finally, neural networks need a high modeling effort since the training time could be significant and a large number of data may be needed to obtain accurate results.

Memory Resources: This metric corresponds to the memory footprint of the technique. For example, the table-based technique consumes a lot of memory resources as compared to an analytic model. Table-based techniques can be considered as memory consuming because of the large number of data points required to store the model. The use of regression in polynomial-based techniques does not require a large amount of memory resources as they do not need to store data. Regarding neural networks, few memory resources are required. In fact, a memory is only used to store weights and biases. However, the memory footprint increases along with the size of the network to implement.

Computational Effort: It represents the computational resources and time needed to perform estimation. We, respectively, associate a high, moderate, and low computational effort to the weights. For table-based approaches, the computational effort increases with table dimensions. This is because of dense search that is required to obtain the correct value for a given entry. Consequently, we can consider it as moderate. For polynomial and analytic models, the estimation time and the computational resources required are low. Regarding NN, the estimation is also low but the computational resources depends on the size of the network. Polynomial models do not imply a significant computation effort, as does not need to perform any additional operation to estimate power. Regarding the computational effort for neural networks, it is considered low due to the time needed to perform the computation of the output with respect to a given input. One can note that this metric is directly related to the network size.

Power Characterization: It indicates if power characterization has to be performed before modeling or not. On the contrary to analytical techniques, table-based, neural networks, and polynomial-based technique do require power characterization.

Accuracy: It represents the fitting capability of a given modeling technique. Regarding neural networks, it was previously remarked that such approach is able to solve complex non linear problem whereas the complexity of an equivalent analytical model or a table would be too important [89]. Neural networks are also able to interpolate better than other modeling approaches. Analytic models are not as efficient and generally lead to a lower accuracy. The same conclusion can be made for table-based approaches, except if the size of the table becomes very significant.

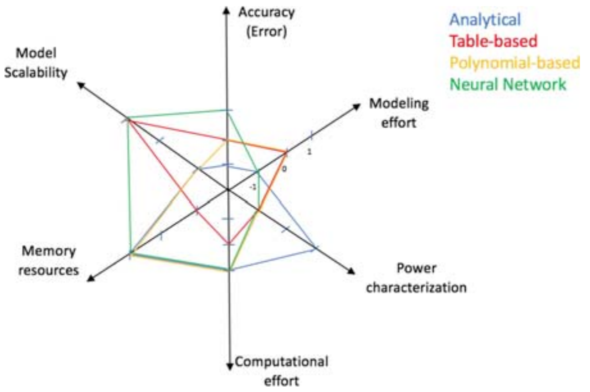

Fig. 8. Graphical comparison of modeling techniques according to defined metrics.

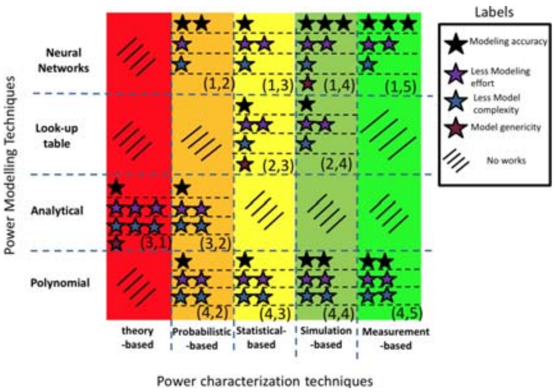

Fig. 9. Classification of the related works.

Polynomial-based solutions constitute a good tradeoff among other approaches.

Modeling Expandability: It corresponds to the capability of a power model to provide power estimation of a composite system. Modeling expandability has been shown in [72], where the table-based power models of IP-based modules were extended and connected together to get the power consumption of a composite system such as a system-on-chip device. The same conclusion can be drawn for neural networks [10], [102].

3) Comparison of Modeling Techniques: From the previous definition, Fig. 8 illustrates the result of different power modeling techniques according to defined metrics.

Even if quantitative results are provided, some general trends can be identified, giving designers the capability to identify the most appropriate modeling technique. At a first glance, the biggest surface highlights the best modeling technique according to a given metric. First, if designers want to quickly obtain power estimations, analytic models deliver a reasonable fitting accuracy with low computational, and modeling efforts as no power characterization is required. However, from our knowledge, the main issue remains how to efficiently generalize an analytical model to a global system, without prohibiting efforts.

