Journal of
Low Power Electronics
Vol. 5, 1–14, 2009

A Low Power Design Space Exploration Methodology Based on High Level Models and Confidence Intervals

Jalel Ktari* and Mohamed Abid

CES-National Engineering School of Sfax, Tunisia

(Received: 21 October 2008. Accepted: 9 February 2009)

Power consumption is nowadays a critical design constraint for circuits and systems. To efficiently guide early choices in the design flow, high-level estimations must be available. In order to address the different abstraction levels and the various targets, a global methodology is proposed here to elaborate suitable models. In this paper we are interested in exploring low consumption architectures in order to deduce that which meet(s) the constraints most. For this aim, several low power methodologies were established. They treat the energy consumption optimization problem at several levels especially on specific components like the hardware, or on the software, or on the communication or on the memory separately, and seldom on the whole system. However, as target architectures become complex, a global methodology that offers more efficient low power exploration becomes necessary. In fact, we propose a low power methodology based on rich performances models as well as a low power exploration technique. A complete model is proposed in order to deduce the total performances of the system according to the architecture and the application parameters. This model is used during exploration thanks to a technique based on simulated annealing. This technique makes it possible to exploit exploration according to several levels of granularity in order to be able to choose the level, which allows ensuring a precise and fast exploration. Moreover, an estimation frameworks and the mathematic approach are presented in this paper using a confidence interval.

Keywords: Low Power Design, Space Exploration Co-Design, High Level Models, Accuracy.

1. INTRODUCTION

Today's embedded system devices are targeting complex multi-media applications where there is a need for significant amount of technology and targets computing in order to achieve the various data processing tasks which could include both video and audio. This makes the design increasingly difficult by integrating a multitude of functionalities while respecting the constraints. Moreover, energy consumption has become one of the principal constraints since those applications are now running on small and mobile battery-operated systems. As the battery capacity growth is too slow (Eveready's law), it becomes crucial for those systems to achieve high performance and low power consumption at the same time. Reducing energy consumption permits also to minimize the thermal dissipation which increases the system reliability and avoids the use of noisy and cumbersome cooling systems.

To achieve these antagonist goals, several low power methodologies were established. The researchers deal with the energy consumption optimization problem at several levels. The efforts were often focused on specific components like hardware, or software, or communication or memory, but seldom on the architecture as a whole. However, the designer needs a more global methodology that offers more efficient low power exploration. He/she needs also methods and tools for estimating the system performance in order to extract the most promising architectural solutions and those which respect the constraints. However, a minority of works deal with this problem. In this paper, we present a low power exploration methodology. It is based on parametric models of performance and energy consumption. These models are exploited by an extensible exploration environment based on the simulated annealing heuristics. This environment makes it possible to extract an adequate solution respecting the various constraints. Thanks to the approach suggested, we have provided the designer with a decision-making system where he/she is guided in the choice of the architectural solution and the application parameters.

The paper is organized as follows. The next section presents the related works. The methodology and the

*Author to whom correspondence should be addressed.
Email: jalel.ktari@enis.rnu.tn

approach are addressed in Section 4. The exploration environment and the MPEG 2 results are presented in Section 5. The accuracy of the approach is treated in Section 6.

2. RELATED WORKS

Hardware/software techniques to reduce energy consumption have become an essential part of current system designs. Extensive researches on power optimization from circuit level to system level have been conducted in these recent years. Such techniques have targeted the memory system due to the prevalent use of data signal and video applications, which focus on exploiting cache to reduce power consumption. The work in Ref. [1] presented an architecture-oriented power minimization approach. In fact, a power simulation tool and performance simulation tool are used to do architecture-level optimizations. A framework for describing the power behavior of system-level designs was proposed by Ref. [2].

The availability of high performance application cores for System on Chip (SoC) devices, which make up these systems is an important portion of the electronic market and has attracted significant research interest. Moreover, in order to maximise the operating time provided by the battery and to satisfy the real time constraint, we need high level IP modeling. So we can maintain performance, low power constraints as well as that of the battery and real time. Some energy estimation tools and monitoring techniques can be presented. Energy simulators such as Wattch[3] and SimplePower[4] estimate the energy consumption in "reasonable" time.[5]

On RTL level, we can mention the DSP-PP,[6] a tool for simulation allowing the estimate of the power dissipated by DSPs. It is composed of two components: the simulator of performance on the cycle level (CPS) and the estimator of the dissipation of power (PDE). It is written in C++ making it possible to consider abstract models. The components of the DSP are modeled like objects integrating the model of consumption. DSP-PP considers the simulation on the cycles level of all the DSP's components: the ways of data and the interconnection and estimate the value of dynamic power, short-circuit of each component of the DSP.

Representative researches in measure-based estimation techniques are SES[7] and PowerScope.[8] SES is an energy-monitoring tool, which collects energy consumption data in a cycle-by-cycle resolution and maps the collected energy consumption data to program structure. PowerScope is based on hardware instrumentation by using a digital multimeter with support of embedded operating system. Therefore, PowerScope is applicable to ordinary embedded systems. EPRO[5] employs measure-based estimation techniques used in SES and PowerScope. However, ePRO is distinct from SES because ePRO does not need any extra hardware module such as profile acquisition module in SES.

Multi-objective optimization has been extensively addressed in system-level low power design and synthesis. Moreover several works treat the low power design of software or hardware targets separately. Indeed, in Ref. [9], a method for low power design for the processors is presented. Refs. [10] and [11] focus on DSPs, Ref. [2] focuses on the memories, Refs. [13] and [14] on FPGAs, Refs. [15] and [16] on communication buses. In those various works, the authors showed that there is an important profit of time to market for the designer, through the developed high-level models. Unfortunately, some of those proposed tools do not treat the architecture exploration and mapping in its entirety, which can be made up of various software and/or hardware resources running together.

Considering the diversity of the possible architectural solutions for an application, the designer needs some methodologies of low power exploration and estimation models of the overall system consumption. These models must take account of the various parameters which influence the performance.

Indeed, an application can have various performances on a given target by varying the algorithmic or architectural parameters. In addition, the majority of the existing partitioning software/hardware approaches do not consider the supplied maximum energy in the choice of the architectural solution.[17] Such a constraint can influence the system. Moreover, since the partitioning and the scheduling are dependent, neglecting the available energy can engender a non-schedulable solution. We thus need an estimate technique for the temporal performances, consumption and cost of the solution to do a low power design space exploration. The works presented in this paper treat more particularly an approach of low consumption exploration. There are many (DSP/FPGA) exploration tools which treat time, area and consumption constraints together like Mogac,[18] Cosyn-LP,[19] Ghali,[20] Codef-LP,[21] etc. The following table summarizes the characteristics of each tool (Table I).

The tools presented in this overview are not usually available, so we require the development of a low power exploration environment that integrates the models of performances. Thus the tool can be extended and enriched

Table I. Low power exploration tools.

	Consumption			
Tools	Hard	Soft	Communication	Total
Ghali	Xpower	Simple power	x	Each case evaluation
Codef-LP	Watt Watcher	Vestim Joule track	x	Sum
Mogac	Available models	Available models	Consumption/packet	Sum
Cosyn-LP	Available models	Available models	Available models	Sum

with models according to the needs. Moreover, most of these approaches explore predefined or monoprocessor architectures. This is the case in the Codef tool and the Ghali methodology. In addition, with the Codef-LP tool, we can explore the solutions area but without taking account of the communication consumption which can be significant.

So, the objective of this work is to define a low power exploration approach. The work consists in:

—Multi-granularity specification: it is a question of specifying the application and the constraints on several granularity levels. This permits more efficient solution exploration.

—Parametric consumption models set up: it consists in establishing parametric models of power which include the consumption of the whole system {hardware + software + communication}. In fact, the application and the architecture parameters will be considered in the performances models in order to have rich models.

—Effective low power architecture exploration: it is about being able to choose the adequate architectural solution where the number of resources to be exploited is not well known. Indeed, the designer can be confronted with the problem concerning the choice of the resources numbers, for instance whether the application needs two or three DSPs.

