2014 24th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)

VPPET: Virtual Platform Power and Energy Estimation Tool for Heterogeneous MPSoC based FPGA Platforms

Santhosh Kumar Rethinagiri*, Oscar Palomar*†, Javier Arias Moreno*, Osman Unsal* and Adrian Cristal*†‡
*Barcelona Supercomputing Center (BSC), Barcelona, Spain
†Universitat Politècnica de Catalunya, Barcelona, Spain
‡IIIA - CSIC - Spanish National Research Council, Spain

***Abstract*—Using low-power symmetric multi-cores on FPGAs are becoming ubiquitous in embedded computing. This is due to the emergence of power and energy as key design metrics, as important as performance. This leads to the requirement of powerful and reliable tools, which will be used for the Design Space Exploration (DSE) based on power and energy at an early stage of the design flow. In this paper, we propose a simulation based virtual platform power and energy estimation tool for heterogeneous Multiprocessor System-on-Chip (MPSoC) based platforms. There are two steps involved in this tool development. The first step is power model generation. For the power model development, we used functional parameters to set up generic power models for different parts of the system. This is a one-time activity. In the second step, a simulation based virtual platform framework is developed to accurately grab the activities used in the related power models generated in the first step. The combination of the two steps leads to a hybrid power estimation, which gives a better trade-off between accuracy and speed. The proposed tool is automated and also scalable for exploring complex embedded multi-core architectures. The efficiency of the proposed tool is validated through multi-cores/processors designed around the FPGAs and extended to accommodate futuristic multi-processors/cores for a reliable energy based DSE. The obtained power/energy estimation results provide less than 4% of error for single core processor, 8% for dual-core processor and 9% for heterogeneous MPSoC based systems when compared to real board measurements.**

I. Introduction

Embedded applications such as video encoding & decoding, software defined radio (SDR), webcasting, etc. are becoming more complex and resource demanding. The computational requirements of such applications are substantial in order to meet real-time constraints and to ensure high quality of services. For these reasons, embedded hardware industries are shifting towards complex heterogeneous Multiprocessor System-on-Chip (MPSoC) architectures as a promising solution to leverage the potential parallelism inhered by complex embedded applications. An example of such commercialized platform is Xilinx Zynq series which embeds low power ARM Cortex-A9 MPcore processors and reconfigurable hardware blocks (CLB). These platforms fulfil the performance need of those complex applications but in turn increase the power consumption of the system. Lately, *ITRS*[1] predicted that power and energy issues are expected to get worse as we move to the next technology nodes. Similar to ITRS, *HiPEAC*[2], an European embedded consortium also released their roadmap-2013 stating that increasing the number of components on a chip, combined with decreasing energy scaling, is leading to the phenomenon of "dark silicon" [8]. These issues have led to the proposal of energy efficient systems with the right combination of different components. There are few tools available in the literature which address the issue of Design Space Exploration (DSE) in the design flow. These tools consider performance as their main design factor without considering power and energy. Now, power and energy have become key design constraints as vital as performance. For this reason, we propose a virtual platform based tool to estimate and optimize power and energy of these complex systems in an earlier phase of the design flow.

At the system-level, the power modeling process is centered around two aspects: the power model granularity and the main activity characterization. The first aspect concerns the granularity of relevant activities on which the power model relies. It covers a large area that starts from simple logic gate switching and stretches out to complex coarse-grain level like the hardware component events. In general, fine-grain power estimation yields a more accurate model with data but it is a time consuming task for system-level designer to handle technological parameters for a rapid DSE. Whereas, coarse-grain power models depend on micro-architectural activities that cannot be determined easily due to the complexity of the system. The second aspect involves the characterization of the activities, which requires a huge number of micro-benchmarking experiments and thus a significant time to extract the power models. The above described aspects yield to the definition of the power model that can be represented by a set of analytical functions or a table of consumption values (LUT). The selected power model granularity depends on the target abstraction level and user requirements in terms of estimation accuracy and speed. In the power estimation process, the developed power models interact with system-level environment in order to grab strictly relevant data depending on the design step. There are two challenges that need to be addressed and they are: first, *what is the appropriate power modeling methodology suitable for MPSoC system-level design that can offer a better trade-off between the time needed to generate the power model and its corresponding accuracy?* Second, *what is the appropriate simulation technique and the abstraction level suitable for rapid MPSoC prototyping and*

[1] $http://www.itrs.net/Links/2012ITRS/Home2012.htm$

[2] $http://www.hipeac.net/system/files/hipeac\_roadmap1\_0.pdf$

978-1-4799-5412-4/14/$31.00 ©2014 IEEE

for accurately extracting the activities for the developed power model (the first challenge)?

To answer the above challenges, we propose an efficient power/energy estimation tool for consumption estimation of heterogeneous MPSoC platforms. The idea here is to develop a virtual platform based power/energy estimation tool, which combines functional-level power models with an Instruction-Accurate (IA) simulation technique for rapid prototyping and fast power estimation. As shown in Fig. 1, functional power modeling part is coupled with a *transactional virtual platform* (OVPSim [1]) simulator in order to obtain the needed functional activities for the power models, which allows us to get a good trade-off between accuracy and speed with reduced modeling effort. In addition, with this tool designers are able to implement and simulate system-level designs of both processors and hardware accelerators together.

This paper is organized as follows. After Section II which presents the related works, Section III briefly describes the proposed estimation methodology. In Section IV, the power modeling methodology is applied to processors and hardware accelerator designed around Xilinx boards. To evaluate the proposed tool in terms of accuracy, experimental results are presented in Section V.