Regarding modeling accuracy, scalability/expandability and the use of memory resources, neural networks outperform other modeling techniques, especially table-based and analytic approaches. This is of further importance when designers want to estimate the power of a system composed of several IPs. However, neural networks necessitate a modeling effort that is considerable in comparison to other techniques due to the training phase and a large dataset.

Finally, polynomial-based techniques deliver a good tradeoff since accuracy, modeling effort, and memory resources are balanced by the characterization and the modeling effort. They do not outperform neither analytic approaches nor neural networks but are a good alternative.

VII. Literature Classification

In this section, we present a detailed analysis of the existing works with respect to two approaches: 1) the modeling approach and 2) the characterization technique. Thereby, Fig. 9 illustrates the resulting classification of all works in a matrix. Each column of the matrix identifies a characterization technique whereas a row represents a modeling technique. The following properties are then evaluated for each cell: modeling accuracy, modeling effort, model complexity, and model genericity. A value, represented with a number of stars, is assigned to each cell so that works can easily be compared.

First, we reuse the modeling accuracy, corresponding to the fitting capability of a model. Then, the modeling effort metrics represent the representative amount of time needed for a possible characterization step and the time required to build the model. As a third metric, the model complexity represents the quantitative amount of resources needed to create the model (e.g., memory resources) as well as the computational effort to provide a power estimation. Finally, the model genericity corresponds to the capability of a model to be independent from a CAD tool or technology and the expandability feature (as previously defined).

For each work described in Tables III–V, their position in the matrix is indicated with (i, j) where i stands for the row index (modeling technique) and j represents the column index (characterization technique). This makes it possible to identify the properties of each work very easily. Finally, the more the stars, the better the approach is (according to the defined metrics).

By analyzing Fig. 9, all works based on analytic modeling, see cells (3, 1) and (3, 2), deliver a relatively low fitting accuracy, whereas the model complexity and genericity are satisfying. This is because they do not necessitate too many resources, and do not depend on specific CAD tools. The model complexity is a little lower than (3, 1) due to the power characterization needed for the probabilistic-based approach (3, 2). The model genericity is also lower because of the need of more low-level information.

Works making use of LUTs are either based on statistical or simulation estimation techniques. They have roughly the same level for all considered metrics, but approaches based on statistics are more expandable. The modeling effort is also satisfying, but the model complexity may rapidly increase due to the size of the table, that grows exponentially with the number of inputs.

When comparing neural networks and polynomial-based approaches, the modeling accuracy is generally better for neural networks even if measurements are realized. However, they achieve the same level of the modeling effort. This is because neural networks use a few memory resources but require a learning phase. Moreover, designers must take into account that the size of networks may have a significant impact on the time that is required to create the model, particularly when real measurements have to be performed.

From Fig. 9, a comparison of the works can also be performed according to the columns. When considering probabilistic-based techniques, analytic, and polynomial models are less complex than neural networks to the detriment of a relative loss of accuracy. Moreover, we can see that there is no significant gain by selecting a specific modeling technique when using a statistical estimation technique. For power estimation techniques, neural networks are the most appropriate when accuracy and model genericity are the main objectives.

Fig. 9 also reveals potential areas of research. In fact, the complexity of neural network models and the modeling effort are some of the main limitations of this type of technique. New innovations have to be proposed to make this technique more suitable. Data preprocessing could leverage the training phase by lowering the number of samples, weighting the most significant inputs. New issues arise, such as the way to properly define the network parameters, e.g., the size, number of neurons, number of hidden layers, etc.

Another scientific question remains open and deals with the capability of a model to cover many ASIC/FPGA families. In most cases, custom libraries are made for particular types of devices and models are not necessarily expandable to other devices. This is a clear limitation that is common to many approaches. Even if technologies are different, some scaling factors could be proposed in order to obtain models that are compatible between several device families.

As design methodologies have evolved toward SoC design, current systems are composed of many IPs that are interconnected together. As previously mentioned, it may be interesting to take into account the real activity of these IPs by propagating activities from one model to another. An issue remains, related to the definition of the model interface that is not common to all models. A solution could be the use of the advanced extensible interface (AXI) standard in order to ensure the interoperability between the models.