The next section treats the methodology of elaborating power and performance models as well as the cost model. We also introduce the estimation method and the design space exploration technique.

3. CONTRIBUTION

The contribution in this work consists in:

A. The increase in the abstraction level of performances models in particularly consumption in order to estimate the latter at the beginning of the design flow. Thus we can define and resize the system according to the objective and the needs, which avoids the returns to initial specifications to adjust them according to new recommendations. This also allows reducing the time and the cost of the design. Due to the existence of a compromise between the results precision and the fastness of the exploration, the estimation models granularity is taken into consideration. For this, we offer a methodology to extract a rich models estimation of the system with the aid of high-level parameters. This methodology considers several factors having an influence on consumption:

1. Application parameters: size of picture, resolution, chrominance, the filter order, etc.
2. Architectural and technological parameters: frequency, Vcc, the target, etc.

B. To explore the solutions space and to be able to extract efficient and realistic solutions quickly, an approach and a low power exploration environment are developed. In fact, the analysis of all the possible architectural combinations is an N-P hard problem. It is useful to practice a heuristic to reduce the solutions area. The suggested approach allows the designer to estimate the total performance of the application based on a library of parametric models. In fact, this approach allows to:

1. Consider all the architecture by taking into account its various components (software, hardware, communication, memory). Thus we have the possibility of enriching the available exploration tool by the different models parameters.
2. Evaluate the system's global performance at the beginning of the design flow using the application performance models and by exploiting models of architectural performances.
3. Explore automatically the architectural solutions space thanks to a heuristic, which extract a solution that answers particularly the low power objectives.
4. Parameter the architecture and the dimension of the solutions space to be explored in order not to be limited to a fixed and/or predefined architecture. In fact, in order not to oblige the designer to exploit a fixed or a predefined architecture, the approach suggested rests on a parametric multiprocessor architecture where the number of components to be exploited is not fixed. It is the exploration methodology that will choose the appropriate solution and thus fix the number of useful components in an attempt to answer the need.

4. GENERAL APPROACH

The low power space exploration requires some information relating to the application on one hand and to the models and performances on the other hand. In fact, it is a question of deducing necessary information for exploration. Moreover, we need estimation and performances models, which are rich and parametric to explore the space well.

This section treats successively the graph and the architectural model, the methodology of elaborating power and performances models as well as the cost model. We also introduce the estimation method and the design space exploration technique.

4.1. Task Graph and Target Description

—The specification model must allow a functional description of the whole application while being independent from its final implementation. Concerning the application specification, it is often represented by a task graph.[17, 18, 21–23] This representation makes it possible to model the tasks as well as the inter-tasks dependencies of the application. In the suggested approach, we start from an application specification in the form of a directed acyclic graph of tasks (DAG). In this graph, the nodes

represent the tasks *Ti* of the system and the dependences between them are represented by arcs. We associate to each arc *Aij* of the graph the quantity of data which the task *Ti* must transfer to the *Tj* task. The task graph specification is not necessarily the best. But since in this research, many granularity levels (system, task, function) are exploited, the task graph is adequate. In fact it permits to model the performance and the consumption and to do the hardware/software partitioning. Moreover, if there is task dependency, Blazewicz/Chetto method allows to solve this problem.

—The architecture will be heterogeneous (mainly software (TI DSP C6201, C55, C67) and hardware: FPGA) in the form of discrete components (DSPs & FPGA) communicating via a buses and having a shared memory.

4.2. Methodology

Our methodology is presented in Figure 1. The approach rests on a task graph specification of the application (Fig. 1(A)). The parameters and performances knowledge is necessary for every task present in the application specification. Concerning the tasks power estimation (Fig. 1(B)), every task is evaluated by measurement or by estimation tools in terms of time and consumption according to the target (each DSP and FPGA) and according to its parameters.

The following stage (Fig. 1(C)) consists in elaborating a library of performance, power models of the application for the various tasks on various targets. Examples of performances library models are presented in our previous works.[24] These models can be recovered manually through direct measurement or through simulation using software[17] and/or hardware[14] estimation (Fig. 1(B)) tools.

The following stage has to do with choosing the architectural solutions (Fig. 1(D)). It is based on the analysis of the available solutions and the retrieval of the adequate architecture. The solution analysis consists in assessing every solution and estimating its performance and its consumption (Fig. 1(E)). Following the analysis of various solutions, it is necessary to retrieve the adequate solution (Fig. 1(F)) which minimizes the cost and respects the constraints. Moreover, one or several solution(s) can be eligible and respect the real time constraints, area and consumption. It is at this moment that the exploration algorithm intervenes to choose the "adequate" solution.

4.3. Energetic, Cost and Performance Models

Nowadays, various works treat this problem for multiprocessors architectures (see Refs. [25–27]). In this study, the exploitation of a ready and validated scheduler is a possible solution and will be used. Concerning the temporal performances, we introduce the partitioning and the scheduling in order to extract the temporal model.

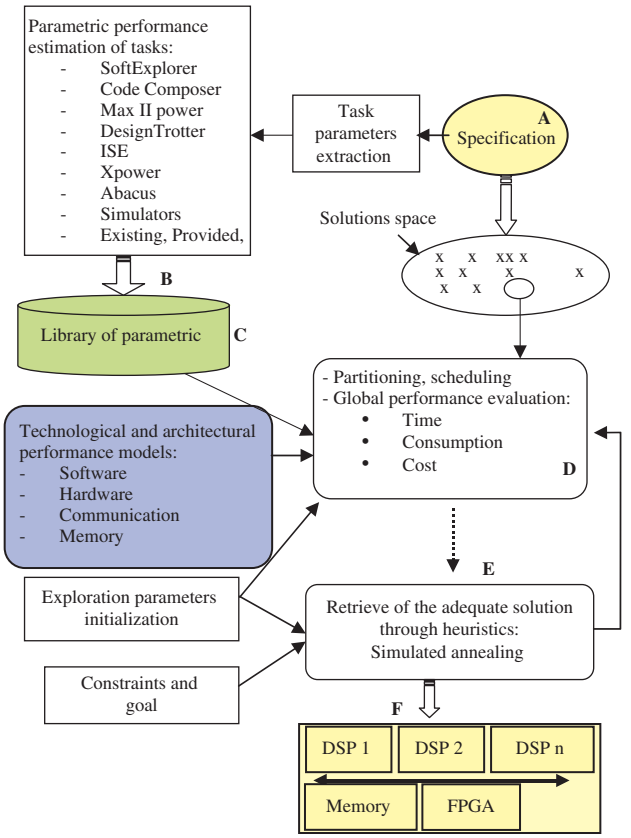

Fig. 1. Low power exploration methodology.

In fact, the partitioning and the scheduling of tasks are two recurrent problems in real time systems. Concerning consumption, we propose in Table II the parametric consumption models. The latter can take into account various parameters:

—The idle state of each unit

—The frequency

—The Vcc, frequency and the buses size,

—The data size to be transmitted,

—The static and dynamic consumption of material modules

4.3.1. The Communication Consumption Models

In this study, communication is managed via a shared bus. The modelling works of discrete bus consumption are not numerous. A buses consumption model represented in Ref. [17] can be exploited as an example (Eq. (1)).

$$P\_\text{bus} = 1/2 * C\_\text{bus} * \text{Vcc}^2 * N\_\text{bits} * M \qquad (1)$$

With N_bits: the buses size, C_bus: capacity, M: is the number of words sent per second and F: the frequency. The buses designer provides Vcc and N_bits. In this work, we have exploited the PCI buses model.

Table II. Parametric consumption models.

