



**HAL**  
open science

# High-Level Power Estimation Techniques in Embedded Systems Hardware: an Overview

Majdi Richa, Jean-Christophe Prévotet, Mickaël Dardaillon, Mohamad Mroué, Abed Ellatif Samhat

► **To cite this version:**

Majdi Richa, Jean-Christophe Prévotet, Mickaël Dardaillon, Mohamad Mroué, Abed Ellatif Samhat. High-Level Power Estimation Techniques in Embedded Systems Hardware: an Overview. *Journal of Supercomputing*, 2023, 79, pp.3771-3790. 10.1007/s11227-022-04798-5 . hal-03770344

**HAL Id: hal-03770344**

**<https://hal.science/hal-03770344>**

Submitted on 14 Feb 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

---

# High-Level Power Estimation Techniques in Embedded Systems Hardware: an Overview

Majdi Richa · Jean-Christophe Prévotet · Mickaël Dardaillon · Mohamad Mroué · Abed Ellatif Samhat

Received: May 5, 2022 / Accepted: August 28, 2022

**Abstract** Power optimization has become a major concern for most digital hardware designers, particularly in early design phases and especially in limited power budget systems (battery-operated hand-held devices, electro-optical pluggable modules, IoT and green energy systems, etc.). Subsequently, early power consumption estimation at design time is crucial for power optimization. The aim of this paper is to present an overview of High-Level power estimation techniques currently available along with a comprehensive comparison between different methodologies and their applications on estimated models. When high speed and high performance are key features of a specific embedded system, increase in energy consumption becomes the main hurdle to be tackled while keeping speed/performance v/s power consumption trade-off at a minimum. This paper provides designers, interested in power consumption modeling, with knowledge on how to select best power estimation techniques applied to designated target models.

**Keywords** High-Level, power consumption, power estimation, learning-based, simulation-based, measurement-based

## 1 Introduction

With design complexity on the rise and High-Level design tools becoming more widely available, the Field Programmable Gate Array (FPGA) market is poised for large-scale adoption of High-Level design methodologies. In fact, most analysts and industry leaders agree that a considerable number of FPGA designers will eventually switch to High-Level design techniques [8]. This also applies to Application Specific Integrated Circuit (ASIC) [18][2], considered as an important part of embedded systems especially with integrated hard-core processors, tightly coupled with internal logic [5]. As the FPGA/ASIC capabilities have grown, so have the designs, current tools and languages (mainly HDL) do not match the complexity required for advanced digital hardware systems. In the recent years several commercial High-Level Synthesis (HLS) tools have reached maturity, providing a new method to implement FPGA/ASIC designs, or at least some parts of it but at the same time creating challenges in power consumption estimation. [26].

Today, modeling FPGA power consumption is a necessity in two scenarios. First, it is required to perform power-oriented design space exploration at design time. The elaboration of power-efficient designs usually requires multiple iterations of power estimation and design refinement steps, which leads to a long design time and low productivity. In this scenario, having accurate and efficient power models can help to rapidly explore design choices and perform high-level simulations. Second, power modeling may also be used at run-time. In most systems today, it is required to have efficient power monitoring and management. This can be highly useful in taking decisions regarding

---

Richa M., Prévotet JC., Dardaillon M.  
Univ Rennes, INSA Rennes, CNRS, IETR - UMR 6164, F-35000 Rennes, France  
E-mail: mricha@insa-rennes.fr, jprevote@insa-rennes.fr, mdardail@insa-rennes.fr

Mroué M., Samhat AE.  
Lebanese University, FoE, CRSI, Beirut, Lebanon  
E-mail: mohamad.mroue@ul.edu.lb, samhat@ul.edu.lb

Dynamic Voltage and Frequency Scaling (DVFS) or tasks' scheduling in CPU-FPGAs systems. In such cases, a power model has to be implemented in the device itself and run either under software control on a processor or in a dedicated hardware component. The main requirement is to have a "simple" yet efficient model that is capable of extracting basic features of input signals to obtain an accurate value of the circuit consumed power while taking into consideration both speed and accuracy.

Embedded systems designers making the switch to High-Level design methodologies, enjoy some obvious advantages, mainly the easy-handling of increased complexity at a higher level of abstraction, shortening design cycles and improving quality by early functionality verification. Widening the scope, High-Level power estimation and modeling methodologies crossed beyond FPGA and ASIC limits, opening new horizons for microcontrollers/microprocessors-based systems with different but various techniques and architectures (soft-cores, hard-cores, customized instruction set, systems on a single chip, etc.). That said, power estimation and modeling at high level became a necessity. This paper provides an extensive overview on high-level power estimation and modeling techniques. It also presents a comprehensive classification and comparison between different methodologies when applied to designated target models.

This paper is organized as follows: in section 2 we list and define various power estimation techniques. In section 3 we sort and classify estimation techniques and their respective target models. The classification discussion is presented in section 4. Finally we conclude in section 5.

## 2 Estimation Techniques

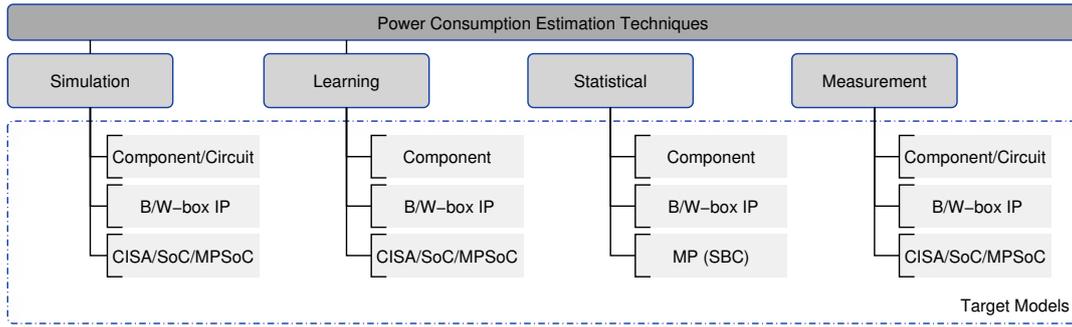
In this section we list and define various power consumption estimation techniques and their corresponding target estimated models. Based on the covered literature, four major power consumption estimation techniques were extracted: 1) the simulation-based methods, 2) the learning-based approaches, 3) the statistical-based methods and 4) the measurement-based alternatives. As for the target models, for each of the previously mentioned estimation techniques, three categories of modeling targets were detected and defined as follows:

1. Components/Circuits: components cover both, operators (arithmetic and logical) and basic logic elements (multiplexers, shift registers, etc.). Circuits modeling is mainly related to capacitance and interconnect wires.
2. White/Black-box IPs: commonly known as reusable units of logic cells. IPs, considered as part of the High-Level design, are valid for both FPGAs and ASIC. A white-box IP is a well known entity where its inputs and outputs are pre-defined. A black-box IP represents an often unknown entity where its inputs can be randomly driven and its outputs are likely to be ignored.
3. Microprocessors/CISA/(MP)SoC: microprocessors ( $\mu$ Ps) are complex modeling targets, their power consumption heavily relies on their architecture and application software. The Customized Instruction Set Architecture (CISA) is a highly optimized approach for simplifying digital hardware and thus reducing power consumption. The Multi-Processor (MP)SoC is mainly an array of processors in a single System on Chip (SoC).