VIII. Conclusion

The power consumption issue in FPGAs and ASICs digital circuits implies a deep understanding of different power measurement, estimation, and modeling approaches. This is very crucial in order to have more efficient computer-aided design strategies and to help designers in making correct design choices. In this article, a review of the works dealing with FPGA and ASIC power modeling is performed and a classification is proposed according to four modeling techniques. We further define different metrics to perform a fair and relative comparison between them. According to the study, it is shown that polynomial-based and neural networks modeling approaches are superior to analytical and table-based in terms of estimation accuracy and estimation speed. After comparing the modeling techniques, the studied works are then classified and compared along with power characterization techniques. This simplifies the comparison of the multiple works by simply considering specific metrics of interest, e.g., modeling accuracy and effort or model complexity. We also identify potential areas of research to improve some limitations of existing techniques.

References

[1] R. Khatoun and S. Zeadally, “Smart cities: Concepts, architectures, research opportunities,” *Commun. ACM*, vol. 59, no. 8, pp. 46–57, Jul. 2016.

[2] F. N. Najm, “A survey of power estimation techniques in VLSI circuits,” *IEEE Trans. Very Large Scale Integr. VLSI Syst.*, vol. 2, no. 4, pp. 446–455, Dec. 1994.

[3] P. Landman, “High-level power estimation,” in *Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, 1996, pp. 29–35.

[4] E. Macii, M. Pedram, and F. Somenzi, “High-level power modeling, estimation, and optimization,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 17, no. 11, pp. 1061–1079, Nov. 1998.

[5] S. Reda and A. N. Nowroz, “Power modeling and characterization of computing devices,” *Found. Trends Electron. Design Autom.*, vol. 6, no. 2, pp. 121–216, 2012.

[6] A. B. Darwish, M. A. El-Moursy, and M. A. Dessouky, *Power Modeling and Characterization*. Cham, Switzerland: Springer, 2020, pp. 47–57.

[7] J.-P. Deschamps, *Hardware Implementation of Finite-Field Arithmetic*. New York, NY, USA: McGraw-Hill, 2009.

[8] *About Post-Synthesis and Post-Implementation Timing Simulation*, Xilinx, San Jose, CA, USA, Feb. 2012.

[9] I. Kuon and J. Rose, “Measuring the gap between FPGAs and ASICs,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 2, pp. 203–215, Jan. 2007.

[10] Y. Nasser *et al.*, “NeuPow: Artificial neural networks for power and behavioral modeling of arithmetic components in 45 nm ASICs technology,” in *Proc. 16th ACM Int. Conf. Comput. Front. (CF)*, 2019, pp. 183–189.

[11] D. Bellizia, D. Cellucci, V. Di Stefano, G. Scotti, and A. Trifiletti, “Novel measurements setup for attacks exploiting static power using DC pico-ammeter,” in *Proc. IEEE Eur. Conf. Circuit Theory Design (ECCTD)*, 2017, pp. 1–4.

[12] P.-E. Gaillardon, E. Beigne, S. Lesecq, and G. D. Micheli, “A survey on low-power techniques with emerging technologies: From devices to systems,” *J. Emerg. Technol. Comput. Syst.*, vol. 12, no. 2, pp. 1–26, Sep. 2015.

[13] P. Gupta and A. B. Kahng, “Quantifying error in dynamic power estimation of CMOS circuits,” in *Proc. 4th Int. Symp. Qual. Electron. Design*, Mar. 2003, pp. 273–278.

[14] S.-M. S. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits Analysis &Amp; Design*, 3rd ed. New York, NY, USA: McGraw-Hill, 2003.

[15] L. Shang, A. S. Kaviani, and K. Bathala, “Dynamic power consumption in Virtex-II FPGA family,” in *Proc. ACM/SIGDA 10th Int. Symp. Field Program. Gate Arrays*, 2002, pp. 157–164.

[16] Xilinx. (2018) *TI Power Solutions for Measuring Power on Xilinx Evaluation Kits*. [Online]. Available: https://www.xilinx.com/support/answers/39037.html

[17] A. Nafkha and Y. Louet, “Accurate measurement of power consumption overhead during FPGA dynamic partial reconfiguration,” in *Proc. Int. Symp. Wireless Commun. Syst. (ISWCS)*, Sep. 2016, pp. 586–591.