Target_energy	Models
DSP	$P\_\text{Idle} * (\text{Texe\_total(DSP}i) - \sum_{\text{task}(i)} \text{Texe}(i)) + \sum_{\text{Task}(i)} \text{Texe}(Ti) * P(Ti)$
FPGA	$\sum_{\text{Task\_active}(i)} P\_\text{dynamic\_Task}(i) * \text{Texe}(i) + \sum_{\text{all\_Task}} P\text{stat} * \text{Texe\_total}$
Memory	$\text{Texe\_total} * P\text{stat} + \sum_{N\_\text{accés}} P\_\text{access}(R/W) * T\_\text{access}$
Buses	$1/2 * C\_\text{bus} * V^2 * N\_\text{bits} * N\_\text{Words\_s} * (2 * N\_\text{data}/N\_\text{bit\_bus} + 2)/F$
System	$\sum_{\text{Target}(i)} \text{Energy\_Target}(i) + \text{Energy\_buses} + \text{Energy\_memory}$

The communication time is: (Eq. (2))[17]

$$\text{Tcomm} = \frac{2 * N\_\text{data}/N\_\text{bit} + 2}{F} \tag{2}$$

So the buses energy will be (Eq. (3)):

$$\text{Energie\_bus} = 1/2 * C\_\text{bus} * V^2 * N\_\text{bits} * M * \frac{2 * N\_\text{data}/N\_\text{bit} + 2}{F} \tag{3}$$

Due to the high abstraction level in which we work, it is rather difficult to have a more precise buses model.

4.3.2. Memory Consumption Model

Concerning the memory, it is useful to model and integrate it in consumption since its size is in full growth in the embedded systems (90% of the area in 2011).[14] In fact, at the time of memory access in reading or writing, this peripheral will consume energy, which is added to its static consumption (Eq. (4)).

$$\text{Energy\_memory} = \text{Texe\_total} * \text{Pstat} + \sum_{N\_\text{accés}} P\_\text{access}(R/W) * T\_\text{access} \tag{4}$$

With P_access: access consumption at the time in reading or writing. The memory designer generally provides this information.

4.3.3. FPGA Consumption Model

The FPGA power model must take account of static and dynamic consumption. Indeed, an FPGA consumption is due to the active tasks consumption as well as all the synthesized tasks consumption whether in an idle state or not (Eqs. (5), (6)):

$$P(\text{FPGA}) = \sum_{\text{Task\_active}} P\_\text{dynamic\_Task}(i) + \sum_{\text{Taches}} P\text{stat} \tag{5}$$

$$\text{Energy (FPGA)} = \sum_{\text{Task\_active}(i)} P\_\text{dynamic\_Task}(i) * \text{Texe}(i) + \sum_{\text{Task}} P\text{stat} * \text{Texe\_total} \tag{6}$$

4.3.4. The Cost Model

Due to the dominant presence of the software part (DSPs), the cost will be a secondary constraint which we cannot modify only by modifying the DSP number. In addition, due to the diversity of the technologies, the cost of each resource is balanced by a cost coefficient (Eq. (7)). (1 mm^2 of DSP area can be less expensive than that of an FPGA).

$$\text{Cost\_tot} = \sum_{\text{Ressources}} \alpha i * \text{Area}(i) \tag{7}$$

4.4. Estimation Method

The architectural solution proposed for any application has three characteristics: energy, cost and time. As these three parameters interact together, there is a necessity to make a solution exploration. For the moment, a particular attention has been given to the power as an objective. Among the key points of the space exploration, we mention the application global performance and the estimation models.

These high-level estimation models will take account of the architectural and algorithmic parameters. The study is undertaken on applications written in C language. For this reason, the estimation method based on the transposed FLPA (Functional Level Power Analyse) has been exploited in order to propose high-level parametric models. These estimation models will be presented in this section.

From the application functional analysis, the FLPA[11] methodology makes it possible to develop a parametric model which represents the target consumption behaviour. This methodology is composed of four stages: (Fig. 2)

—Functional analysis which determines the parameters influencing the power model.

—The characterization of each parameter is done to qualify its influence on the application power consumption.

—The global model is established according to the available parameters.

—The model validation by measurements.

Thus, we can take account of the algorithmic characteristics, in order to evaluate the consumption at the

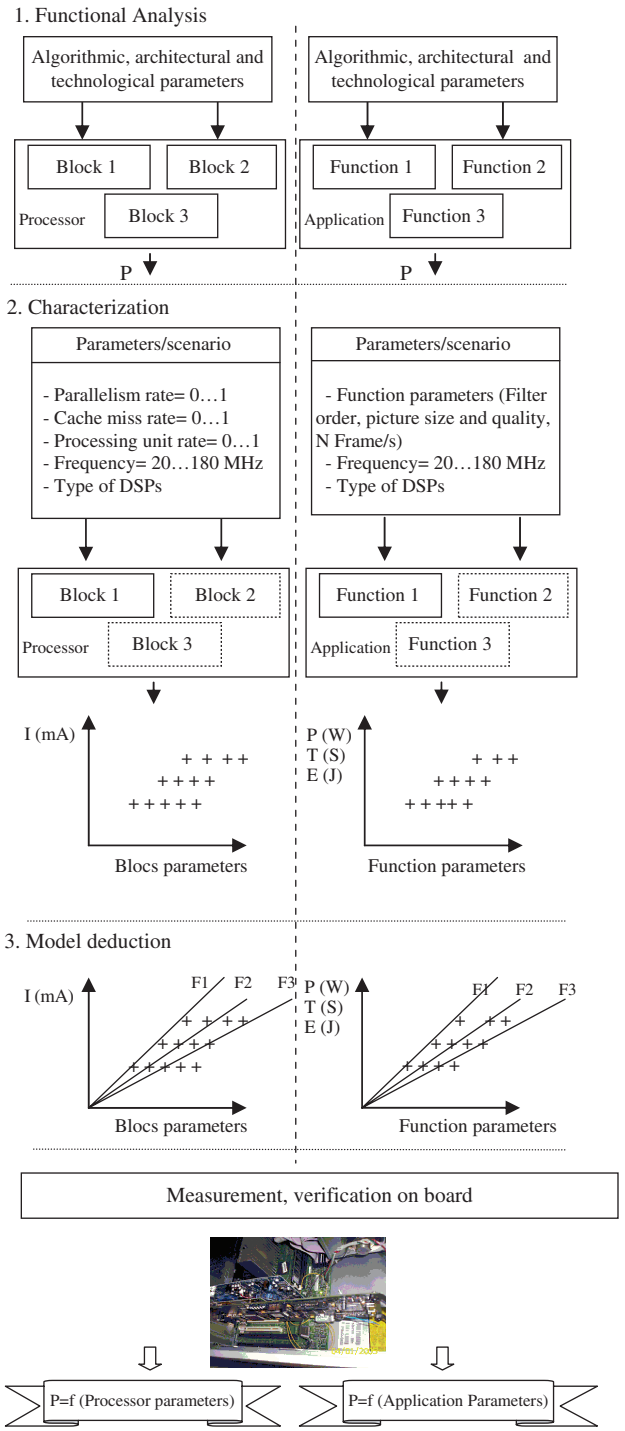

Fig. 2. (a) FLPA Methodology for processor (b) Transposed FLPA for IP(SW).

algorithmic level according to the variation parameters. In fact, this methodology starts from the extraction of the algorithmic, architectural and technological parameters, which have a direct influence on the application consumption (image size, image resolution, computing precision, DSP target, and frequency). The following stage consists in extracting the consumption variation according to each parameter through estimations or measurements thanks to scenarios.

The final stage has to do with elaborating mathematical consumption laws according to these parameters. A confrontation of established models with measurements on a DSP board is possible in order to have an idea about their precision. These models established for various signal-processing applications thanks to this methodology will be integrated in a library for exploration. With this library, we have the possibility to explore many parametric models of the various tasks. These models can take account of:

—The application parameters: image size, resolution, chrominance, the filter order…

—Architectural and technological parameters: frequency, Vcc, the target…

This library contains ready and parametric models exploitable for various applications. Thus, in case of modification of the application task specification, we do not need to re model every thing. Unfortunately, this library is not always complete and often requires updates. Indeed, a problem arises if a new task is presented in the application by addition from the designer or by modification. In this case, various solutions are possible in order to propose an estimation model of the new task performances and this is:

—Through tools like Softexplorer, Design trotter, the ISE-Xpower environment, Code composer, max II power, Quartus power play analyzer and processors simulators.