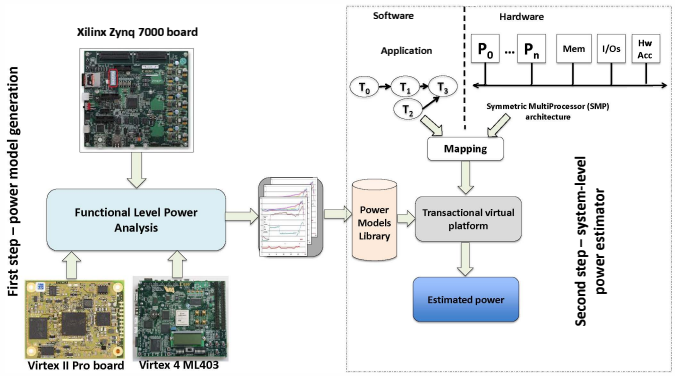

Fig. 1. Power estimation methodology flow

II. Related works

Significant research efforts have been devoted to develop tools for estimating power and energy consumption at different abstraction levels in the design flow of the embedded system [3]. Xilinx Power Estimator (XPE) [19] is used as a primary tool to estimate power for FPGA based Xilinx platform. XPE is a spreadsheet based tool and tightly coupled with the Vivado[3] design to estimate power for future generation products at the design cycle. The drawback of this tool is that it is not able to predict application based dynamic power. Beside, our tool simulates the full application and estimates the dynamic power of an application. In order to achieve a better trade-off between power estimation speed and accuracy, several studies have been proposed in the literature to evaluate system power consumption at higher abstraction levels [2]. In an attempt to reduce simulation time, recent efforts have been done to build up fast simulators using *SystemC* and *Transaction Level Modeling* (TLM). Nevertheless, power estimation at the TLM level is still under research and is not well established. Recently, TLM Power3 [10] power estimation methodology was proposed. This methodology was primarily developed for OpenRISC platform and suffers in terms of heterogeneous architectures as the system developers have to build the power models from scratch. This methodology did not report running a full application on the simulator and estimating the dynamic energy for the state-of-the-art heterogeneous MPSoC platform. Brandolese et al. [6] described an area estimation methodology for design space exploration for FPGA design at the System-Level. This methodology is based on two-level approach designed in behavorial style and the area is expressed in terms of FF's and LUT's. This methodology is not generic as it uses specific in-house tools for their development.

Recently, Bouhadiba et al. [5] proposed a system-level simulation method including functionality and power consumption estimation for softcore based FPGA circuit but their approach suffers in terms of speed as they use Cycle-Accurate (CA) simulation. To overcome this drawback the *Functional Level Power Analysis* (FLPA) was proposed [12], which relies on the identification of a set of functional blocks that influence the power consumption of the target component. The model is represented by a set of analytical functions or a table of consumption values which depend on functional and architectural parameters. Once the model is built, the estimation process consists of extracting the appropriate parameter values from the design, which will be injected into the model to compute the power consumption. However, when complex hardware or software components are involved, some parameters may be difficult to determine with precision. This lack of precision may have a non-negligible impact on the final estimation accuracy. Lately, McPAT [9] and GPUWattch [13] were proposed. These power estimation tools take the output of the performance simulator as their input and predict the power and energy of the system. However, these tools do not predict per task power and energy of the applications and also lack FPGA power estimation.

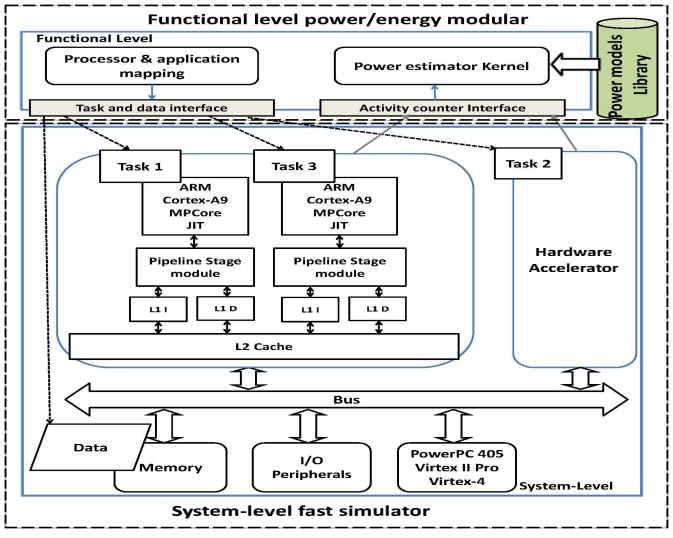

Fig. 2. Standalone power estimator

To overcome this disadvantage, a hybrid power estimation methodology (HSL) was proposed in [18] by combining Instruction Set Simulator (ISS) with functional level power

[3]http://www.xilinx.com/products/design-tools/vivado/

model but this methodology suffers in terms of porting the application and it needs significant amount of estimation time as it uses the interpreted ISS for simulation. Similar to previous work, a signature-based power model for MPSoC on FPGA [14] was proposed. Today, the Open Virtual Platform by Imperas Inc. [1] uses the same level of simulation but also tackles the simulation speed problem by proposing the OVPSim simulator since processors are not ISS but use code morphing and Just-In-Time (JIT) compilation. This technique is used in our tool. In this work, we propose to couple the OVPSim simulator with the functional level power models which offers reasonable trade-off between estimation speed and accuracy. In our previous work [15] [17], we introduced this methodology for power estimation of mono-processor based platforms and proved to be accurate and faster than the state-of-the-art tools ant this methodology was extended to accommodate DSP processor based platforms [16]. In this paper, we propose power/energy estimation for complex multi-core/processor based heterogeneous platforms at system-level and also propose a reliable design space exploration based on energy for the applications involving heterogeneous architectures. To the best of our knowledge, this is the first work that proposes simulation based power/energy estimation and DSE tool for heterogeneous platform which includes FPGA based hardware accelerators for the state-of-the-art platform at the system-level.