[18] M. A. Rihani, F. Nouvel, J. Prävotet, M. Mroue, J. Lorandel, and Y. Mohanna, “Dynamic and partial reconfiguration power consumption runtime measurements analysis for ZYNQ SoC devices,” in *Proc. Int. Symp. Wireless Commun. Syst. (ISWCS)*, Sep. 2016, pp. 592–596.

[19] R. Jevtic and C. Carreras, “Power measurement methodology for FPGA devices,” *IEEE Trans. Instrum. Meas.*, vol. 60, no. 1, pp. 237–247, Jan. 2011.

[20] J. Oliver, F. Veirano, D. Bouvier, and E. Boemo, “A low cost system for self measurements of power consumption in field programmable gate arrays,” *J. Low Power Electron.*, vol. 13, no. 1, pp. 1–9, 2017.

[21] M. Najem, P. Benoit, F. Bruguier, G. Sassatelli, and L. Torres, “Method for dynamic power monitoring on FPGAs,” in *Conf. Dig. 24th Int. Conf. Field Program. Logic Appl. (FPL)*, 2014, pp. 1–6.

[22] J. J. Davis, E. Hung, J. M. Levine, E. A. Stott, P. Y. Cheung, and G. A. Constantinides, "KAPow: High-accuracy, low-overhead online per-module power estimation for FPGA designs," *ACM Trans. Reconfig. Technol. Syst.*, vol. 11, no. 1, p. 2, 2018.

[23] Z. Lin, S. Sinha, and W. Zhang, "An ensemble learning approach for *in situ* monitoring of FPGA dynamic power," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 9, pp. 1661–1674, Sep. 2019.

[24] J. P. Oliver, F. Favaro, and E. Boemo, "A framework to compare estimated and measured power consumption on FPGAs," *J. Low Power Electron.*, vol. 15, no. 4, pp. 329–337, 2019.

[25] C.-Y. Tsui, M. Pedram, and A. M. Despain, "Exact and approximate methods for calculating signal and transition probabilities in FSMs," in *Proc. IEEE 31st Design Autom. Conf.*, 1994, pp. 18–23.

[26] D. Marculescu, R. Marculescu, and M. Pedram, "Stochastic sequential machine synthesis targeting constrained sequence generation," in *Proc. IEEE 33rd Design Autom. Conf.*, 1996, pp. 696–701.

[27] J. Monteiro, S. Devadas, and B. Lin, "A methodology for efficient estimation of switching activity in sequential logic circuits," in *Proc. IEEE 31st Design Autom. Conf.*, 1994, pp. 12–17.

[28] F. N. Najm, "Transition density: A new measure of activity in digital circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 12, no. 2, pp. 310–323, Feb. 1993.

[29] C.-Y. Tsui, M. Pedram, and A. M. Despain, "Efficient estimation of dynamic power consumption under a real delay model," in *Proc. IEEE Int. Conf. Comput.-Aided Design (ICCAD)*, 1993, pp. 224–228.

[30] R. Marculescu, D. Marculescu, and M. Pedram, "Switching activity analysis considering spatiotemporal correlations," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, 1994, pp. 294–299.

[31] P. H. Schneider and S. Krishnamoorthy, "Effects of correlations on accuracy of power analysis-an experimental study," in *Proc. IEEE Int. Symp. Low Power Electron. Design*, 1996, pp. 113–116.

[32] S. Garg, S. Tata, and R. Arunachalam, "Static transition probability analysis under uncertainty," in *Proc. IEEE Int. Conf. Comput. Design VLSI Comput. Process. (ICCD)*, 2004, pp. 380–386.

[33] L. Lavagno, I. L. Markov, G. Martin, and L. K. Scheffer, *Electronic Design Automation for IC Implementation, Circuit Design, and Process Technology*. Boca Raton, FL, USA: CRC Press, 2017.

[34] L. W. Nagel, "Spice2: A computer program to simulate semiconductor circuits," Ph.D. dissertation, Electron. Res. Lab., Univ. California at Berkeley, Berkeley, CA, USA, 1975.

[35] C. X. Huang, B. Zhang, A.-C. Deng, and B. Swirski, "The design and implementation of powermill," in *Proc. Int. Symp. Low Power Design*, 1995, pp. 105–110.