—Through the targets datasheets indicating average consumption, the consumption abacus, or using the simplistic consumption models like $P = 0.063$ Freq Area. With these techniques, we have the possibility of having performance models rather quickly with a relatively average precision.

—Through measurements on a board in case of availability.

5. EXPLORATION TOOL AND RESULTS

We present in this section the exploration environment which rests on two tools. The first one is useful for the task graph specification and performances capture. The second one is dedicated for the evaluation and the solutions space exploration in order to extract the adequate solution (Fig. 3). The necessary information for the exploration includes the various possible implementations of each task. Since a task can have several performances according to these parameters, each implementation takes account of algorithmic and architectural parameters during the data capture of the execution time, the average power, the maximum power and the resulting data size. This description is managed by a graphic interface written in java. It allows generating a text file containing this information. This textual description of the application will be the principal entry for computing under the Matlab environment.

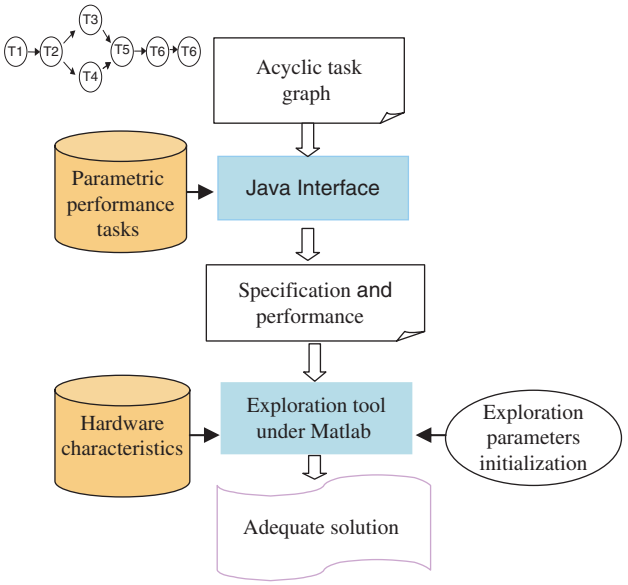

Fig. 3. Exploration environment.

5.1. Heuristics

It is to be signaled out that the choice of the best solution on a high level of abstraction is not so easy because of the possible number of architectural combination. The design space exploration is one of the necessary stages in the embedded systems conception. It permits to solve the problem of space complexity in order to reach the adequate solution quickly. In addition, the exploration complexity is related to the application complexity. Indeed, for an application containing N tasks functioning on a monoprocessor architecture, the number of possible solutions is established by the following law (Eq. (8)):[29]

$$U_n = \sum_{q=1}^{n} \sum_{i=0}^{q} \frac{(-1)^{q-i} i^n}{(q-i)!i!} \tag{8}$$

In case of a multiprocessor architecture (p processors), the problem becomes more and more complicated. The number of possible resolutions would be (Eq. (9)):

$$U_n = \sum_{q=1}^{n} p^q \sum_{i=0}^{q} \frac{(-1)^{q-i} i^n}{(q-i)!i!} \tag{9}$$

Let us take the case of a problem composed of ten tasks running on three processors. The number of possible solutions exceeds in this case 10^7 combinations.[29] The designer of such system is unable to manage theses solutions neither manually nor in a precise way quickly.

In order to extract an adequate solution among those present in the space of solutions which respect the system constraints, a meta-heuristics is necessary in order to solve this optimization of an NP hard problem.[28] In fact, with the iterative traditional improvements algorithms, the research process is reiterated until any modification makes the solution less efficient. Figure 4 shows that this iterative improvement algorithm does not lead in general to a global minimum, but only to a local minimum.

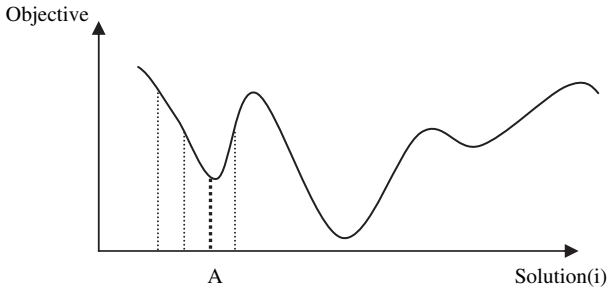

Fig. 4. Objective function of an optimization problem.

With meta-heuristic (simulated annealing, tabu method), it is a question of authorizing a temporary degradation of the solution, during the current configuration change.[30] These algorithms yield acceptable near-optimal solutions for a variety of problems. But there is no known universal algorithm that works efficiently for all problems. A control mechanism of degradations allows to avoid the heuristic process divergence. Consequently, it becomes possible to be extracted from the trap which represents a local minimum, to leave and to explore another more promising "zone." During this work, we have exploited the algorithm of "simulated annealing." The advantage of this method is its aptitude to get a good quality solution. Moreover, it is a general method; it is applicable and easy to program, for all the problems, which concern the techniques of iterative optimization. On average, the simulated annealing algorithm finds better solutions compared to Genetic Algorithm and tabu method within reasonable computational time.[31]

In addition, this heuristic offers a great flexibility, because the new constraints can be easily built-in. It is about a method where the vicinity complete exploration of the current solution is replaced by a random vicinity solution. We accept this solution if the variation D of the cost is negative. If not, we go nevertheless to the solution of a higher cost with a probability $\exp(-D/T)$ parameterized by a positive real T called temperature. We start again the new solution process after having lowered the temperature T slightly. We stop when T becomes negligible, i.e., lower than a small real positive where the corresponding acceptance probability is almost null. At this time, the probability of going up on a less suitable solution is quasi-null, and the method behaves like a local research. The simulated annealing can thus escape the local minima since it agrees to increase the cost. It gives very good results if it is led rather slowly: $T_n + 1 = f(T_n)$. Indeed, the choice of the decrease diagram is crucial in this algorithm because an extremely fast decrease can trap the solution in the vicinity of a local minimum.

Simulated annealing algorithm:

choose an initial solution (Sol_Initial[1 .. N])
choose an initial & final temperature $T0$ & Tf;
Current_solution = Sol_initial
While($T(i) < Tf$)
{ New_solution = find a near current_solution
Calculate Δ cost = Cost(NewSol)
− Cost(Current_solution)
If Δ cost ≤ 0
Current_solution = New_solution
Else
R = rand[0 .. 1];
if $R \leq \exp(-\Delta\text{cost}/T(i))$
Current_solution = New_solution
end
end
$T(i+1)$ = decreasing function ($T(i)$) //cooling function
}

To start the retrieve by a simulated annealing, the algorithm parameters must be selected in a good way. It is the case of the initial temperature, the decrease function of temperature and the criterion of stop.

- Initial temperature To: we can calculate it as preliminarily using the following algorithm:
—To make 100 solutions randomly; to evaluate the average Δ cost of the corresponding changes.
—To choose an initial rate of acceptance Ro of 50% for example in order to explore the maximum of space.
—To deduce To from the relation Ro = $\exp(-\Delta\text{cost}/\text{To})$
- Decrease of the temperature: can be carried out according to the law $T(k+1) = 0.9 * T(K)$
- Program stop: can be operated after 2 or 3 successive stages of temperature without any new acceptance. In this way, we guarantee that the selected solution is not local. (Fig. 5)
- Essential checking during the first execution of the program:
—The generator of a random real number in $[0, 1]$ must be quite uniform.
—The "quality" of the result must vary little when the program is launched several times with different initial configurations.