Figure 1 shows an overview of the power estimation techniques and their respective estimated models.

### 2.1 Functional simulation-based methods

The simulation-based power consumption estimation techniques are commonly used in Computer-Aided Design (CAD) tools. This is mainly done by applying various sets of inputs for a given system to be simulated and subsequently collecting and capturing sets of desired outputs leading to power consumption values. Current and voltage values, capacitance, operating frequency and switching activity are the main required set of information in order to obtain power consumption estimation. The accuracy of such techniques is proportional to both processing time and computer memory resources' usage [28]. Here we list some of this technique's target models based on related work:



**Fig. 1** Estimation techniques and target models

### 2.1.1 Components/Circuits

The simulation-based modeling technique described in [36] is initiated by an iterative optimization step during model generation phase. This approach consists of defining modeling variables that are suitable for High-Level Power Estimation (HLPE). Five variables are mainly defined: 1) FPGA device, 2) basic component (generated using Coregen), 3) Hamming distance, 4) signal distance and 5) signal bitwidth. The power characterization is done using XPower while static energy is being deducted from the total power consumption and pads power is being estimated using ORINOCO pad estimator tool (mainly used for pad capacitance extraction). The main advantage of such a technique is that every component has to be characterized only once. As for the disadvantages, this method does not include neither registers nor memory power estimation and it is not fully automated. The RMS error varies between 6.68% and 12.61%.

*TLM POWER3*, a methodology described in [13], consists of creating an add-on SystemC library that accumulates dynamic energy. It is based on the extended generic payload where new physical units for voltage, distance and area have been added. Output reports are generated at the end of this process listing: 1) power consumption, utilization and physical layout and 2) automatic add-up of power and energy used globally. This approach is described to be an easy-to-use power estimation add-on to SystemC TLM.

A pre-silicon power estimation methodology using circuit-level simulation for coarse-grained FPGA architecture has been proposed in [9]. Resource characterization has been used in order to find the effective capacitance. This is done by global wire modeling taking into account the longest interconnect wires. The efficiency in this case is dependent on a number of known simplifications of the FPGA architecture. The power estimation is therefore achieved by finding the utilization of each resources' switching activity. This includes extraction of resource usage (LUT, FF, Carry, etc...) and subsequently averaging the switching activity of each resource. As a result, an average estimation error of 18% has been recorded while the maximum went up to 27%. This proposed method was found to be efficient and accurate, based on clock power, switching activity and dominant interconnect capacitance power. However, being specific to Spartan-3 family, presents a major disadvantage in addition to RAM and shift registers being excluded from estimation.

The hierarchical library-based power estimation concept presented in [23] applies to novel circuit of components, emerging memory devices, architecture of time-multiplexing technique and many others at different levels. It supports coarse-grain or fine-grain estimation defined by users for achieving complexity-accuracy trade-off. The power estimation procedure is a top-down approach; FPGA power is divided into logic, routing and clock power. After extracting power estimation parameters, the calculation of the corresponding power is done while ignoring the detailed implementation of the component. The power estimation of a single component (when reaching the user-defined granularity) is achieved by extracting static power consumption from library while the dynamic power is extracted from the switching activity and the operating frequency. As results, the accuracy of this power estimation methodology depends on the power information given by the hierarchical power library. The information in context includes: static power, dynamic power, ReRAM power estimation and power gating.

In [22], *fpgaEVA-LP2* is a mixed-level power model associated with cluster-based logic blocks and island style routing structures. This methodology takes into consideration 3 power sources: 1) switching power, 2) short-circuit power and 3) static power. Two types of signal transitions are covered: the functional transition (used to perform the required logic) and the spurious transition also known as glitches. The dynamic power model consists of a switch-level model for interconnects and a macromodel for LUTs. This particular paper focuses on transaction density and glitch analysis. While assuming a signal probability of 0.5 and a transaction probability of 0.85, a large number of input vectors is generated. The total power consumption can be broken down into static power and dynamic power (switching and short-circuit). The dynamic power model includes a switch-level model for interconnects and a macromodel for LUTs. The static power, also known as leakage power includes: reverse-bias, sub-threshold, gate tunneling, gate induced drain, etc. This power analysis framework can be used to investigate the impact of circuits, architectures and CAD algorithms upon FPGA power dissipation. As a result, the absolute average error is about 8%.

In [4], Modeling and Analysis of Real-time and Embedded systems (MARTE), a standard Unified Modeling Language (UML) profile promoted by the Object Management Group, is used to create an extension for modeling dynamic power management. For this purpose, a subset of MARTE has been selected, mainly the Hardware Resource modeling (HRM). The Dynamic Power Management (DPM) profile has been developed taking into consideration the requirements of modeling the DPM schemes of modern embedded systems with complex strategies over multiple hardware components. The DPM profile provides model reuse by separating component-level aspects from platform-level aspects on one side and platform-level aspects from application-level on another side.

### 2.1.2 White/Black-box IPs

The proposed simulation-based *Transpilation* method in [16], translates the gate-level component from Verilog to C, trains an energy model while providing an executable script to run system simulation and subsequently calculates energy consumption. This method is fast compared to co-simulation and test-benches approaches but it is partially valid since it does not take into consideration glitches effects and static energy. The resulting estimation error is relatively high (between 55.2% and 152.8% at 10000 samples).

The power consumption estimation of an IP in [7] at Transaction-Level modeling (TLM) is done by capitalization of its existing transaction level model. For this proposed method, switching activity and cells extractions are used. Switching information is extracted using test cases after generation of Value Change Dump (VCD) files. Cells are extracted after converting TLM into a synthesizable format by design compiler. The total power consumption is then calculated as the summation of leakage power and dynamic power. These simulation-based TLM results reveal a total power estimation error of 12.03% and a speed-up of 20x compared to RTL's.