[36] C. Piguet, *Low-Power CMOS Circuits*. Boston, MA, USA: Kluwer, 2018.

[37] M. Bushnell and V. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits* (Frontiers in Electronic Testing), vol. 17. New York, NY, USA: Springer, 2004.

[38] S. Alipour, B. Hidaji, and A. S. Pour, "Circuit level, static power, and logic level power analyses," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT)*, 2010, pp. 245–249.

[39] R. Burch, F. Najm, P. Yang, and T. Trick, "McPOWER: A Monte Carlo approach to power estimation," in *Proc. Int. Conf. Comput.-Aided Design (ICCAD)*, vol. 92, 1992, pp. 90–97.

[40] M. G. Xakellis and F. N. Najm, "Statistical estimation of the switching activity in digital circuit," in *Proc. 31st Design Autom. Conf.*, 1994, pp. 728–733.

[41] Y. Park and E. Park, "Statistical power estimation of CMOS logic circuits with variable errors," *Electron. Lett*, vol. 34, no. 11, pp. 1054–1056, 1998.

[42] Y. A. Durrani and T. Riesgo, "Efficient power analysis approach and its application to system-on-chip design," *Microprocess. Microsyst.*, vol. 46, pp. 11–20, Oct. 2016.

[43] G. Verma, C. Dabas, A. Goel, M. Kumar, and V. Khare, "Clustering based power optimization of digital circuits for FPGAs," *J. Inf. Optim. Sci.*, vol. 38, no. 6, pp. 1029–1037, 2017.

[44] Synopsys. (2018). *Delivers Accurate Dynamic and Leakage Power Analysis*. [Online]. Available: https://tinyurl.com/y6qz7n36

[45] Cadence. (Aug. 28, 2018). *Genus Synthesis Solution*. [Online]. Available: https://tinyurl.com/y3wqd59y

[46] *Xilinx Power Estimator User Guide: UG440*, Xilinx, San Jose, CA, USA, 2017.

[47] Xilinx. (2014). *XPower Analyzer Overview*. [Online]. Available: https://tinyurl.com/y57ck4va

[48] A. K. Sultania, C. Zhang, D. K. Gandhi, and F. Zhang, *Power Analysis and Optimization*. Cham, Switzerland: Springer Int. Publ., 2017, pp. 177–187.

[49] P. E. P. Estimators, *Power Analyzer*. Accessed: Sep. 4, 2016. [Online]. Available: https://www.altera.com/support/support-resources/operation-and-testing/power/pow-powerplay.html

[50] J. Lamoureux and S. J. E. Wilton, "Activity estimation for field-programmable gate arrays," in *Proc. Int. Conf. Field Program. Logic Appl.*, Aug. 2006, pp. 1–8.

[51] J. B. Goeders and S. J. E. Wilton, "VersaPower: Power estimation for diverse FPGA architectures," in *Proc. Int. Conf. Field Program. Technol.*, Dec. 2012, pp. 229–234.

[52] X. Tang, E. Giacomin, G. D. Micheli, and P. E. Gaillardon, "FPGA-SPICE: A simulation-based architecture evaluation framework for FPGAs," *IEEE Trans. Very Large Scale Integr. VLSI Syst.*, vol. 27, no. 3, pp. 637–650, Mar. 2019.

[53] T. Arslan, A. Erdogan, and D. Horrocks, "Low power design for DSP: Methodologies and techniques," *Microelectron. J.*, vol. 27, no. 8, pp. 731–744, 1996.

[54] R. Burch, F. N. Najm, P. Yang, and T. N. Trick, "A Monte Carlo approach for power estimation," *IEEE Trans. Very Large Scale Integr. VLSI Syst.*, vol. 1, no. 1, pp. 63–71, Mar. 1993.

[55] B. S. Landman and R. L. Russo, "On a pin versus block relationship for partitions of logic graphs," *IEEE Trans. Comput.*, vol. C-20, no. 12, pp. 1469–1479, Dec. 1971.

[56] H. A. Hassan, M. Anis, and M. Elmasry, "Total power modeling in FPGAs under spatial correlation," *IEEE Trans. Very Large Scale Integr. VLSI Syst.*, vol. 17, no. 4, pp. 578–582, Apr. 2009.