III. Power estimation methodology

This section describes our power estimation methodology that is divided into two steps as shown in Fig. 1. The **first step** concerns the power model development for the system functional components. In our framework, the FLPA methodology is extended to develop generic power models for different target platforms. The main advantage of this methodology is to obtain power models, which rely on the functional parameters of the system with a reduced number of experiments. As explained in the previous section, FLPA comes with few consumption laws, which are associated with consumption activity values of the main functional blocks of the system. The generated power models have been adapted to system-level design, so the required activities can be obtained from a system-level environment. For a given platform, generation of power model is a one-time activity. To estimate the power consumption of a platform, the first part is to divide the architecture into different functional blocks and to assign a parameter depending on the functionality of the block.

There are two types of functional parameters: *algorithmic parameters* that depend on the executed algorithm (typically, instruction per cycle (IPC) and cache miss rate (γ) for a processor and area utilization (α) for a hardware accelerator) and *architectural parameters* that depend on the component configuration set by the designer (typically, clock frequency, bus frequency and number of processor cores). For instance, Table I presents the common set of parameters of our generic power model.

The second part is the characterization of the embedded system power consumption by varying the parameters. These variations are obtained by using micro-benchmarking assembly and C programs (called scenarios) to stimulate each functional block separately. Characterization is performed by taking measurements on real boards. Finally, a curve fitting of the graphical representation allows us to determine the power consumption models by regression. The analytical form or a table of values expresses the obtained power models. In our work, this approach has been adapted to model power consumption for processors, memory system, reconfigurable hardware and I/O peripherals as it has been proved to be fast and precise [12].

TABLE I. Generic power model parameters

	Name	Description
Algorithmic	τ	External memory access rate
	γ	Cache miss rate for a processor
	IPC	Instruction Per Cycle
	α	area utilization for a CLB
Architectural	$F_{processor}$	Frequency of the processor
	F_{bus}	Frequency of the bus
	N	Number of cores

The **second step** of the methodology defines the main part of our tool that includes the *functional level power /energy modeler* and *fast IA simulator* as shown in Fig. 2. The functional power modeler evaluates the consumption of the target system with the help of elaborated power models from the **first step**. It takes into account the architectural parameters (e.g. frequency, number of processors, processor cache configuration, etc.) and the application mapping. It also requires the different activity values of algorithmic parameters on which the power models rely. In order to accurately collect the needed activity values, the functional power modeler communicates with a JIT/SystemC simulator at Instruction-Accurate (IA) level. The combination of the above two components described at different abstraction levels (functional and IA) leads to a standalone hybrid power estimation tool that gives a good trade-off between accuracy and speed.

The vital functionality of this tool is to offer a detailed power analysis by means of a complete simulation of the applications on heterogeneous MPSoC prototypes. This process is initiated by the functional power modeler through *the data and task interface* (Fig. 2). In this way, the mapping information is transmitted to the JIT simulator. ***Our simulator consists of processors and hardware components which are instantiated from the OVP library and few other models such as memory controller, ICAP, configurable logic blocks (CLB), bus controller which are written in SystemC to build a virtual prototype of the target system.*** We would like to highlight ***two points***: ***First***, that processors are described using IA level that sequentially execute the instructions and have no notion of concurrency of micro-architecture. ***Second***, the whole simulation prototype of Zynq, Virtex 4 and Virtex II Pro was build from scratch by using SystemC programming language.

In the power estimation step, the simulator collects the activities that are influenced by the application and the input data. ***At the end of the simulation, the values of the activities such as IPC, cache miss rate, area occupied, bus accesses, number of slices and number of switching are transmitted to the power estimator kernel using the activity counter interface in order to calculate the global power/energy consumption of the chosen platform as illustrated in Fig. 2***.

The following section will discuss the **first part** i.e., the elaboration of the power model for the Xilinx Zynq, Virtex-4

TABLE II. GENERIC POWER MODELS

Processors	Power models
ARM Cortex-A9 (single core)	P(mW) = 0.59 $F_{processor}$ + 0.39 IPC + 0.19 (γ1) + 0.50 (γ2) + 8.93
ARM Cortex-A9 (dual core)	P(mW) = 0.63 $F_{processor}$ + 1.4 (γ2) + b $\sum_{i=1}^{2}(\gamma 1_{c1-c2})$ + c $\sum_{i=1}^{2}(IPC_{c1-c2})$ + 12.45
Virtex II Pro PowerPC	P(mW) = 4.1 (γ) + 6.3 $F_{processor}$ + 1599
Virtex-4 PowerPC	P(mW) = 8.546 (γ) + 9.3 $F_{processor}$ + 836

& Virtex II Pro FPGA platforms by using FLPA methodology.

IV. POWER MODEL GENERATION

In order to prove the effectiveness of the proposed power estimation methodology, we used Xilinx platforms (Zynq, Virtex-4 and Virtex II Pro). The Zynq contains a dual core ARM Cortex-A9 processor that has 32 KB, 4-way set associative instruction and data caches for each core and a common 256 KB L2 cache. The Virtex II Pro and Virtex-4 FPGAs contains two PowerPC405 processors that have a 16KB, 2-way set associative and 32 KB, 4-way set associative instruction and data caches respectively. In addition, these FPGA boards have a large number of CLB for implemention of hardware accelerators. As explained above, we extended the FLPA methodology to generate generic power models for the target platform. As a **first step**, we divided the architecture into different functional blocks such as processor, pipeline stage unit, memory system, reconfigurable logic, etc. Then, we started the characterization of each component in order to extract the related power consumption models.

Processor power model: Table II shows the power consumption models for the ARM Cortex-A9 processor and its memory system. The input parameters on which the power models rely are the frequency of the processor ($F_{processor}$), IPC ($0 \leq IPC \leq 2$), and the cache miss rate ($0 \leq \gamma\_1$ & $\gamma\_2 \leq 100$ (%)). The system designer chooses the frequency of the processor and bus while cache miss rate and IPC are considered as activities of the processor, which could be extracted from the simulation environment.