Simulated annealing itself is not always suitable for all combinatorial optimization problems. Empirically,[31,32] it has been observed that for simulated annealing to work satisfactorily, the cost function should not contain narrow and steep valleys. The acceptable near minimal solutions should not be within such narrow and steep valleys. The energy (cost) function should change smoothly upon changes in states. Simulated annealing usually finds a near optimal solution, but not the global minimum itself if it has a low probability of being found.[31] Finding suitable temperature schedules and best values of control factors for simulated annealing often needs careful experimentation. In spite of these problems, simulated annealing is being successfully used in a large number of practical problems, including VLSI chip design and layout, channel routing, graph drawing, image processing, coding theory, graph coloring and partitioning, satisfiability, and so on. The simulated annealing algorithm finds and reports near-optimal solutions in a shorter average time interval compared to genetic algorithms and tabu search in most of the test problems.[31,32]

5.2. Implementation

In order to implement the tool with Matlab (Fig. 6), a mono-objective exploration is done. It is allowed according to designer choices: to reach low consumption architectural solutions under real time constraints. As the designer has the possibility of imposing only the maximum number of processors in architecture without fixing

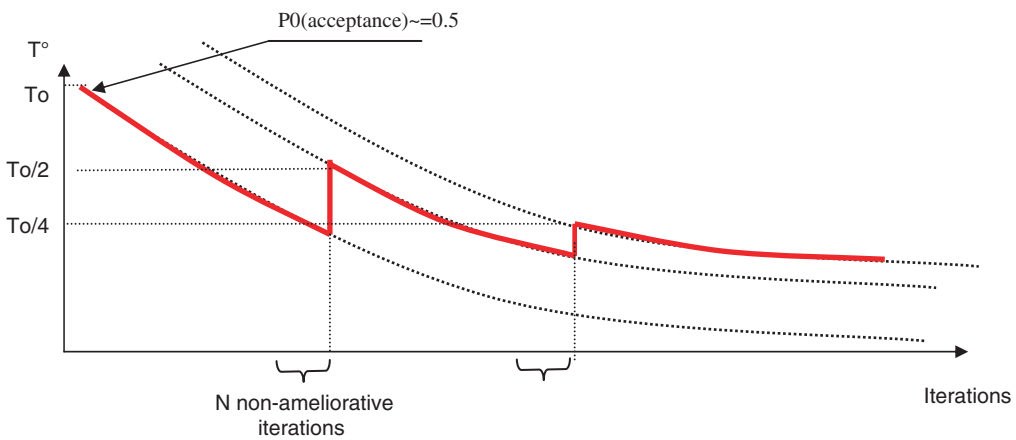

Fig. 5. Simulated annealing halt condition.

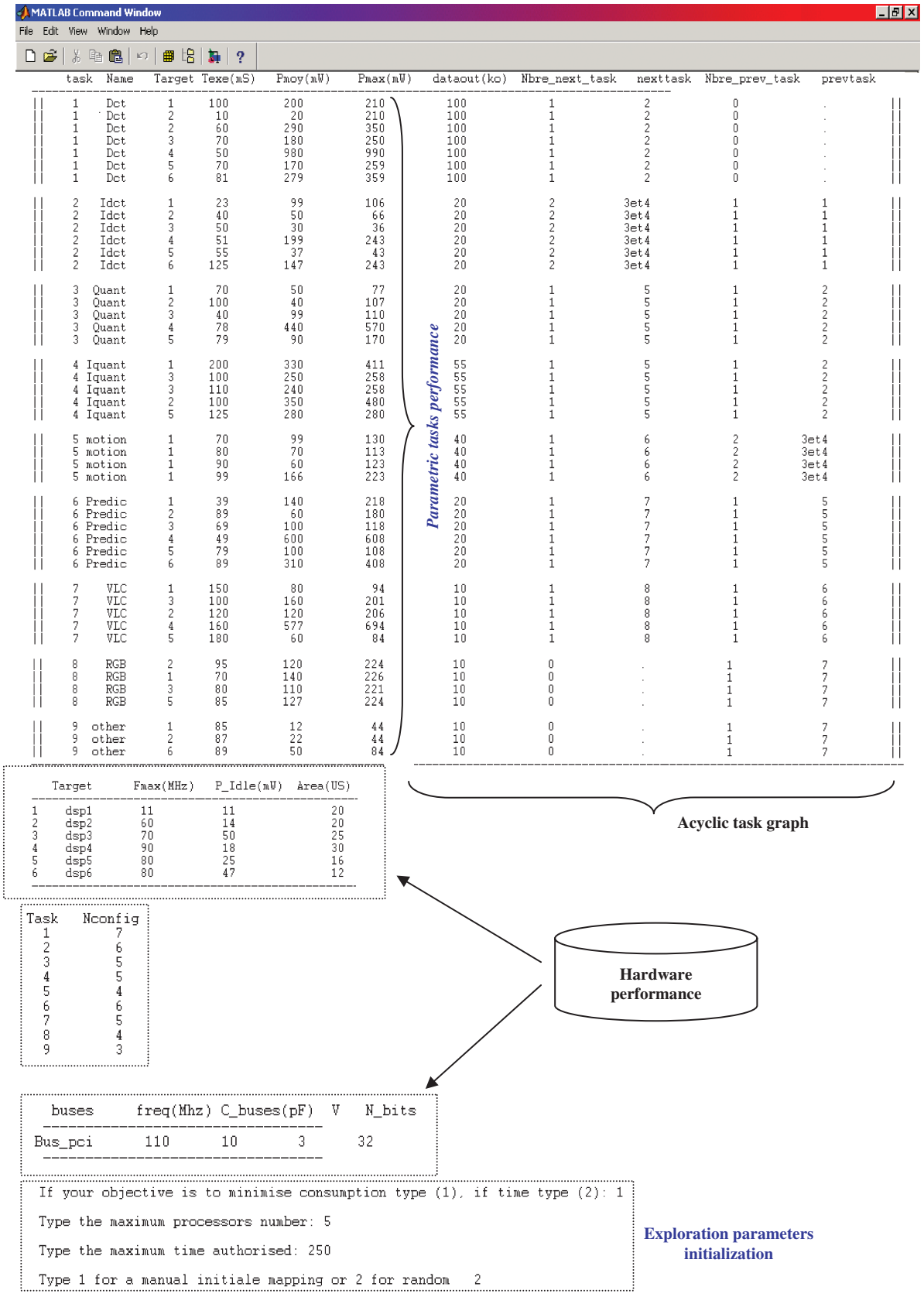

Fig. 6. Exploration with matlab.

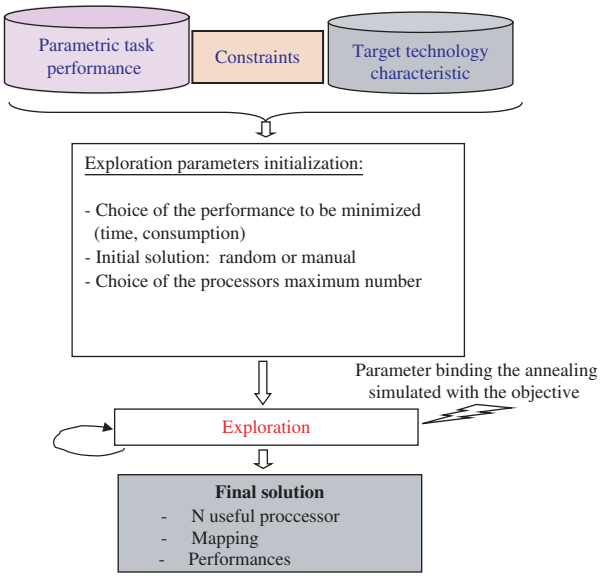

Fig. 7. Exploration methodology.

the exact number, this number will not be fixed. The algorithm will explore the most promising solutions among those which respect the real time constraints and the maximum number of processors. It is the tool, which extracts the number of useful processors as well as the adequate architectural mapping and the performances of the whole system. This parameter setting of the number of the treatment units will allow the designer on the one hand, not to limit himself/herself to a unique architecture when designing the product and to be guided by the tool when choosing the hardware solution on the other hand.