A simulation-based High-Level IP characterization methodology presented in [38] is based on separating the activity of the IP from its implementation. The aim was to create a power model that can be used at different frequencies, layouts and implementation technologies. Therefore, the goal was to abstract the power consumption model of an IP with relatively simple equations. Two IPs were analyzed: 1) a microblaze processor and 2) BRAM memory. While being precise and simple, the average percentage error of the created models was found to be under 15%.

The methodology described in [15] aims at providing power estimates of communication systems ahead of implementation. The proposed approach is solely related to hardware with no software considerations assuming that the wireless communication system is entirely constructed using interconnected hardware IP cores set. This process can be done in two phases: 1) IP characterization using Xilinx Power Analyzer (XPA) and modeling using SystemC and 2) building the system by IP interconnections (previously developed in phase one). The aforementioned technique has been applied to an Orthogonal Frequency Division Multiple Access (OFDMA) IP part of the Long-Term Evolution (LTE), the 4<sup>th</sup> generation of radio technology for mobile communications. As for the results, the maximal absolute error was found to be less than 4% compared to estimated values by XPA.

In comparison with [7], [27] took one step forward and included thermal analysis. Several tools were presented and compared in order to simulate the power and thermal behavior of a chip along with its functionality. The aim was to augment a functional SystemC/TLM model with

non-functional information and subsequently perform power and thermal analysis. The focus was on co-simulating an existing thermal model with a functional SystemC/TLM simulation. Experiments were conducted on both a soft-core processor (MicroBlaze) and an IP (VGA controller); no fundamental issues were detected with the co-simulation techniques on both the SystemC/TLM simulation and the external power/thermal solvers.

The methodology described in [32] is used to estimate energy dissipation in hardware at any level of abstraction in SystemC with *Powersim*. This technique is based on a *Measure space* that consists of the definition of the universe set formed by an elementary event and its duration (as a pair). As for the application of the model described at system-level and composed of different modules, the total energy dissipation can be found by summing the energy dissipated by each module. The energy models can be divided into three sets: 1) logic operators (and, or, xor, not, etc.), 2) arithmetic operators (add, subtract, etc.) and 3) the computational cost. Two test applications were analyzed: a JPEG encoder implemented in hardware and FIR filter implemented in a microcontroller. As for the results, the mean relative error in estimation was about 15.8%.

### 2.1.3 Microprocessors/CISA/(MP)SoC

The HLS-designed, Customized Instruction Set Architecture (CISA) proposed in [3] can be categorized as a simulation-based approach. A well defined selected number of MIPS instructions is being used. For synthesis, power estimation and area analysis, Vivado HLS has been adopted. The estimation is based on the number of consumed cycles by CISA. This technique has been tested on a Dot Product (DP) model and a 128-bit Advanced Encryption Standard (AES-128). The main advantages of such a methodology can be highlighted as follows: a) the power consumption is only 50% to 70% of the total power, b) the dynamic power is 20% to 60% of the fully implemented soft core, c) the smaller the number of instructions, the less resources used and the smaller the occupied area, and d) the pipelining ability. One major disadvantage to mention is that CISA is slower in execution than a full featured ISA.

The technique described in [35] is based on abstraction execution profiles called event signatures. The system-level power modeling framework is based on the *Sesame* MPSoC simulation tool. This signature-based power estimation provides an abstraction of processor activity and communication in comparison to traditional power models. It includes computational and communication event signatures as well as signature-based system-level power estimation. This simulation-based methodology was validated by *Daedalous* tool with an M-JPEG encoder application. The results were compared to measurements done using a PMBus (based on I<sup>2</sup>C) current sensor. The estimation error was about 7% with a standard deviation of 5%. One major disadvantage of this technique is that it relies heavily on many existing tools which prevents full integration and automation.

However, in [40] a top-down power and performance estimation methodology for heterogeneous multiprocessor SoC is proposed and applied at the Electronic System Level (ESL). This is achieved by combining model-based design and spreadsheet power models. The benefits of such a technique is that 1) the estimations can be done early in the design, 2) different design options can be assessed with minimal efforts, 3) dynamic power management is taken into consideration and 4) the possibility of a fast power and performance analysis. For the modeling process, two steps are followed: 1) the characterization of the MPSoC platform and 2) the application modeling. As a use case, an actor-oriented model of an MP3 decoder has been adopted. The estimation performance accuracy was not expected to be high since this approach is intended to provide relative figures for different design options comparison.

The simulation-based methodology presented in [41] is mainly applied for extending system performance modeling frameworks. An *XScale*-based case study is used for verification and a number of components is characterized such as: processors, caches, buses, SRAM, DRAM, peripherals, etc. This technique provides a realistic validation of a system-level execution-driven power modeling (with a fair level of power details for various models). It is considered to be scalable, efficient and validated for incorporating fast and accurate power modeling capabilities into SystemC. The worst case percentage error between measured and estimated values was below 10% with an average error of 5%.

In [42] the High-Level power estimation model for SoC is based on extended and optimized event signatures. The aforementioned estimation model has three primary parts: 1) the summation of every section's power consumption of the SoC design, 2) power consumption produced by event

signatures consisting of bus, memory and communication power sub-models and 3) power consumption disparity due to the subtrahend of the number of LUTs of the complete design and subsequent summation of each part. This proposed model takes into consideration the effect of experiential power estimation data, performance events in addition to the LUT changes. The validation results revealed an error rate of 1.185%.

The multi-core processor (dynamic) power consumption estimation tool presented in [14] extracts information at the instruction level as well as at the architectural level. The MIPS32-based processor has been created using Open Virtual Platform (OVP) and virtual machine interface libraries of Imperas' Instruction Set Simulator (ISS). Two, four and eight processor cores were simulated using ISS where different benchmarks were used as application. As a result, it was noted that the energy consumption is inverse-proportional to the number of cores used on the platform. The reason behind is that the same application takes more time on a platform having less processor cores.

On another hand, a formal description of a power consumption estimation approach of embedded systems is presented in [39]. An embedded system consisting of a hardware and a software is denoted as a System Model (SM). The hardware part of the system is reflected in the Operational Model (OM) while the software part is described in the Application Model (AM); UML extended with MARTE profile elements, was used for this purpose. To analyze the power consumption of the system, SM, both OM and AM were combined and converted into a stochastic Petri net. As an example and direct application to the proposed method, an industrial control system was used and its average power consumption was estimated without being compared to other references.