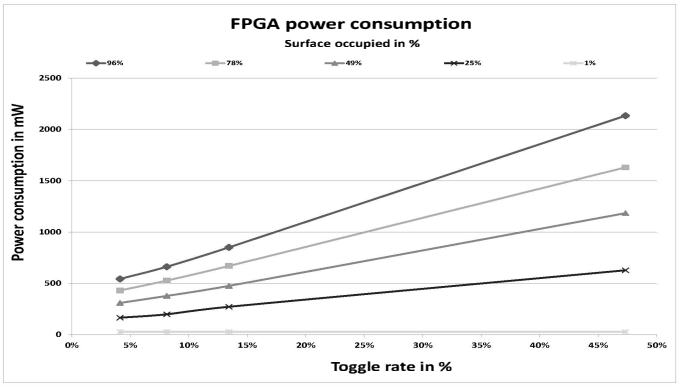

Fig. 3. FPGA power consumption for different surfaces occupied for Zynq board

FPGA power model: A power model has been built for the reconfigurable part of the FPGA component on the Zynq/Virtex-4/Virtex II Pro boards. For a given FPGA device,

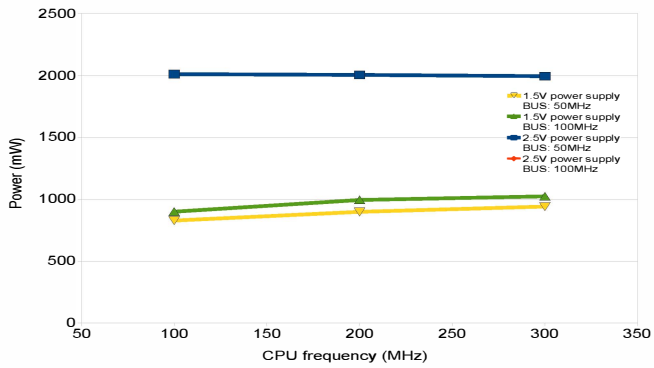

Fig. 4. Mutex power consumption

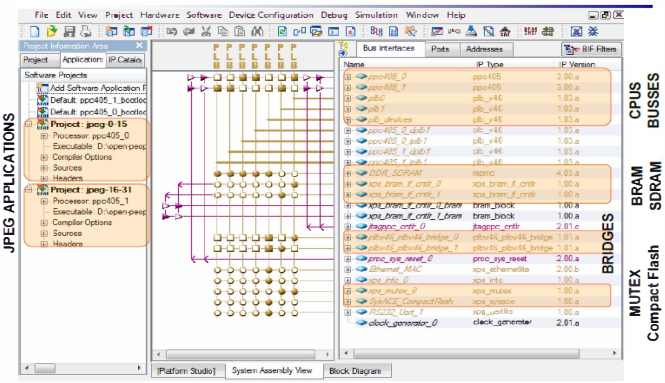

Fig. 5. Xilinx EDK 10.1 design for two PowerPC processors with shared memory

the parameters that can be extracted from the high-level specifications are frequency F, switching rate β and utilized area α of targeted FPGA. Using a high-level architecture synthesis tool such as GAUT [7], these parameters can be predicted with good estimates. According to the experimental results, the model does not come as a multi-linear equation of the above-mentioned parameters. For this reason, an automated 3 entries lookup table of consumption values written in C is used. The power is estimated by interpolation of the above mentioned 3 input parameters. For instance, Fig 3 illustrates the variation of the FPGA power consumption according to area utilization by changing the toggle rate.

```
for i = 0 to N
  mutex.lock()
    jpeg.load("image" + i + ".bmp")
  mutex.unlock()

  jpeg.compress()

  mutex.lock()
    jpeg.save("image" + i + ".jpeg")
  mutex.unlock()
```

Fig. 6. JPEG mutex implementation between the two PowerPC processors

Extrapolation for heterogeneous multiprocessor architectures: The above developed power models will be used in the framework of system-level estimation of heterogeneous

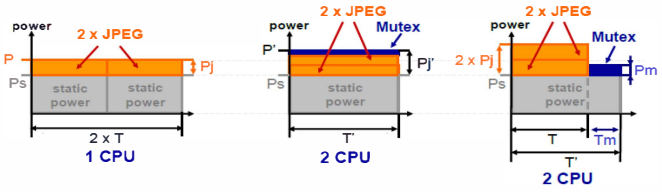

Fig. 7. Power/time/energy consumption and measurement

multiprocessor architectures that may contain several processors and hardware accelerators. This approach is mandatory in the design flow for two reasons. First, system-level estimation can be achieved with an acceptable accuracy and 10-1000x faster than the lower-levels. Second, it allows exploring architectures that cannot be implemented due to hardware resource limitation or unavailability of the platform. For instance, we cannot exceed dual-core ARM Cortex-A9 based architecture using our Zynq platform and similarly for two PowerPC multiprocessor based Virtex-4 and Virtex II Pro platforms. Thus, it is important to have a scalable approach to address the complex system power estimation issue. We mention here that it is necessary to compute the energy before the deduction of the total power consumption. Equation 1 gives the total energy consumption of the platform. The parameters used are the processor (E_{p_i}) and the conventional configurable logic blocks (E_{CLB}). The equation also involves the energy consumption of the synchronization part (E_{sync}) required to access the shared memory (E_{mem}), the shared I/O resources ($E_{I/O}$) and energy variation depending on the number of switching ($E_{switching}$). Whereas, n and m are the total number of processor cores and hardware accelerators (CLB) respectively. In our FPGA platform, synchronization between parallel tasks running on different processors or hardware accelerators is performed by a call to a hardware mutex and its power consumption as shown in Fig. 4.

$$E_{total} = \sum_{i=1}^{n} E_{p_i} + E_{mem} + E_{sync} + E_{I/O} + \sum_{j=1}^{m} E_{CLB} + E_{switching} \quad (1)$$

Fig. 5 shows that the two PowerPC processors are configured inside the Xilinx EDK platform with a shared memory (RAM) and their configuration with the Processor Local Bus (PLB). There are two bridges configured to connect the two PLB buses of the two processors. In order to run the application, we use two methods. First, we split the application into two different parts and then run it on the two different processors, where first processor starts the application and second processor completes the application. Second, as shown in Fig. 5 there are two JPEG applications ported on the two processors to run independently. These two processors are synchronized by using the mutex call as shown in Fig. 6.