[57] J. A. Clarke, A. A. Gaffar, and G. A. Constantinides, "Parameterized logic power consumption models for FPGA-based arithmetic," in *Proc. Int. Conf. Field Program. Logic Appl.*, Aug. 2005, pp. 626–629.

[58] Z. Chen and K. Roy, "A power macromodeling technique based on power sensitivity," in *Proc. 35th Design Autom. Conf. (DAC)*, Jun. 1998, pp. 678–683.

[59] Z. Chen, K. Roy, and E. K. P. Chong, "Estimation of power sensitivity in sequential circuits with power macromodeling application," in *IEEE/ACM Int. Conf. Comput.-Aided Design Dig. Tech. Papers*, Nov. 1998, pp. 468–472.

[60] X. Tang, L. Wang, and H. Xu, "An accurate dynamic power model on FPGA routing resources," in *Proc. IEEE 11th Int. Conf. Solid-State Integr. Circuit Technol.*, Oct. 2012, pp. 1–3.

[61] R. Jevtic, B. Jovanovic, and C. Carreras, "Power estimation of dividers implemented in FPGAs," in *Proc. 21st Ed. Great Lakes Symp. Great Lakes Symp. VLSI (GLSVLSI)*, 2011, pp. 313–318.

[62] J. Das, A. Lam, S. J. E. Wilton, P. H. W. Leong, and W. Luk, "An analytical model relating FPGA architecture to logic density and depth," *IEEE Trans. Very Large Scale Integr. VLSI Syst.*, vol. 19, no. 12, pp. 2229–2242, Dec. 2011.

[63] K. K. W. Poon, S. J. E. Wilton, and A. Yan, "A detailed power model for field-programmable gate arrays," *ACM Trans. Design Autom. Electron. Syst.*, vol. 10, no. 2, pp. 279–302, Apr. 2005.

[64] Y. Leow, A. Akoglu, and S. Lysecky, "An analytical model for evaluating static power of homogeneous FPGA architectures," *ACM Trans. Reconfig. Technol. Syst.*, vol. 6, no. 4, pp. 279–302, Dec. 2013.

[65] H. Mehri and B. Alizadeh, "Analytical performance model for FPGA-based reconfigurable computing," *Microprocess. Microsyst.*, vol. 39, no. 8, pp. 796–806, 2015.

[66] A. Soni, Y. K. Leow, and A. Akoglu, "Post-routing analytical wirelength model for homogeneous FPGA architectures," in *Proc. Int. Conf. Reconfig. Comput. FPGAs (ReConFig)*, 2018, pp. 1–8.

[67] A. Raghunathan, S. Dey, and N. K. Jha, "Register-transfer level estimation techniques for switching activity and power consumption," in *Proc. Int. Conf. Comput.-Aided Design*, Nov. 1996, pp. 158–165.

[68] S. Gupta and F. N. Najm, "Power macromodeling for high level power estimation," in *Proc. ACM 34th Annu. Design Autom. Conf.*, 1997, pp. 365–370.

[69] M. Barocci, L. Benini, A. Bogliolo, B. Riccó, and G. De Micheli, "Lookup table power macro-models for behavioral library components," in *Proc. IEEE Alessandro Volta Memorial Workshop Low Power Design*, 1999, pp. 173–181.

[70] S. Gupta and F. N. Najm, "Power modeling for high-level power estimation," *IEEE Trans. Very Large Scale Integr. VLSI Syst.*, vol. 8, no. 1, pp. 18–29, Feb. 2000.

[71] Y. A. Durrani, T. Riesgo, and F. Machado, "Statistical power estimation for register transfer level," in *Proc. Int. Conf. Mixed Design Integr. Circuits Syst. (MIXDES)*, 2006, pp. 522–527.

[72] Y. A. Durrani and T. Riesgo, "Power estimation for intellectual property-based digital systems at the architectural level," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 26, no. 3, pp. 287–295, Sep. 2014.

[73] S. Gupta and F. N. Najm, "Power macro-models for DSP blocks with application to high-level synthesis," in *Proc. Int. Symp. Low Power Electron. Design*, Aug. 1999, pp. 103–105.

[74] G. Bernacchia and M. C. Papaefthymiou, "Analytical macromodeling for high-level power estimation," in *IEEE/ACM Int. Conf. Comput.-Aided Design. Dig. Tech. Papers*, 1999, pp. 280–283.