## 2.2 Learning-based methods

By definition, learning-based techniques are subsets of Artificial Intelligence (AI) methods, providing automatic power consumption estimation, acquired and improved from experience without being explicitly stored or programmed. Usually, in such techniques, data sets, commonly known as training sets, correlating inputs to outputs have to be collected and applied prior to initiating learning methods. Based on the covered literature, tree-based algorithms and Artificial Neural Networks (ANNs), both under the supervised learning methods, are being widely adopted. A supervised learning method uses special patterns to identify specific characteristics. A pattern is always consistent and recurrent [12]. Here we list some of this technique's target models based on related work:

### 2.2.1 Components/Circuits

In [29] a power model based on neural networks was presented in order to predict the power consumption of digital operators in FPGAs. Any digital hardware system can be represented by a set of components, each with a specific function and selected during the High-Level design process. The suggested method represents each operator (component) by two sub-models denoted by M1 and M2. M1 predicts the power consumption for given Signal Rates (SR) and Percentage Logic High (PLH), while M2 estimates the signal activity of the outputs in addition to the PLH according to the inputs. A characterization phase is required by applying a set of stimuli to a given operator and collecting power information using XPA tool. Both M1 and M2 are implemented using Multi-Layer Perceptrons (MLP) feed forward neural networks. As for the results, various operators were chosen (adders, multipliers, etc...); the percentage error was about 0.01% at the component level and less than 8% at the system level (mainly composed of cascaded components).

*NeuPow*, a complement of the methodology described in [29] and [31], is an Artificial Neural Network (ANN) for power and behavioral modeling of arithmetic components in 45 nm ASIC described in [30]. The proposed method is achieved by propagating predictors between the connected neural models in order to estimate dynamic power consumption of individual components. Two different ANN models are used for each arithmetic component: 1) for estimating power 2) for determining the activity propagation. Initially, data set vectors are generated and fed into the inputs of both ANNs. Dynamic power and output data depend only on the input data for a fixed technology and frequency. Bit width, packet length and number of packets represent the stimuli parameters. This technique is divided into two sections: power and behavioral characterization, and, power and behavioral modeling. The final step is to construct the library based on the two previously

mentioned ANNs: ANNP to predict power consumption and ANNB to predict the output feature vector. This learning-based estimation methodology applies for both component and system level. As for the results, an RMSE of +/- 1.5% was recorded for the component level and 8.5% for the system level with an estimation speed up of about 2490x compared to traditional tools.

### 2.2.2 White/Black-box IPs

The *HL-Pow*, a learning-based methodology described in [24], is a power modeling framework based on machine learning. It describes an automated feature construction flow to efficiently identify and extract features that exert a major influence on power consumption. The *HL-Pow* design flow has two phases: 1) power model training with a collection of applications and 2) power inference for new applications. In the training phase, a number of representative applications described in C or C++ are used to generate training samples for power modeling. Two main types of features are taken into consideration: architecture and activity. For the power model generation, *HL-Pow* constructs a total number of 256 features consisting of 11 architecture features and 245 activity features. The regression models were built using many supervised learning methods, such models are: 1) linear regression, 2) Support Vector Machine (SVM) and 3) tree-based model. As for the results, the power modeling error was 4.67% away from measured power.

The decision tree-based learning method described in [20] relies on extracting data-dependent invocation-by-invocation power model from gate-level cycle-by-cycle power traces. The process starts by applying input vectors simulation to black box IP and its TLM model. Based on output traces, invocation-by-invocation power model with fine-grain data-dependent is used while internal signal activity is indirectly observed from switching activity of I/O signals. SciKit-learn machine learning library was used for model selection and training. The trained cycle-level models are combined to predict invocation-level execution power of the IP. As a result, the method provides a 9x faster prediction than cycle-level with less than 2% average error.

However and opposed to [20], the learning-based power modeling approach described in [19] is based on annotating functional hardware descriptions and capabilities allowing the capture of data-dependent resources, block or I/O activity without significant loss in simulation speed. Activity models are first generated then machine learning techniques are leveraged in order to synthesize power models at different granularities. Compared to commercial gate-level or RTL estimation tools, this approach achieved an estimation accuracy of 10%, 9% and 3% for cycles-, block- and invocation-level respectively. This technique is fully automated by integrating with commercial HLS tools.

### 2.2.3 Microprocessors/CISA/(MP)SoC

Existing CPU power models rely on either generic analytical power models or simple regression-based techniques that suffer from large inaccuracies. Today, machine learning techniques are proposed to build accurate power models. Authors in [17] present a hierarchical power modeling approach that deals with power models for CPUs at micro-architecture level. A decision tree is build for a RISC-V core that can predict cycle by cycle power with less than 2.2% error.

More recently, a micro-architecture power modeling framework has been proposed in [43]. It provides a power modeling flow and methodology and trains ML calibration models using given configurations of the core and provide power estimates for other configurations. Compared with state-of-the-art power models, this approach can reduce the mean absolute percentage error (MAPE) under different cross-validation strategies by 3% to 6%.

## 2.3 Statistical-based methods

The statistical power estimation methods have common features with the previously described simulation-based techniques. A random bit sequence (stimuli) is applied to the primary inputs of a specific logic circuit and subsequent simulation is performed using an off-the-shelf power estimator until a satisfactory precision is hit. Here we list some of this technique's target models based on related work:

### 2.3.1 Components/Circuits

For the statistical-based methods applied to component/circuits, it is very rare to find on-going works on the topic. A previous work, [25] deals with power consumption estimation by analyzing statistical properties of interface signals. It is mainly based on the evaluation of the transition activity which is 40% smaller than the real values, which leads to an accurate power consumption estimation. More recently, authors in [34] presented an evaluation of power estimation flows used in standard cells ASIC for video applications. The results show that flows based on statistical approaches can present a relative error about 40% for evaluated designs.

### 2.3.2 White/Black-box IPs

Again, due to lack of recent work, the amount of references is found to be at minimum. In [10], the authors propose a High-Level power estimation model based on input signals statistics defined as: the input signal probability, the average input signal density and the spatial correlation. In their experiments, authors indicate an average error of 29.63% regarding power estimation. More recently, in [6], the methodology has been applied on a SDRAM IP core and the maximum estimation error was found to be 12.5%.

### 2.3.3 Microprocessors/CISA/(MP)SoC

In contrast with [14], [33] is a power estimation prediction technique based on a statistical model. Its portability is analyzed on new generations of Raspberry Pi Single Board Computer (SBC). In such systems, the power consumption does not depend on the hardware only but also on the use of the software and its internal characteristics. Two different benchmarking tools have been used having different computational behaviors, parallel applications source code compilations, collection of performance counters, linear regression model training and model validation. The final statistical model is based on linear regression and subsequent generation of 8 power coefficients leading to one unified model formula. As for the results, the maximum prediction error was about 18.46%, an error dispersion of 4.14% and an average error of 4.76%

## 2.4 Measurement-based methods

A natural approach for evaluating power consumption consists in conducting real measurements on hardware after design implementation. This is usually done either a) via built-in current sensors (on-board or on-chip), or b) via a well defined external instrumentation setup [37].