While running the application on the Virtex-II Pro board, we measured the power across the jumper and its details are shown in Fig. 7. From Fig. 7, we can notice that there are three different measurements from left to right. First one from the left denotes the power measured across the Virtex-II Pro running two applications with one processor and it take twice the time (2 x T), Ps denotes the static power and Pj denotes the dynamic power running the board. Second one denotes two different PowerPC running two different JPEG applications, here static power remains the same Ps, while there is a change in the dynamic power (Pj') and reduced time (T'). In the third one, JPEG application is split to run on both the processors and here mutex power comes into play (Pm). From this figure, we are able to notice that processor power increases linearly with a addition of synchronisation power to the power model and its memory and I/O devices if any.

In our platforms, synchronization between parallel tasks running on different cores or hardware accelerators is performed by a call to pthread or hardware mutex. Several experiments have been conducted to evaluate the additional power cost of this hardware component. This study includes three parameters, which are number of processors/cores, processor and bus frequencies. Experimental results show that the mutex power consumption depends mainly on AXI bus frequency for Zynq and PLB for Virtex platforms.

So far, power models for the single core and dual core processor, FPGA and heterogeneous multiprocessor architecture have been developed. Estimation of the overall power consumption for single core processor, dual-cores/processors and heterogeneous multiprocessor architecture at system-level will be briefed with the results in the next section as the **second step** of power estimation methodology with the help of different benchmarks.

V. Virtual Platform Power Estimation Results

In the **second step**, system-level prototypes of ARM Cortex-A9 and PowerPC405 based architectures have been developed. This prototype uses different SystemC models especially the JIT simulator provided by OVP for the target processor. Furthermore, cache parameters, pipeline stage unit and bus latencies are set to emulate the real platform behaviour. A set of counters are injected into the simulator to determine the values of different IPC and cache miss rates: read data miss, write data miss and read instruction miss.

In the next step, we estimate the total power consumption of each task using the power models shown in Table II (single core). The first study is to prove that our tool is accurate. For the evaluation of the accuracy at system-level, we simulated various benchmarks available in the industry on our tool for PowerPC mono-processor and ARM Cortex-A9 single core processor as shown in Fig. 8. It shows that our tool exhibits a negligible maximum error equal to 5% compared to the real board measurements for single-core/processor based platform. This is due to better accuracy of the captured activities in the simulator.

A. Dual-core and homogeneous multiprocessor architecture

The second study involves a processor architecture with identical cores/processors to run the multimedia benchmarks. All the cores/processors execute the same workload. Fig. 9 reports the total energy consumption in J. Compared to real board measurements, our tool achieved a maximum error

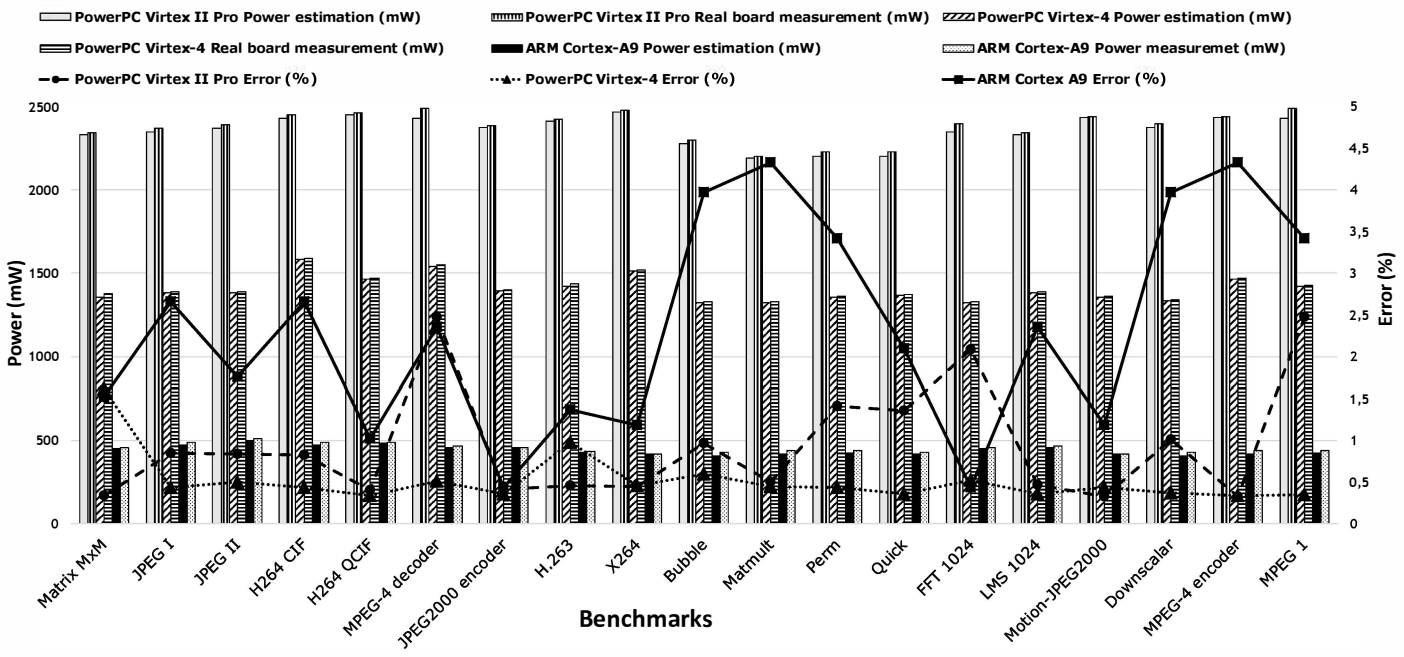

Fig. 8. Power estimation of different benchmark applications running on a single-core ARM Cortex-A9 processor and a single core PowerPC processor