[75] L. Shang and N. K. Jha, "High-level power modeling of CPLDs and FPGAs," in *Proc. IEEE Int. Conf. Comput. Design VLSI Comput. Process. (ICCD)*, Sep. 2001, pp. 46–51.

[76] A. Bogliolo, L. Benini, and G. De Micheli, "Regression-based RTL power modeling," *ACM Trans. Design Autom. Electron. Syst.*, vol. 5, no. 3, pp. 337–372, Jul. 2000.

[77] A. Lakshminarayana, S. Ahuja, and S. Shukla, "High level power estimation models for FPGAs," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, Jul. 2011, pp. 7–12.

[78] R. Jevtic and C. Carreras, "Power estimation of embedded multiplier blocks in FPGAs," *IEEE Trans. Very Large Scale Integr. VLSI Syst.*, vol. 18, no. 5, pp. 835–839, May 2010.

[79] T. Jiang, X. Tang, and P. Banerjee, "Macro-models for high level area and power estimation on FPGAs," in *Proc. 14th ACM Great Lakes Symp. VLSI (GLSVLSI)*, 2004, pp. 162–165.

[80] L. Deng, K. Sobti, Y. Zhang, and C. Chakrabarti, "Accurate area, time and power models for FPGA-based implementations," *J. Signal Process. Syst.*, vol. 63, no. 1, pp. 39–50, 2011.

[81] C. Najoua, B. Mohamed, and B. M. Hedi, "Accurate dynamic power model for FPGA based implementations," *Int. J. Comput. Sci. Issues*, vol. 9, no. 2, pp. 84–89, Mar. 2012.

[82] G. Verma, V. Khare, and M. Kumar, "More precise FPGA power estimation and validation tool (FPEV_tool) for low power applications," *Wireless Pers. Commun.*, vol. 106, no. 4, pp. 2237–2246, Jun. 2019.

[83] J. J. Davis, J. M. Levine, E. A. Stott, E. Hung, P. Y. K. Cheung, and G. A. Constantinides, "STRIPE: Signal selection for runtime power estimation," in *Proc. 27th Int. Conf. Field Program. Logic Appl. (FPL)*, 2017, pp. 1–8.

[84] D. Kim, J. Zhao, J. Bachrach, and K. Asanoviá, "Simmani: Runtime power modeling for arbitrary RTL with automatic signal selection," in *Proc. MICRO*, Oct. 2019, pp. 1050–1062.

[85] M. Makni, S. Niar, M. Baklouti, and M. Abid, "HAPE: A high-level area-power estimation framework for FPGA-based accelerators," *Microprocess. Microsyst.*, vol. 63, pp. 11–27, Aug. 2018.

[86] N. Abdelli, A. Fouilliart, N. Julien, and E. Senn, "High-level power estimation of FPGA," in *Proc. IEEE Int. Symp. Ind. Electron.*, Jun. 2007, pp. 925–930.

[87] D. Helms, R. Eilers, M. Metzdorf, and W. Nebel, "Leakage models for high level power estimation," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 8, pp. 1627–1639, Aug. 2018.

[88] S. Mars, A. El Mourabit, A. Moussa, Z. Asrih, and I. El Hajjouji, "High-level performance estimation of image processing design using FPGA," in *Proc. Int. Conf. Elect. Inf. Technol. (ICEIT)*, 2016, pp. 543–546.

[89] F. Scarselli and A. C. Tsoi, "Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results," *Neural Netw.*, vol. 11, no. 1, pp. 15–37, Jan. 1998.

[90] W.-T. Hsieh, C.-C. Shiue, and C.-N. Liu, "A novel approach for high-level power modeling of sequential circuits using recurrent neural networks," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2005, pp. 3591–3594.

[91] X. Gao, Y. Yan, Y. Cao, and W. Qiang, "Neural network macromodel for high-level power estimation of CMOS circuits," in *Proc. IEEE Int. Conf. Neural Netw. Brain*, vol. 2, 2005, pp. 1009–1014.

[92] K. Roy, "Neural network based macromodels for high level power estimation," in *Proc. IEEE Int. Conf. Comput. Intell. Multimedia Appl. (ICCIMA)*, vol. 2, 2007, pp. 159–163.