*Built-in measurements:* This kind of measurements is portable and may be achieved using 1) on-board current sensors and voltage regulators (mostly monitored over the standard Power Management (PM) Bus or using on-board Analog-to-Digital Converters - ADCs) or 2) on-chip current sensors (mainly implemented in ASIC). These techniques' integrity is however dependent on the fidelity of the sensors in context and is highly affected by both the electrical noise and interference.

*External measurements:* Another alternative would be to use external instruments to measure power consumption. This is mainly achieved by connecting analog acquisition instruments (basically built using high accuracy ADCs) to the Device Under Test (DUT). This is only possible when the DUT permits such a procedure by exposing 1) the core supply load (via a Shunt power resistor) and/or 2) a set of test points representing the analog power consumption on specific sub-circuits. These acquisition systems are usually hooked to PC software for storage and analysis.

Here we list some of this technique's target models based on related work:

### 2.4.1 Components/Circuits

The power characterization methodology described in [11] relies upon a set of measurements where both static and dynamic power are extracted. The proposed method is based on downloading different hardware designs with a variable number of resources (connections, CLBs, LUTs, registers,

IOBs, etc.) on FPGAs. The total power consumption is determined based on power consumption subsets related to each component. The variation of static and dynamic power consumption with respect to multiple FPGA targets is monitored by implementing a unified design on many identical FPGA boards. This methodology is mainly used to determine the unit power consumption of the direct horizontal and vertical lines in addition to LUTs.

#### 2.4.2 White/Black-box IPs

A cycle-accurate energy measurement and High-Level characterization methodology based on the switched capacitor technique was adopted in [21]. For the measurement, a pipelined Analog to Digital Converter (ADC), a vector generator and an Ethernet-based system management CPU has been utilized. The energy, for both low- and High-Level approaches, has been characterized. Since our main focus is on High-Level techniques, we only present the HL part of the energy characterization. For this purpose, a state-machine-based technique has been selected while separating static and dynamic power consumption. This method applies to SRAM-based FPGAs. Two IPs were analyzed, LCD and SDRAM controllers, however no comparison with alternative methods has been explicitly revealed, thus no % error has been published.

#### 2.4.3 Microprocessors/CISA/(MP)SoC

The method of power consumption estimation proposed in [1] is based on an experimental bench developed for Virtex-6 FPGA. This method consists of duplicated Processing Units (PUs) mainly MicroBlaze softcore processors (1, 2, 4, 8 and 16 cores) where three different C programs were executed for each of the five designs. Two versions v1 and v2 of the five implemented designs were used to evaluate the proposed method. Results of measured values were compared to XPA simulated values and it was noticed that the 16 cores system has higher estimation accuracy (% error less than 4) than the 1 core system (% error higher than 30).

### 3 Approaches classification

Various power consumption estimation techniques and methodologies were investigated along with associated target models. In order to sort and classify the aforementioned techniques and models, a set of metrics had to be defined. Given the variety and extensiveness of the covered literature, and due to the lack of common ground, the metrics had to be divided into quantitative and qualitative sets. We list and define the extracted metrics as follows:

- Dependency: a given estimation technique may or may not be dependent on specific hardware, software, tools and/or technologies.
- Characterization: a given estimation technique may or may not require a characterization phase ahead of estimation and modeling. The characterization reveals the distinctive nature of a given target model and highlights its special features/characteristics such as capacitance, switching activity, resources, etc.
- Estimation effort: a qualitative metric represented by either little effort (+), moderate effort (++) and considerable effort (+++).
- Estimation error: this quantitative metric is not always explicit yet it could be totally absent. The % error could be either minimum (min), maximum (max) or average (mean/avg).
- Modeling effort: similarly to the estimation effort, this qualitative metric is also represented by either (+), (++) or (+++).
- Modeling level: this metric reveals the level at which the model was created. Possible options are: RTL, system, TLM, component, HLS, IP or a combination of any two (ex: Comp/Sys, IP/Sys, etc.).

Table 1 encapsulates and classifies both the power consumption estimation techniques and their investigated target models based on the previously described set of metrics. References were selected based on their exposed data and on the extracted information following the diagram in Figure 1. The table's entries were sorted based on the number of occurrences in the literature for both the estimation techniques and their respective target models in descending order.

**Table 1** Classification

Methodology	Target model	Reference	Dependency	Characterization	Estimation effort	Estimation error	modeling effort	modeling level
Simulation	IP	[16]	n/a	No	+	mean 100%, max 150%	n/a	System
		[7]	n/a	No	++	mean 12%, max 20%	n/a	TLM
		[38]	n/a	Yes	++	mean 15%, max 50%	n/a	TLM
		[15]	n/a	Yes	+++	max 4%	++	System
		[27]	LIBTLM PWT	No	+++	n/a	+	TLM
		[32]	Powersim	No	++	mean 15.8%	+	IP/Sys
	[3]	Vivado DS	No	++	n/a	n/a	HLS	
	[35]	n/a	No	++	avg 7%	++	System	
	[41]	XScale P	Yes	++	mean 5%, max 10%	+++	TLM	
	[42]	n/a	No	+++	1.185%	+	System	
	[14]	Virtual platforms	No	++	n/a	n/a	Instruction	
	[39]	MARTE	No	++	max 2%	++	Comp/Sys	
	[40]	ESL	Yes	++	n/a	++	System	
	[36]	User Intervention	Yes	++	min 6.68%, max 12.61%	++	Component	
	[13]	TLM Power3	No	+++	n/a	n/a	TLM	
	[9]	Spartan-III	Yes	+	mean 18%, max 27%	n/a	Comp/Sys	
	[23]	n/a	No	++	min 24%	n/a	Comp/Sys	
	[22]	n/a	Yes	++	avg 8%	++	Component	
	[4]	MARTE	No	++	n/a	++	Comp/Sys	
	Learning	IP	[24]	n/a	Yes	++	4.67%	++
[20]			Virtual platforms	No	++	mean 3%, max 15%	++	TLM
[19]			n/a	Yes	++	avg 7.33%	++	System
$\mu$ P/CISA/(MP)SoC		[17]	RISC-V	Yes	++	max 2.2%	++	Gate
		[43]	RISC-V	Yes	++	max 3%	++	System
Component		[30]	n/a	Yes	+++	mean 1.5%, avg 8.5%	+++	Comp/Sys
		[29]	XPA	Yes	++	min 8%, max 17%	++	Comp/Sys
Statistical	IP	[10]	n/a	Yes	+	avg 29.63%	+	System
		[6]	DataFit tool	Yes	++	max 12.5%	++	System
	P (SBC)	[33]	RPI SBC	No	++	avg 4.76%	+	System
		[25]	REPLICA	Yes	++	n/a	++	System
		[34]	RC/iRun/Cadence	No	+	40%	n/a	RTL
Measurement	IP	[21]	SRAM FPGA	Yes	+	n/a	+	System
		[1]	n/a	No	++	min 4%, max 30%	n/a	System
	Component	[11]	n/a	Yes	+	n/a	n/a	Component