This method is based on: (Fig. 7)

—Parametric tasks performances present in the textual description file: With the diversity of the existing performance values according to the algorithmic and architectural established parameters, a model library can be analyzed by the tool. This allows the performance evaluation of each task and the adjustment of its parameters according to the objective.

—-Target technology: The characteristics of each target technology are necessary in order to be able to consider the total performance of the whole system. Among these characteristics, we can mention: the Vdd, the frequency, buses size, idle power, etc . . .

—Constraints: the designer defines the application constraints which will be provided to the tool in order to accept or refuse the solutions extracted during the exploration.

Since the target architecture is parametric with a variable number of resources and since the area, technology and cost constraints, the maximum number "Nmax" of processing units will be fixed in the beginning. Thanks to the optimization heuristics, which will explore the solutions space, the number of useful resources *N*_choice will be fixed. For that, the tool test several configurations by

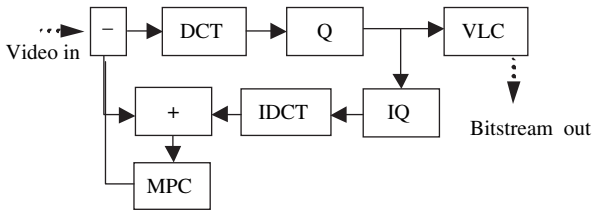

Fig. 8. MPEG2 Encoder tasks.

evaluating the total performance of each one by exploiting the information in entry. At the end of exploration, the tool provides the solution, which answers the objective as well as its temporal and energy performance. The tool manages the architectural mapping. It associates each treatment unit to the adequate tasks in order to achieve the selected goal.

5.3. MPEG-2 Results and Analysis

An initialization of the solution space exploration parameters is necessary. The user or the designer has the possibility to choose a random initial solution or a particular solution according to his/her knowledge on the application behaviour. The system constraints will be considered during each solution evaluation in order to satisfy the constraints. In this case, the designer will impose the real time constraint as well as the maximum number of resources to be exploited. In addition, the exploration algorithm makes it possible to extract the most promising solution according to the objective. Thus, we extract the necessary resources number to exploit in order to implement the application. The tool thus guides the designer when choosing the target architecture in terms of number and type of resources on a high abstraction level.

We present here some results of our methodology. We have considered the MPEG2 application. The most important tasks are: motion estimation, prediction (MPC), DCT, Inverse DCT, Quantization, Inverse Quantization and VLC (Variable Length Coder). (Fig. 8)[33]

Figure 9 shows the exploration results in a virtual space composed of 6 DSPs (2∗C5510, 2∗C6701 and 2∗C6201) communicating via a shared PCI buses with an objective to extract a low consumption solution respecting a strict real time constraint.

Up to now, hardware modules are not included in the exploration. The algorithm converges "quickly" towards solutions using only three processors (500 iterations). It is thanks to the simulated annealing heuristics that the space complexity problem is reduced. Thus, the user can know the adequate DSPs number for the application and also the architectural mapping which minimizes the whole system consumption while respecting the constraints (Table III).

In the Figure 9(A), we present the result of the solution space exploration based on the simulated annealing heuristics. The tool explores the space through this heuristics

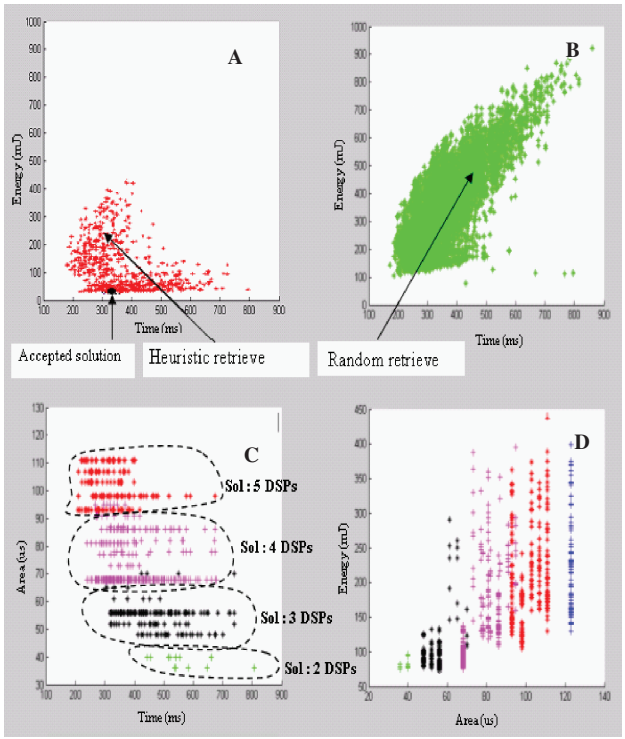

Fig. 9. Exploration results.

and converges towards the solutions whose consumptions are less than 100 mJ/GOP (Group of Pictures). Moreover, in the Figure 9(B), we present as an indication the random global solution space exploration. With this "intelligent" heuristics, a time saving in the adequate solution retrieve has been proven. Furthermore, we can conclude from the Table III that the C6201 is not adequate for the low power design because it's not selected. For more legibility, the Figure 9(D) shows the area and the consumption evolution for various architectural solutions whose number of processing elements is variable: from two to six processors.

Figures 9(C)–(D) shows also the energy evolution according to the area and/or time, thus allowing a detailed knowledge concerning the consumption variation space according to the solution number of processing units.

Thus, the designer has the possibility to extract adequate target architecture for his/her product with minimum parametric information on a high level of abstraction. In addition, the tool proposes an adequate mapping of the tasks graph in order to have a system which answers the objective and the constraints.

Table III. Bests low power solutions.

MPEG2-1 GOP (group of picture)			
Architecture	2∗C5510 & C6701	1∗C5510 & 2∗C6701	2∗C5510 & C6701
Execution time (mS)	70.61	53.36	65.67
Average power (W)	1.13	1.61	1.22
Energy (mJ)	86.57	85.91	80.12
Tasks mapping/(DSP)			
1st C5510: (A)	MPC/(C)	MPC/(C)	MPC/(C)
2nd C5510: (B)	DCT/(A)	DCT/(D)	DCT/(B)
1st C6701: (C)	IDCT/(B)	IDCT/(D)	IDCT/(B)
2nd C6701: (D)	Quant/(A)	Quant/(A)	Quant/(A)
	IQuant/(B)	IQuant/(A)	IQuant/(C)
	VLC/(B)	VLC/(D)	VLC/(A)

6. ESTIMATION ACCURACY

When we discuss measurements or the results of measuring instruments, there are several distinct concepts involved which are often confused with one another like the distinction between accuracy, uncertainty and precision. In fact, accuracy refers to the agreement between a measurement and the true or correct value. The accuracy cannot be discussed meaningfully unless the true value is known or is knowable. But precision refers to the repeatability of measurement. It does not require the knowledge of the correct or the true value.

It is sometimes possible to identify an interval so that we can assert that this interval "covers" the true value of the measure with a certain given probability P. This interval is then called a confidence interval for the estimate value. The width of the confidence interval is a measure of the uncertainty about the position of the true value of the estimated parameter.

The probability P is arbitrarily chosen by the designer. It is called the confidence level for the confidence interval, an is denoted by $(1-\alpha)$. The most frequently chosen values for α are 0.05 and 0.01, corresponding to 95% and 99% confidence levels.

In most practical research, the standard deviation (σ) for the population such as the energy consumed or power is not known. In this case, the standard deviation is replaced by the estimated standard deviation (s). Since the standard error is an estimate for the true value of the standard deviation, the distribution of the sample mean X is no longer normal with mean (μ) and standard deviation $(\sigma/n^{0.5})$. Instead, the sample mean (μ) follows the t distribution with mean and standard deviation $(s/n^{0.5})$. The (t) distribution is also described by its degrees of freedom. For a sample of size n, the (t) distribution will have $n-1$ degrees of freedom. As the sample size n increases, the (t) distribution becomes closer to the normal distribution, since the standard error approaches the true standard deviation (σ) for large n.