[93] L. Hou, X. Wu, and W. Wu, "Neural network based power estimation on chip specification," in *Proc. IEEE 3rd Int. Conf. Inf. Sci. Interact. Sci.*, 2010, pp. 187–190.

[94] A. A. Sagahyroon and J. A. Abdalla, "Dynamic and leakage power estimation in register files using neural networks," *Circuits Syst.*, vol. 3, no. 2, p. 119, 2012.

[95] P. Ramanathan, B. Surendiran, and P. Vanathi, "Power estimation of benchmark circuits using artificial neural networks," *Pensee*, vol. 75, p. 9, Sep. 2013.

[96] J. Lorandel, J.-C. Prévotet, and M. Hélard, "Efficient modelling of FPGA-based IP blocks using neural networks," in *Proc. IEEE Int. Symp. Wireless Commun. Syst. (ISWCS)*, 2016, pp. 571–575.

[97] A. N. Tripathi and A. Rajawat, "Fast and efficient power estimation model for FPGA based designs," *Microprocess. Microsyst.*, vol. 59, pp. 37–45, Jan. 2018.

[98] Y. Zhou, H. Ren, Y. Zhang, B. Keller, B. Khailany, and Z. Zhang, "PRIMAL: Power inference using machine learning," in *Proc. 56th Annu. Design Autom. Conf.*, 2019, pp. 1–6.

[99] H. Dhotre, S. Eggersglüß, K. Chakrabarty, and R. Drechsler, "Machine learning-based prediction of test power," in *Proc. IEEE Eur. Test Symp. (ETS)*, 2019, pp. 1–6.

[100] Y. Nasser *et al.*, "NeuPow: A CAD methodology for high level power estimation based on machine learning," *ACM Trans. Design Autom. Electron. Syst.*, to be published. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02518770

[101] A. Suissa, O. Romain, J. Denoulet, K. Hachicha, and P. Garda, "Empirical method based on neural networks for analog power modeling," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 5, pp. 839–844, May 2010.

[102] Y. Nasser, J.-C. Prévotet, and M. Hélard, "Power modeling on FPGA: A neural model for RT-level power estimation," in *Proc. 15th ACM Int. Conf. Comput. Front. (CF)*, 2018, pp. 309–313.

Yehya Nasser was born in Beirut, Lebanon. He received the first Bachelor of Science degree in physics and electronics from Lebanese University, Beirut, in 2012, the second Bachelor of Science degree in communication engineering and the Master of Science degree in computer and communication engineering from Lebanese International University, Beirut, in 2014 and 2016, respectively, and the Doctor of Philosophy degree in electronics engineering (with European Doctorate Label) from the School of Engineering, INSA Rennes, Rennes, France, in 2019.

In 2019, he joined the Research and Development Department of 5G, NOKIA, Lannion, France, as a 5G FPGA/ASIC Design Engineer. His research interests lie in the area of power consumption in digital design, machine learning techniques to model, and to early estimate the power consumption in digital circuits.

Jordane Lorandel received the Ph.D. degree from the IETR-INSA Rennes, France, in 2015.

In September 2016, he joined the Computer Sciences Department, CY Cergy-Pontoise University, Cergy-Pontoise, France, as an Associate Professor. He is also a Member of ETIS Laboratory (UMR 8051 CNRS), Cergy-Pontoise. His major interests are real time embedded and reconfigurable systems. He is also involved in the high-level modeling of embedded systems in order to perform fast power and performance evaluation.

Jean-Christophe Prévotet received the Ph.D. degree and the Research Habilitation degree (HDR) from the Pierre et Marie Curie University, Paris, France, in 2003 and 2019, respectively.

He is currently an Associate Professor with IETR/INSA de Rennes, Rennes, France. His major interests are embedded, reconfigurable systems, and real time systems in general. In particular, his applicative subjects deal with communication systems and the way to optimize their architecture onto real platforms. He is also deeply involved in power modeling of embedded systems hardware, such as FPGAs and ASICs.

Maryline Hélard received the Engineering and Ph.D. degrees from INSA Rennes, Rennes, France, in 1981 and 1884, respectively, and the Habilitation degree in 2004.

After being with Orange Labs as Research Engineer, she joined INSA as a Full Professor in 2007, where she led the IETR Digital Communications Department. She was involved in several collaborative research projects related to her research interests including wired and wireless communications and MIMO techniques.