of 9% for both PowerPC and ARM based multi- processor/core architectures. This accuracy is obtained because of two main reasons. First, power models are extracted from real board measurements. Second, additional activities that are intrinsic to parallel processing such as synchronization and communication overheads are accurately evaluated by using our IA simulator. The above-mentioned reasons encourage us to consider architectures with a higher number of cores in the context of exploring new complex processor architectures. To quantify VPPET in terms of power estimation accuracy, we compared it with McPAT [9], a widely used state-of-the-art tool for power estimation for the same processor architecture cores. The average estimation error of McPAT [9] is 30% when compared to real board measurements and 6x slower than VPPET while running with Gem5 [4] simulator for all the processor cores and applications.

B. Heterogeneous multiprocessor architecture

In this section, we emphasize the benefit of our tool in the context of heterogeneous architecture. In general, the choice of a hardware accelerator is driven principally by the performance requirements of the application and the processor usage of each task. In order to implement heterogeneity, we developed SystemC models of CLB and an arbiter. Furthermore, the configuration access controller is embedded with the CLB model, which is similar to ICAP controller in the FPGA platform. The main functionalities of this CLB model are to execute program as per the processor's instruction, to access the memory and to issue an acknowledgement to the processor when the task is completed. At the end of the simulation, the simulator will provide the execution time of the tasks running on the hardware accelerators, number of slices occupied by the task and number of switching to the power estimator kernel to calculate the total energy. This simulator is similar to perfecto [11] with enhanced multi-core processor IP and AXI 4 bus with split transaction and burst enabled.

We use MPEG 2 Part 2 application as benchmark. IDCT task is the most time consuming task in this application. Thus, it is selected to be implemented on a CLB. Various trade-offs can be done between the amount of consumed hardware resources, execution time and power consumption. This task is highly regular and has large repetition spaces in its multiple hierarchical levels. Such large repetition spaces allow us to fully exploit the existing partitioning in VHDL. The synthesized hardware occupies 5% of the FPGA board. According to the FPGA power model, the energy consumption of the chosen hardware motion estimation is around 450 mJ offering 50% of energy saving compared to the software execution and 40% of reduction in execution time. In terms of energy consumption, we observe that until a certain number of processors, the total system energy consumption decreases as the execution time is reduced, and then it tends to stabilize as the system performance improves. The issue of increasing the number of processors/cores over a certain limit tends to be ineffectual, as it just adds new conflicts at the shared memory and bus level, leading to more waiting cycles and this will be discussed in detail in the next DSE section. We compared our tool with XPE [19] for dynamic power estimation for the same processor architecture cores with FPGA hardware accelerators. The average estimation error of XPE [19] is 23% when compared to real board measurements.

C. Design space exploration based on energy

At present, the exploration phase in the design flow of an embedded system focuses more on multi-objective optimisation problems, which tries to identify a solution with the optimal function cost involving criteria such as time, area and power. In order to find the best implementation solution, a set of experiments have to be considered and evaluated. For the above mentioned reason, we tried to estimate the energy