From the consumption and time probability density for each task running on each DSP, it is important to extract the global performance and consumption. For this reason, the consumption probability density will be calculated using the Eqs. (10) and (11).

For example, f_t is the execution time probability density of task$_i$ and f_p is the power probability density extracted

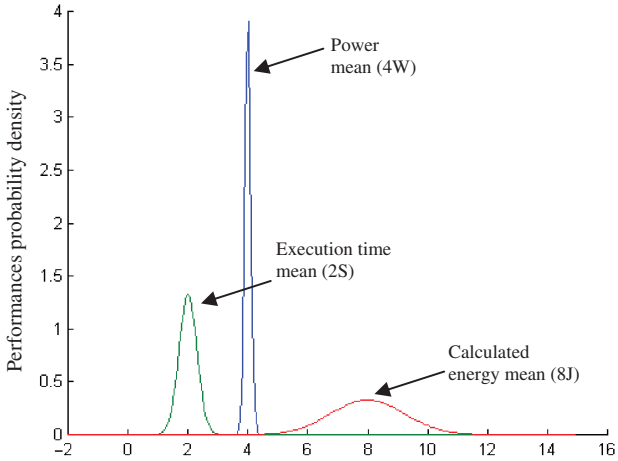

Fig. 10. Example of time, power and the calculated energy probability density.

using measures or estimation, so f_e which is the energy probability density will be equal to:

$$f_e(e) = \int_{-\infty}^{+\infty} \frac{1}{|u|} f_t(u) * f_p\left(\frac{e}{u}\right) du \tag{10}$$

$$f_e(e) = \frac{1}{2\Pi\sigma_t\sigma_p} \times \int_{-\infty}^{+\infty} \frac{1}{|u|} \exp^{1/2[((u-m_t)/\sigma_t)^2+(((e/u)-m_p)/\sigma_p)^2]} du \tag{11}$$

The Figure 10 shows as an example the time, power and the calculated energy probability density of the task$_i$ running on DSP$_j$.

We can conclude that the energy probability density is also a normal distribution with mean (μ_e) and standard deviation (σ_e). Thanks to this approach, we can estimate the mean and the standard deviation of the whole application consumption. This permits also to compute the expected accuracy of estimations. Concerning the confidence interval: for a population with unknown mean μ and unknown standard deviation, the confidence interval for a chosen confidence level $(1-\alpha)$ and for a population based

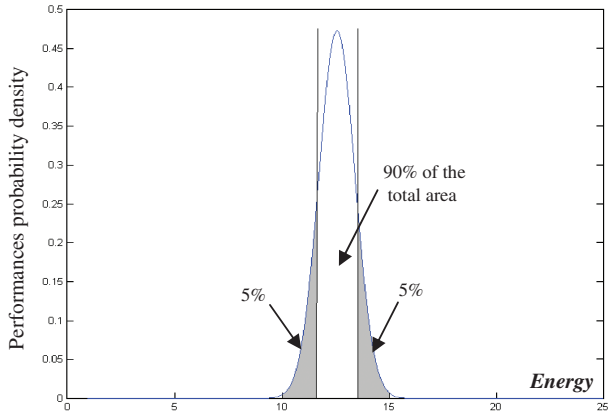

Fig. 11. Graphical accuracy and precision of the methodology.

on a n random sample, is:

$$\overline{X} - t_\alpha \frac{\widehat{\sigma}}{\sqrt{n}} \leq \mu \leq \overline{X} + t_\alpha \frac{\widehat{\sigma}}{\sqrt{n}} \tag{12}$$

Where t_α is the t "*student distribution*" with $n-1$ degrees of freedom.

We show in Table IV the consumption confidence interval of the most important MPEG2 tasks running on a chosen DSP(C5510, C6201 and C6701). These intervals are based on measures or estimation. The error estimation with SoftExplorer is ±7% with confidence level (95%). While for estimations with (six) measures on board, we compute the mean and the deviation for each task:

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{6}(\text{measures}(i)) \tag{13}$$

$$\widehat{\sigma} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{6}(\text{measures}(i)-\overline{X})^2} \tag{14}$$

Based on the consumption confidence interval of each task and through the proposed mathematical approach, we are able to compute the accuracy of the whole multiprocessor application and the confidence interval for a chosen confidence level (90%).

Table IV. Accuracy of the energetic confidence interval.

Tasks/DSPs	Estimation method	Confidence interval (Joule)	Confidence level (%)	Estimated $\overline{X}$ and σ
Motion estimation/C5510	SoftExplorer	4J ± 7%	95	$\overline{X} = 4$
		[4 − 0.28 4 + 0.28]		$\sigma = 0.27$
Prediction/C5510	SoftExplorer	2J ± 7%	95	$\overline{X} = 2$
		[2 − 0.14 2 + 0.14]		$\sigma = 0.13$
DCT/C6201	6 Measures	[2.3 − 0.12 2.3 + 0.12]	95	$\overline{X} = 2.30$
		2.3J ± 5.6%		$\sigma = 0.128$
Quantif/C67	6 Measures	[4.27 − 0.16 4.27 + 0.16]	95	$\overline{X} = 4.27$
		4.27J ± 3.8%		$\sigma = 0.164$
Application: MPEG2 Probabilistic estimation		[12.57 − 0.297 12.57 + 0.297]	90	$\overline{X} = 12.57$
		12.57J ± 2.3%		$\sigma = 0.712\hat{}0.5$

The global consumption of this application is about 12.57 J $\pm$ 2.3% with a confidence level of 90%. Figure 11 shows the normal distribution of the energy consumed by the target composed of three DSPs. We have only 90 chances in 100 that the required MPEG2 energy value is within the confidence interval, but the precision around the predicted value is so high ($\pm$2.3%).

7. CONCLUSION

In this paper, the low power design methodology for embedded systems is studied. A methodology and an environment of low consumption design space exploration are proposed. The developed environment exploits a rich performance model of time and energy that takes account of many algorithmic and architectural parameters. Moreover in creating and/or buying a library containing rich models, we can afterwards exploit it in other applications and save time while the conception of an improved or updated version of the product. This allows us to establish the characteristics and the mechanisms necessary in order to extract an architectural solution, which meets the needs. In fact, such models can be used, for example, by an operating system, which could choose the algorithm parameters to respect the constraints of consumption according to the context. We would have thus an approach of power and energy management at the algorithmic level in order to carry out an adaptive control. The key points of this problem are approached through a parametric analysis method and a heuristics based on the simulated annealing.

In the future works, it is interesting to exploit this environment to explore the solutions space of a more significant application like H264 in order to validate the approach. Moreover throughout this work, parametric models of H264 are established on various levels of granularity. In addition, in the work already made, the maximum power constraint supported by the target architecture is not considered yet during the exploration.

Moreover, we have proposed a novel methodology for dealing with accuracy and precision in high level consumption estimation. Mathematical approach provides a probabilistic estimation computing to retrieve the confidence interval. This methodology offers the designer an opportunity to model the application consumption in a high level for different target. It permits also to extract the energetic performance of the whole architecture without a need to a multiprocessor board. This will help the designer in choosing the adequate application implementation, and in giving a good idea about the constraints for further development.