## 4 Discussions

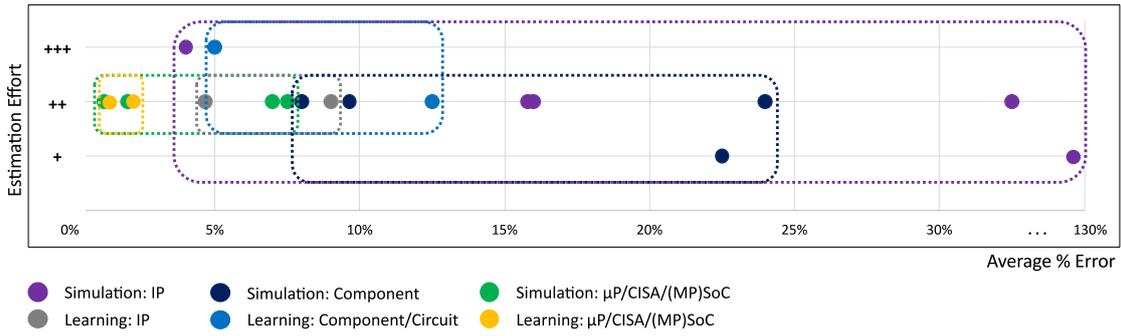
Based on the covered literature, the following is a comprehensive comparison between various estimation techniques applied to common target models:

- For the white/black-box IPs target models, the estimation error was found to be almost inverse-proportional to the estimation effort targeting the simulation-based methods; the error range was higher and wider than the learning-based counterparts. Considerable estimation efforts in simulation-based methods definitely lead to satisfactory estimation % error, highly comparable to the learning-based alternatives, yet providing better results in some specific cases. As a global comparison, the learning-based techniques provide better performance in estimating power consumption at moderate efforts.
- For the component/circuit target models, a smaller and narrower range of estimation error was clearly detected with the learning-based methods. These latter, achieved with considerable efforts, provide higher efficiency compared to the simulation-based alternatives where little to moderate efforts were applied.
- $\mu$ P/CISA/(MP)SoC target models, estimated using the simulation-based methods, showed a % error  $>30\%$ . Learning-based alternatives presented better results with error round 3%. However, good estimation results can be obtained by just applying moderate efforts.

Figure 2, a cluster-based representation, combines a number of power consumption estimation techniques applied to various target models. The purpose of having such a global chart is to get a wider understanding of [estimation technique, target model] pairs' efficiency and subsequent selection. The vertical axis represents the estimation effort while the horizontal one represents the average estimation % error. The chart shows only simulation- and learning-based methods as the first method is widely used and the second method is the most recent.

This graphical representation reveals the distribution of the cluster-based power estimation methodologies applied to specific target models v/s their respective estimation efforts. The following analysis was noted:

*Simulation-based techniques:* The simulation-based power consumption estimation techniques are best suitable for  $\mu$ P/CISA/(MP)SoC target models with estimation error  $<5\%$  in specific cases, especially when software applications targeting (soft- and hard-core)  $\mu$ Ps are involved. This is usually done with moderate efforts. The component/circuit target models come next in the classification



**Fig. 2** Estimation Effort v/s Average % Error - Clusters Chart

under the same estimation methods with also moderate efforts, leaving behind the white/black-box IP models, unless considerable efforts were exerted. The major advantages of such a technique are: a) its generic nature and b) its relatively high precision. High memory requirements and slow execution time remain major drawbacks especially when fast prototyping is required and when resources are limited.

*Learning-based techniques:* Also known as pattern recognition, Machine Learning-based power consumption estimation techniques, when applied to white/black-box IP and  $\mu$ P/CISA/(MP)SoC models even with moderate efforts, provide higher efficiency to when applied to component/circuit models. However, considerable estimation efforts, definitely push the efficiency higher when dealing with component/circuit target models especially when artificial neural networks are adopted. The major advantages of such techniques can be summarized as follows: 1) easy identification of trends and patterns, 2) no human intervention is needed, 3) ability to handle multi-dimensional and multi-variety data, and most importantly 4) continuous improvements coupled with wider range of target models. Time/resources requirements, errors susceptibility and the need of prior data acquisition/collection can be considered as main hurdles.

*Statistical-based techniques:* even though these methods have not been used in recent work, they made it to the classification. Little to moderate estimation efforts lead to estimation errors around 3.5% for the processor-based targets and 12.5% for IP targets. These techniques are usually simple to apply and easy to interpret their outputs. However, the estimation ability may decrease drastically due to data sparsity and modeling overfitting.

*Measurement-based techniques:* Even though tightly coupled with instruments' fidelity, probing level and sensing accuracy, the measurement-based methods are considered as reliable alternatives in estimating power consumption. Due to lack in estimation accuracy results exposure, these techniques did not make it to the global clusters chart. As for the estimation efforts, little to moderate levels were applied. It's important to mention that measurement results are often used in characterization for later data processing and subsequent power estimation using various methodologies but mainly machine learning techniques. Measurement-based techniques are non-invasive and relatively easy to handle but the dependency on the instrumentation (that can be sometimes bulky) and its accuracy might affect the fidelity of the results and thus the estimation % error. Moreover, these techniques often target a particular IP or circuit and lacks of genericity.

## 5 Conclusion

In this overview paper many power consumption estimation-related topics were covered. High-Level power estimation techniques and target models were presented as the main concern and purpose of this paper. The covered literature was extensive with a considerable variety of evaluation methods and contribution levels applied to many sets of target models.

Knowing that, the papers' classification had to be done in a tabular format (Table 1) where some of the metrics were quantitative and some others qualitative. A global cluster-based chart (Figure 2) representing the power estimation techniques and their respective target models v/s the

estimation efforts has been generated. The purpose of that representation was to have a general and wide understanding of which estimation technique applied to which target model results in a better estimation accuracy while keeping an eye on required estimation efforts. The ultimate purpose of this work was to provide guidance for designers (dealing with power estimation and modeling methods) in terms of assessment, comparison and selection of the appropriate power estimation techniques applied to target models based on performance, accuracy and efforts.