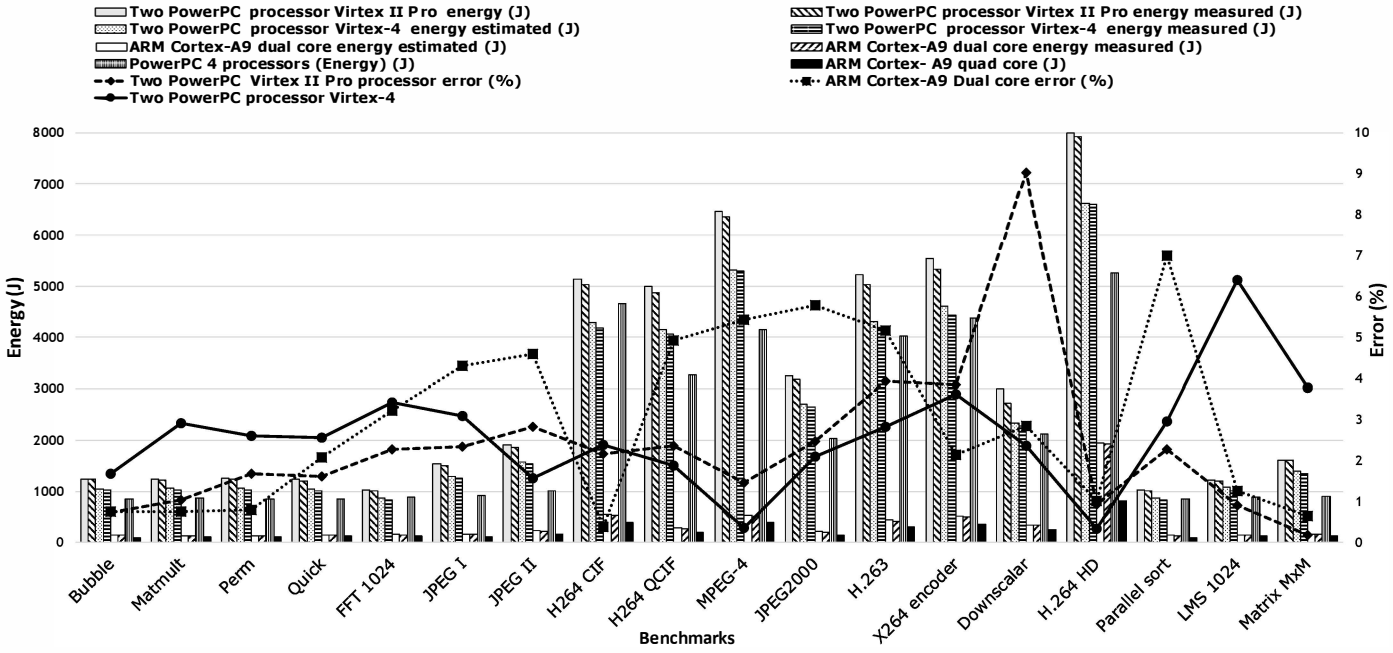

Fig. 9. Energy estimation of different benchmark applications running on a dual-core ARM Cortex-A9 processor and PowerPC multiprocessor

for different types of multimedia applications with 1 to 4 core processor ARM Cortex-A9 based-architecture with and without CLB. We used parallel algorithm to run the multimedia benchmarks on the different processor cores and hardware accelerators. Fig. 10 illustrates the detailed energy estimation results for different configurations with and without hardware accelerators and validated against the real board measurements in order to have a reliable DSE. From Fig. 10, we are able to extract several conclusions. Depending on the execution time, we are able to find the optimal architecture that could satisfy the application requirement in terms of computation rate. For instance, depending on the frame rate to achieve such as 15 or 30 frames per second (f/s), the H.264 decoder could be executed more or less efficiently. From the simulation results, 1, 2 and 4 processor cores based-architectures offer respectively a total performance of 9f/s, 15 f/s and 19 f/s and by adding FPGA block to the processing core makes the system more energy efficient. We conclude that using 2 processor cores are sufficient to reach a rate of 15 f/s, however the 4 processors based-architecture offer a more energy-efficient solution. Indeed, by increasing area cost, the total energy is decreased by 30% while moving from 2 to 4 processors. Nevertheless, this behaviour is not the same for all applications as shown in Fig.10. For example, sorting and JPEG applications using 4 cores instead of 2 cores tends to be less efficient due to communication constraints. Consequently, the corresponding additional area cost for 4 processor cores cannot be justified. We would like to state that adding FPGA resources will be energy efficient and scalable for applications having complex computational algorithm rather than simple application. The result of the DSE shows that for a simple application (JPEG and Sorting), it would be better to implement on a single core and some part on the FPGA as they give energy reduction of 35% to 40%. For the same configuration going ahead with multicore and FPGA is not efficient as communication and area constraint consume more power. Applications such as H.264 running with HD videos are better to be executed with multi-core processor combining with FPGA resources as they are 40% to 50% efficient than running on single core with FPGA. Adding FPGA to low power processors provides the best energy efficiency in almost all the situations and in some cases it is far better than other devices such as big FPGA boards. The maximum obtained error is around 9% for all the design implementations as shown in Fig. 10. As stated before, we cannot exceed more than dual-core based architecture using the Zynq platform and two processor for the Virtex-II-Pro. With this tool, we are able to predict for the future multi-core/processor architectures (4 cores) with and without CLBs and also with a reliable accuracy.

VI. Conclusion

This paper proposes a virtual platform based power and energy estimation tool for heterogeneous MPSoC based FPGA platform. First, a power modeling methodology has been defined to address the global system consumption that includes processors, reconfigurable hardware, etc. Secondly, the functional power models are coupled with a fast virtual platform simulation technique to obtain the needed micro-architectural activities for the power models, which allows us to reach a better trade-off between accuracy and speed. In addition, using functional power models brings transparency regarding low level implementation, reduces the number of dependent parameters and eases extraction of the required data. Experimental results show that our tool exhibits less than 4% average error compared with the real measurements. With the proposed tool, the designer can explore several implementation choices: single core, multi core and heterogeneous multiprocessor architecture based platforms.

As future work, we will focus on more complex heterogeneous architectures by adding softcore processors such as microblaze and ARM7 models to this tool and to perform