References

1. Y. Cao and H. Yasuura, A system-level energy minimization approach using datapath width optimization. *Proceedings of the International Symposium on Low Power Electronics and Design ISLPED'01*, CA **(2001)**, pp. 231–236.
2. L. Benini, R. Hodgson, and P. Siegel, System-level power estimation and optimization. *International Symp. on Low Power Electronics and Design ISLPED*, United State **(1998)**, pp. 173–178.
3. D. Brooks, V. Tiwari, and M. Martonosi, Wattch: A framework for architectural-level power analysis and optimizations. *Proc. International Symp. on Computer Architecture ISCA*, United State **(2000)**, pp. 83–94.
4. W. Ye, N. Vijaykrishnan, M. Kandemir, and M. J. Irwin, The design and use of simplepower: A cycle accurate energy estimation tool. *Proc. of 37th Design Automation Conf.*, United State **(2000)**, pp. 340-345.
5. W. Baek, Y. Kim, and J. Kim, ePRO: A tool for energy and performance profiling for embedded applications. *Proc. of Intl. SoC Design Conf. (ISOCC'04)*, Seoul, Korea **(2004)**, pp. 372–375.
6. D. Q. Minh, L. Bengtsson, and P. Edefors, DSP-PP: A simulator/estimator of power consumption and performance for parallel DSP architectures. *Proc. 21st IASTED Intl. Conf. Applied Informatics*, Austria **(2003)**.
7. D. Shin, H. Shim, and Y. Joo, Energy-monitoring tool for low-power embedded programs. *IEEE Design and Test off Computers*, United States **(2002)**, p. 7
8. J. Flinn, PowerScope: A tool for profiling the energy usage of mobile applications. *Proc. of the 2nd IEEE Workshop on Mobile Computer Systems and Applications*, Louisiana, USA **(1999)**, pp. 2–10.
9. P. Pakdeepaiboonpol and S. Kittitornkun, Low energy optimization for MPEG-4 video encoder on ARM-based mobile phones. *1st International Symposium on Wireless Pervasive Computing*, Thailand **(2006)**, pp. 232–235.
10. J. Ktari and M. Abid. System level power and energy modeling for signal processing applications. *2nd IEEE International Design and Test Workshop IDT*, Egypt **(2007)**, pp. 218–221.
11. J. Laurent, N. Julien, E. Senn, and E. Martin. Functional level power analysis: An efficient approach for modeling the power consumption of complex processors. *Proceedings of the IEEE Design, Automation and Test in Europe Conference and Exhibition DATE'04* **(2004)**, pp. 666–667.
12. F. Marteil, N. Julien, E. Senn, and E. Martin, A complete methodology for memory optimization in DSP applications. *The Euromicro Conference on Digital System Design DSD*, Greece **(2004)**, pp. 98–103.
13. A. Garcia, L. Gonzales, and R. Felix, Power consumption management on FPGAs. *15th International Conference on Electronics, Communication and Computers*, Mexico, March **(2005)**, pp. 240–245.
14. D. Elleouet, Y. Savary, N. Julien, and D. Houzet, A FPGA power aware design flow. *The 16th International Workshop on Power and Timing Modeling, Optimization and Simulation Patmos'06*, France, September **(2006)**.
15. K. Lahiri and A. Raghunathan, Power analysis of system-level on-chip communication architectures. *International Conference on Hardware/Software Codesign and System Synthesis, CODES+ISSS*, Sweden, September **(2004)**, pp. 236–241.
16. M. Caldari, M. Conti, M. Coppola, P. Crippa, S. Orcioni, L. Pieralisi, and C. Turchetti, System-level power analysis methodology applied to the AMBA AHB bus. *Design, Automation and Test in Europe Conference and Exhibition DATE03*, Germany **(2003)**, pp. 32–37.
17. V. Kappagantula and N. Mahapatra, PAP: PowerAware partitioning of reconfigurable systems. *HPCA/SSRS*, California, USA **(2003)**, pp. 25–32.
18. P. R. Dick and K. Niraj, MOGAC: A multiobjective genetic algorithm for hardware-software co-synthesis of distributed embedded systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 920 **(1998)**.
19. P. Bharat, P. Dave, G. Lakshminarayana, and K. Jha, COSYN: Hardware–software co-synthesis of heterogeneous distributed embedded systems. *IEEE Transaction (VLSI) Systems* 7, 92 **(1999)**.

20. K. Ghali, O. Hammami, and I. Hermann, Multiobjective design of embedded processors on FPGA platforms. *The 24th International Conference on Distributed Computing Systems Workshops ICDCS*, Japan **(2004)**, pp. 871–875.

21. P. Guitton-Ouhamou, C. Belleudu, and M. Auguin, Energy optimization in Hw/Sw tool: Design of low power architecture system. *IEEE International Workshop on System on Ship for Real-Time Systems IWSOC*, Canada **(2003)**, pp. 38–43.

22. H. Tmar, J. P. Diguet, A. Azzedine, J.-L. Philippe, and M. Abid, RTDT: A static QoS manager, RT scheduling, HW/SW partitioning cad tool. *Microelectron. J.* 37, 1208 **(2007)**.

23. A. Azzedine, J.-P Diguet, and J. L. Pillippe, Large exploration for HW/SW partitioning of multirate and aperiodic real-time systems. *Proceedings of the Tenth International Symposium on Hardware/Software Codesign, CODES*, Colorado, USA **(2002)**, pp. 85–90.

24. J. Ktari, M. Abid, N. Julien, and J. Laurent, Power consumption and performance's library on DSPs: Case study MPEG2. *Journal of Computer Science* 3, 168 **(2007)**.

25. J. Pinot, S. Bhattacharyya, and A. Edward. A hierarchical multiprocessor scheduling system for DSP applications. *Proceedings of the IEEE Asilomar Conference on Signals, Systems, and Computers, ASILOMAR* **(1996)**, pp. 122–126.

26. T. Bandyopadhyay, B. Susnata, and B. Swapan, Multi processor scheduling algorithm for tasks with precedence relation. *TENCON Proceedings Analog and Digital Techniques in Electrical Engineering*, Thailand **(2004)**, pp. 164–167.

27. Z. Lichen, H. Jiwu, and Z. Yi, Scheduling algorithms for multiprocessor real-time systems. *International Conference on Information, Communications and Signal Processing, ICICS*, Singapore **(1997)**, pp. 1470–1474,

28. N. Chabini, A heuristic for reducing dynamic power dissipation in clocked sequential designs. *The 17th International Workshop on Power and Timing Modeling, Optimization and Simulation PATMOS*, Sweden **(2007)**, pp. 64–74.

29. A. Baghdadi, N. Zergainoh, W. O. Cesario, and A. A. Jerraya, Combining a performance estimation methodology with a hardware/software codesign flow supporting multiprocessor systems. *IEEE Transaction on Software Engineering* 28 **(2002)**.

30. M. Hasan, T. Alkhamis, and J. Ali, A comparison between simulated annealing, genetic algorithm and tabu search methods for the unconstrained quadratic Pseudo-Boolean function. *Elsevier Computer & Industrial Engineering* 323 **(2000)**.

31. P. Ahmed, R. Tavakkoli-Moghaddam, and N. Safaei, A comparison of heuristic methods for solving a cellular manufacturing model in a dynamic environment. Working Paper Series, ISSN Number 1363–6839, University of Wolverhampton Business School **(2004)**.

32. G. K. Palshikar, Simulated annealing: A heuristic optimization algorithm. *Dr. Dobb's Journal* 26 **(2001)**.

33. J. Sohn, H. Kim, J. Jeong, E. Jeong, and S. Lee, A low power multimedia SoC with fully programmable 3D graphics and MPEG4/H.264/JPEG for mobile devices. ISLPED, USA **(2007)**, pp. 238–243.

Jalel ktari

Jalel ktari *received his Diploma in Electrical Engineering and his M.S. in Electrical and Computer Engineering from the National Engineering School of Sfax, Tunisia, in 2003 and 2005, respectively. He is currently working towards his Ph.D. in the low power design in the same university. His research interests include Co-design and low power DSPs as well as design space exploration.*

Mohamed Abid

Mohamed Abid is currently Professor at Sfax University in Tunisia. He holds a Diploma in Electrical Engineering in 1986 from the University of Sfax in Tunisia and received his Ph.D. degree in Computer Engineering in 1989 at University of Toulouse in France. His current research interests include Hardware-Software System on Chip co-design, reconfigurable FPGA, real time system and embedded system. He has authored/co-authored over 100 papers in international journals and conferences. He served on the technical program committees for several international conferences. He also served as a co-organizer of several international conferences.