As a future sight and given the extensiveness of current applications and continuous enhancements coupled with hardware-backed computational power, machine learning techniques might be nominated and foreseen as the most effective and global approaches for power consumption estimation and subsequent power modeling.

## References

1. Affi, S.M., Verdier, F., Belleudy, C.: Power estimation method based on real measurements for processor-based designs on FPGA. Proceedings - 2014 International Conference on Computational Science and Computational Intelligence, CSCI 2014 **2**, 260–263 (2014). DOI 10.1109/CSCI.2014.133
2. Ahuja, S., Mathaikutty, D.A., Singh, G., Stetzer, J., Shukla, S.K., Dingankar, A.: Power estimation methodology for a high-level synthesis framework. Proceedings of the 10th International Symposium on Quality Electronic Design, ISQED 2009 pp. 541–546 (2009). DOI 10.1109/ISQED.2009.4810352
3. Ananthanarayana, T., Lopez, S., Lukowiak, M.: Power analysis of HLS-designed customized instruction set architectures. Proceedings - 2017 IEEE 31st International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2017 pp. 207–212 (2017). DOI 10.1109/IPDPSW.2017.59
4. Arpinen, T., Salminen, E., Hämäläinen, T.D., Hännikäinen, M.: MARTE profile extension for modeling dynamic power management of embedded systems. Journal of Systems Architecture **58**(5), 209–219 (2012). DOI 10.1016/j.sysarc.2011.01.003. URL <http://dx.doi.org/10.1016/j.sysarc.2011.01.003>
5. Carballo, P.P., Espino, O., Neris, R., Hernández-Fernández, P., Szydzik, T.M., Núñez, A.: Scalable video coding deblocking filter FPGA and ASIC implementation using high-level synthesis methodology. Proceedings - 16th Euromicro Conference on Digital System Design, DSD 2013 pp. 415–422 (2013). DOI 10.1109/DSD.2013.52
6. Chalil, A.: Implementation of power estimation methodology for intellectual property at SoC level. Proceedings of the 2nd International Conference on Communication and Electronics Systems, ICCES 2017 **2018-January**, 1010–1013 (2018). DOI 10.1109/CESYS.2017.8321234
7. Chandoke, N., Sharma, A.K.: A novel approach to estimate power consumption using SystemC transaction level modelling. 12th IEEE International Conference Electronics, Energy, Environment, Communication, Computer, Control: (E3-C3), INDICON 2015 pp. 1–6 (2016). DOI 10.1109/INDICON.2015.7443519
8. Cong, J., Liu, B., Neuendorffer, S., Noguera, J., Vissers, K., Zhang, Z.: High-level synthesis for FPGAs: From prototyping to deployment. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **30**(4), 473–491 (2011). DOI 10.1109/TCAD.2011.2110592
9. Degalahal, V., Tuan, T.: Methodology for high level estimation of FPGA power consumption. Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC **1**, 657–660 (2005). DOI 10.1145/1120725.1120986
10. Durrani, Y.A., Abril, A., Riesgo, T.: Efficient power macromodeling technique for IP-based digital system. Proceedings - IEEE International Symposium on Circuits and Systems pp. 1145–1148 (2007). DOI 10.1109/ISCAS.2007.378252
11. El-dib, D., El-dib, D.A., Alkabani, Y., El-moursy, M., Shawkey, H.H.: Automated FPGA Power Characterization Methodology. International Journal of Electrical and Computer Sciences IJECS-IJENS **16**(2) (2016)
12. Gallo, C.: Artificial neural networks. Artificial Neural Networks (January 2015), 1–426 (2011). DOI 10.4324/9781315154282-3
13. Greaves, D., Yasin, M.: TLM POWER3: Power estimation methodology for SystemC TLM 2.0. Lecture Notes in Electrical Engineering **265 LNEE**, 53–68 (2014). DOI 10.1007/978-3-319-01418-0-4
14. Irfan, M., Masud, S., Pasha, M.A.: Development of a High Level Power Estimation Framework for Multicore Processors. In: Proceedings of 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, IMCEC 2018 (2018). DOI 10.1109/IMCEC.2018.8469473
15. Jordane, L., Prévotet, J.C., Héliard, M.: Fast Power and Energy Efficiency Analysis of FPGA-based Wireless Base-band Processing. HIP3ES, Prague, Czech Republic (2016). URL <http://arxiv.org/abs/1601.00834>
16. Knodtel, J., Schwabe, W., Lieske, T., Reichenbach, M., Fey, D.: A Novel Methodology for Evaluating the Energy Consumption of IP Blocks in System-Level Designs. 2018 28th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS) pp. 46–53 (2018). DOI 10.1109/PATMOS.2018.8464149. URL <https://ieeexplore.ieee.org/document/8464149/>
17. Kumar, A.K.A., Gerstlauer, A.: Learning-based CPU power modeling. 2019 ACM/IEEE 1st Workshop on Machine Learning for CAD, MLCAD 2019 (2019). DOI 10.1109/MLCAD48534.2019.9142100
18. Lakshminarayana, A., Ahuja, S., Shukla, S.: High Level Power Estimation Models for FPGAs. 2011 IEEE Computer Society Annual Symposium on VLSI pp. 7–12 (2011). DOI 10.1109/ISVLSI.2011.79
19. Lee, D., Gerstlauer, A.: Learning-Based, Fine-Grain Power Modeling of System-Level Hardware IPs. ACM Transactions on Design Automation of Electronic Systems **23**(3) (2018). DOI 10.1145/3177865. URL <https://dl.acm.org/doi/10.1145/3177865>
20. Lee, D., Kim, T., Han, K., Hoskote, Y., John, L.K., Gerstlauer, A.: Learning-Based Power Modelling of System-Level Black-Box IPs. IEEE/ACM International Conference on Computer-Aided Design pp. 847–853 (2015)