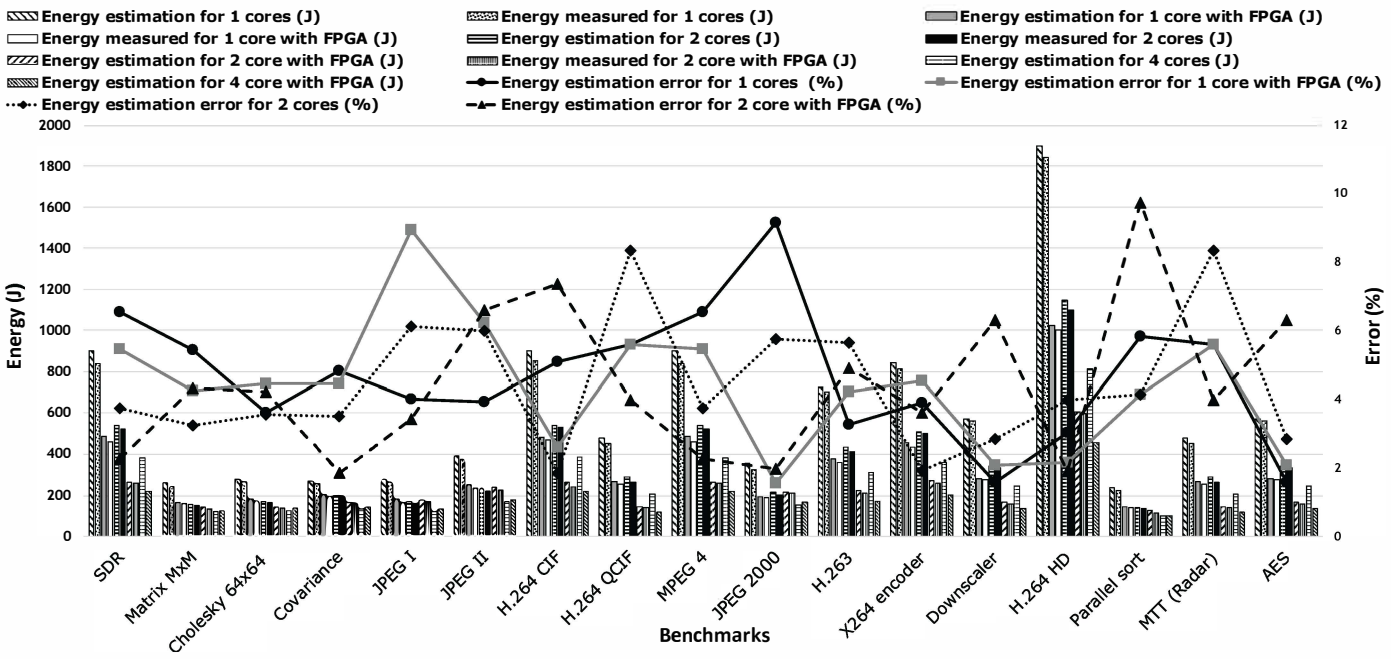

Fig. 10. Energy based design space exploration of ARM Cortex-A9 with FPGA hardware

automatic design space exploration. Furthermore, in order to obtain more accurate power estimations, some power model refinements must be realized. Another important extension will be to consider thermal and reliability aware model based design space exploration for futuristic platform at the system-level

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under the ParaDIME Project (www.paradime-project.eu), grant agreement no 318693.

REFERENCES

[1] B. Bailey. System level virtual prototyping becomes a reality with OVP donation from imperas. Technical report, EDA, June 2008.

[2] L. Benini, R. Hodgson, and P. Siegel. System-level power estimation and optimization. In *Proceedings of the 1998 international symposium on Low power electronics and design*, ISLPED, pages 173–178, New York, USA, 1998. ACM.

[3] L. Benini and G. d. Micheli. System-level power optimization: Techniques and tools. *ACM Trans. Des. Autom. Electron. Syst.*, 5(2):115–192, Apr. 2000.

[4] N. Binkert et al. The gem5 simulator. *In SIGARCH Comput. Archit. News*, 39(2):1–7, Aug. 2011.

[5] T. Bouhadiba, M. Moy, and F. Maraninchi. System-level modeling of energy in tlm for early validation of power and thermal management. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, pages 1609–1614, March 2013.

[6] C. Brandolese, W. Fornaciari, and F. Salice. An area estimation methodology for fpga based designs at systemc-level. In *Design Automation Conference, 2004. Proceedings. 41st*, pages 129–132, 2004.

[7] P. Coussy, C. Chavet, P. Bomel, D. Heller, E. Senn, and E. Martin. Gaut: A high-level synthesis tool for dsp applications. In P. Coussy and A. Morawiec, editors, *High-Level Synthesis*, pages 147–169. Springer Netherlands, 2012.

[8] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger. Dark silicon and the end of multicore scaling. In *Proceedings of the 38th Annual International Symposium on Computer Architecture*, ISCA '11, pages 365–376, New York, NY, USA, 2011. ACM.

[9] S. L. et al. Mcpat: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *MICRO 2009*.

[10] D. Greaves and M. Yasin. Tlm power3: Power estimation methodology for systemc tlm 2.0. In *Specification and Design Languages (FDL), 2012 Forum on*, pages 106–111, Sept 2012.

[11] P.-A. Hsiung, C.-S. Lin, and C.-F. Liao. Perfecto: A systemc-based design-space exploration framework for dynamically reconfigurable architectures. *ACM Trans. Reconfigurable Technol. Syst.*, 1(3):17:1–17:30, Sept. 2008.

[12] J. Laurent, N. Julien, and E. Martin. Functional level power analysis: An efficient approach for modeling the power consumption of complex processors. In *Proceedings of the Design, Automation and Test in Europe Conference*, Munich, 2004.

[13] J. Leng et al. Gpuwattch: Enabling energy optimizations in gpgpus. *SIGARCH Comput. Archit. News*, 41(3):487–498, June 2013.

[14] R. Piscitelli and A. D. Pimentel. A signature-based power model for mpsoc on fpga. *VLSI Des.*, 2012:6:6–6:6, Jan. 2012.

[15] S. Rethinagiri, R. Ben Atitallah, S. Niar, E. Senn, and J. Dekeyser. Fast and accurate hybrid power estimation methodology for embedded systems. In *Design and Architectures for Signal and Image Processing (DASIP), 2011 Conference on*, pages 1–7, Nov 2011.

[16] S. K. Rethinagiri, O. Palomar, J. Arias Moreno, O. Unsal, A. Cristal, and M. Biglari-Abhari. System-level power and energy estimation methodology for open multimedia applications platforms. In *VLSI (ISVLSI), 2014 IEEE Computer Society Annual Symposium on*, JUL 2014.

[17] S. K. Rethinagiri, O. Palomar, R. Ben Atitallah, S. Niar, O. Unsal, and A. C. Kestelman. System-level power estimation tool for embedded processor based platforms. In *Proceedings of the 6th Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools*, RAPIDO '14, pages 5:1–5:8, New York, NY, USA, 2014. ACM.

[18] S.-K. Rethinagiri, O. Palomar, O. Unsal, A. Cristal, R. Ben-Atitallah, and S. Niar. Pets: Power and energy estimation tool at system-level. In *Quality Electronic Design (ISQED), 2014 15th International Symposium on*, pages 535–542, March 2014.

[19] Xilinx. *Xilinx Power Estimator 2014.2*, Jan. 2014.