21. Lee, H.G., Nam, S., Chang, N.: Cycle-accurate energy measurement and high-level energy characterization of FPGAs. Proceedings - International Symposium on Quality Electronic Design, ISQED **2003-Janua**, 267–272 (2003). DOI 10.1109/ISQED.2003.1194744
22. Li, F., He, L.: Power Modeling and Characteristics of Field Programmable Gate Arrays. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **24**, 1–13 (2005)
23. Liang, H., Chen, Y.C., Luo, T., Zhang, W., Li, H., He, B.: Hierarchical Library Based Power Estimator for Versatile FPGAs. Proceedings - IEEE 9th International Symposium on Embedded Multicore/Manycore SoCs, MCSoc 2015 pp. 25–32 (2015). DOI 10.1109/MCSoc.2015.44
24. Lin, Z., Zhao, J., Sinha, S., Zhang, W.: HL-Pow: A Learning-Based Power Modeling Framework for High-Level Synthesis. In: 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 574–580. IEEE (2020). DOI 10.1109/ASP-DAC47756.2020.9045442. URL <https://ieeexplore-ieee-org.rproxy.insa-rennes.fr/stamp/stamp.jsp?tp={&}arnumber=9045442>
25. Ludewig, R., Ortiz, A.G., Murgan, T., Glesner, M.: Power estimation based on transition activity analysis with an architecture precise rapid prototyping system. Proceedings of the International Workshop on Rapid System Prototyping **2002-January**, 138–143 (2002). DOI 10.1109/IWRSP.2002.1029749
26. Marjanovic, J.: Low vs high level programming for FPGA. Proceedings of the 7th International Beam Instrumentation Conference, IBIC 2018 pp. 527–533 (2018). DOI 10.18429/JACoW-IBIC2018-thoa01. URL <http://jacow.org/IPAC2018/papers/thob03.pdf>
27. Moy, M., Helmstetter, C., Bouhadiba, T., Maraninchi, F.: Modeling Power Consumption and Temperature in TLM Models To cite this version : HAL Id : hal-01339441 Modeling Power Consumption and Temperature in TLM Models. Leibniz Transactions on Embedded Systems **3**(3), 0–29 (2016)
28. Nasser, Y., Lorandel, J.: RTL to Transistor Level Power Modelling and Estimation Techniques for FPGA and ASIC : A Survey. IEEE TCAD Circuits and Systems pp. 1–15
29. Nasser, Y., Prévotet, J.C., H elard, M.: Power modeling on FPGA: A neural model for RT-level power estimation. 2018 ACM International Conference on Computing Frontiers, CF 2018 - Proceedings (1), 309–313 (2018). DOI 10.1145/3203217.3204462
30. Nasser, Y., Sau, C., Pr evotet, J.C., Fanni, T., Palumbo, F., H elard, M., Raffo, L.: Neu Pow: Artificial Neural Networks for Power and Behavioral Modeling of Arithmetic Components in 45nm ASICs Technology. ACM International Conference on Computing Frontiers 2019, CF 2019 - Proceedings pp. 183–189 (2019). DOI 10.1145/3310273.3322820
31. Nasser, Y., Sau, C., Pr evotet, J.C., Fanni, T., Palumbo, F., H elard, M., Raffo, L.: NeuPow: A CAD Methodology for High-level Power Estimation Based on Machine Learning. ACM Transactions on Design Automation of Electronic Systems **25**(5) (2020). DOI 10.1145/3388141. URL <https://doi.org/10.1145/3388141>
32. Orcioni, S., Giammarini, M., Scavongelli, C., Vece, G.B., Conti, M.: Energy estimation in SystemC with Powersim. Integration, the VLSI Journal **55**, 118–128 (2016). DOI 10.1016/j.vlsi.2016.04.006. URL <http://dx.doi.org/10.1016/j.vlsi.2016.04.006>
33. Paniego, J.M., Libutti, L., Puig, M.P., Chichizola, F., De Giusti, L., Naiouf, M., De Giusti, A.: Unified Power Modeling Design for Various Raspberry Pi Generations Analyzing Different Statistical Methods. In: Communications in Computer and Information Science (2020). DOI 10.1007/978-3-030-48325-8-4
34. Perleberg, M.R., Goebel, J.W., Melo, M.S., Afonso, V., Agostini, L.V., Zatt, B., Porto, M.: ASIC power-estimation accuracy evaluation: A case study using video-coding architectures. 9th IEEE Latin American Symposium on Circuits and Systems, LASCAS 2018 - Proceedings pp. 1–4 (2018). DOI 10.1109/LASCAS.2018.8399919
35. Piscitelli, R., Pimentel, A.: A high-level power model for MPSoC on FPGA. IEEE Computer Architecture Letters **11**(1), 13–16 (2012). DOI 10.1109/L-CA.2011.24
36. Reimer, A., Schulz, A., Nebel, W.: Modelling macromodules for high-level dynamic power estimation of FPGA-based digital designs. Proceedings of the International Symposium on Low Power Electronics and Design **2006**, 151–154 (2006). DOI 10.1145/1165573.1165609
37. Richa, M., Pr evotet, J.C., Dardaillon, M., Mrou e, M., Samhat, A.E.: An Automated and Centralized Data Generation and Acquisition System. In: 28th IEEE International Conference on Electronics, Circuits, and Systems (ICECS), pp. 1–4 (2021). DOI 10.1109/ICECS53924.2021.9665490. URL <https://ieeexplore.ieee.org/document/9665490/>
38. Rogers-Vall e, M., Cantin, M.A., Moss, L., Bois, G.: IP characterization methodology for fast and accurate power consumption estimation at transactional level model. Proceedings - IEEE International Conference on Computer Design: VLSI in Computers and Processors pp. 534–541 (2010). DOI 10.1109/ICCD.2010.5647622
39. Shorin, D., Zimmermann, A.: Formal description of an approach for power consumption estimation of embedded systems. In: 2014 24th International Workshop on Power and Timing Modeling, Optimization and Simulation, PATMOS 2014 (2014). DOI 10.1109/PATMOS.2014.6951890
40. Streub uhr, M., Rosales, R., Hasholzner, R., Haubelt, C., Teich, J.: ESL power and performance estimation for heterogeneous MPSoCs using SystemC. In: Forum on Specification and Design Languages (2011)
41. Varma, A., Debes, E., Kozintsev, I., Klein, P., Jacob, B.: Accurate and fast system-level power modeling: An XScale-based case study. ACM Trans. Embed. Comput. Syst. **7**(3), 1–20 (2008). DOI 10.1145/1347375.1347378
42. Wang, L., Wang, X., Wang, T., Yang, Q.: High-level power estimation model for SOC with FPGA prototyping. Proceedings - 4th International Conference on Computational Intelligence and Communication Networks, CICN 2012 pp. 491–495 (2012). DOI 10.1109/CICN.2012.124
43. Zhai, J., Bai, C., Zhu, B., Cai, Y., Zhou, Q., Yu, B.: McPAT-Calib: A RISC-V BOOM Microarchitecture Power Modeling Framework. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems pp. 1–1 (2022). DOI 10.1109/tcad.2022.3169464. URL <https://ieeexplore.ieee.org/document/9761982/>