# HW/SW Partitioning of Embedded Applications with Variable Cost Distribution of Calculation on a Reconfigurable Architecture

**GHAFFARI Fakhreddine [1&2] ; AUGUIN Michel[1] ; BENJEMAA Maher[2] ; ABID Mohamed[2]**

[1] Laboratoire d'Informatique, Signaux et Systèmes
de Sophia-Antipolis, les Algorithmes
bat. Euclide, 2000, route des Lucioles BP 121,
06903 Sophia-Antipolis Cedex

[2] Laboratoire de recherche GMS.
Ecole Nationale d'Ingénieurs de Sfax,
BPW 3038 Sfax Tunisie.

Ghaffari@i3s.unice.fr, auguin@i3s.unice.fr, Maher.Benjemaa@enis.rnu.tn, Mohamed.Abid@enis.rnu.tn

***Abstract:*** The objective of this paper is to present a hardware-software partitioning approach for applications which possess variable execution time tasks. The methodology consists to allocate and schedule the functionalities of an application with data-dependent execution times on an architecture composed of a processor connected to a dynamically reconfigurable datapath (FPGA).

***Key-Words:*** Codesign, partitioning, genetic algorithm, variable execution time, conditioned DFG.

## 1. Introduction

The analysis of movement in sequences of images is an active axis of research due to its importance in many applications: TV - supervision, compression for telecommunications, medical diagnosis etc. However, this type of application requires often powers of calculation adapted. The problem is more worse for real time embedded systems that are submitted to consumption, area and cost constraints.

To implant a great power of calculation in an embedded system implies generally the use of a parallel heterogeneous architectures to research the best performances / size /consumption trade-off. In this case, new design methodologies considered as Hardware/Software Codesign are necessary. The approach of Codesign consists to characterize the totality of the tasks of an application and to undertake their distribution on Hardware or Software targets. The major interest of this methodology resides in the research of an application/architecture trade-off satisfying the numerous constraints of design such that the cost, performances, the area, the consumption, Time To Market, the flexibility… The efficient design of these heterogeneous systems necessitates a global approach in which hardware and software parts are conceived in parallel and in an interactive manner.

One of the key phases of the Codesign approach is the hardware/software partitioning that consists in distributing functionalities of a system specification on the units of the target architecture. Face of the complexity of hardware/software partitioning problem, several approaches have been developed in literature.

The approach used in the University of California/Berkeley in the Ptolemy environment considers a manual partitioning. This partitioning is guided by the surface constraints, speed and flexibility [1]. In [2] the partitioning approach begins with an initial partition where all operations, excepting those unlimited period, are allocated to the hardware parts. The partitioning is refined by the migration of operations from the hardware to the software to obtain a partition with a lesser cost. Another approach of partitioning is adopted in the University of Linkoping [3], it is formulated as a graph partitioning problem. It realizes the transposition of a graph of control and a data flow graph on a unique partitioning graph. Different types of arcs are used between the nodes to reflect dependences. We quote also the methodology of Braunschweig University in the COSYMA system [4] where the tool of hardware/software partitioning is automatic and based on a simulated annealing algorithm. Finally, a recent approaches use a theory of clustering for instance [5] to privilege the roundup on a same unit the functions or tasks which communicate. This technique allows the minimization of the total execution time by canceling times of communication between tasks regrouped in a same cluster.

Nevertheless, all these approaches consider that functions of the application have constant calculation times, with a fixed number of resources. Well, but many applications, especially in images processing, show variable calculation costs according to the incoming images.

The detection of movement on a fixed bottom of image is used in embedded intelligent cameras. A such application is based on sequences of images processing operations. Among these operations, we distinguish those that preserve a fixed execution time from image to another, from whose times of calculation vary according to the nature of images.

It is on this last type of operations that we focus our study and on which it's a matter of to take into account the variable distribution of the cost of calculations during the hardware/software partitioning of tasks.

This paper is organized as follows. We present the target architecture in the paragraph two. We introduce the deduced model of applications in the paragraph three. Our partitioning approach is detailed in the paragraph four. Results and analyses are presented in the paragraph five before concluding.

## 2. Target Architecture

An important factor in the evolution of modern electronic systems is advances in new architecture based on programmable components. The power of calculation produced by FPGAs circuits comes from their specialization in comparison to executed program needs (treat-to-measure). The fact that it is reprogrammable allows correcting, modifying, and adapting totality of the functionalities after their implementation on the circuit. In the project EPICURE[1] the considered architecture is constituted of a processor connected to a dynamically reconfigurable circuit through a generic interface as depicted in figure 1. This type of architecture is well adapted to conceive embedded systems like intelligent cameras. The flexibility of the reconfigurable allows adapting of treatment to the environment in which is placed the camera. The dynamically reconfiguration authorizes a best exploitation of hardware resources and therefore to decrease the silicon area. To experiment our works we consider the EXCALIBUR system from Altera which integrates in a same circuit a processor and a reconfigurable unit [6].

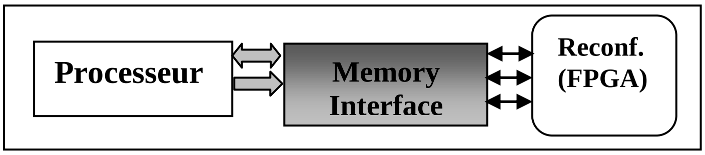

*Figure 1 :The target architecture*

## 3. Retained Model

Images processing applications can be easily modeled by a Data Flow Graphs (DFGs), at least in medium and low levels of treatment. The initial description of the system is an oriented graph without cycle. The nodes of the graph represent the treatments (calculations) and arcs of the graph represent data dependences. For instance, we can model the movement detection application of on a fixed bottom of image by the Data Flow Graphs as shown in figure 2.

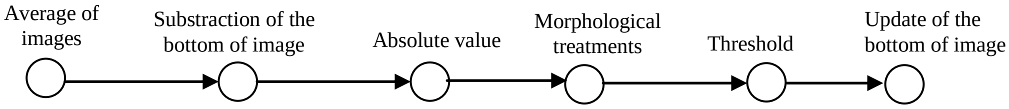
Average of images
Substraction of the bottom of image
Absolute value
Morphological treatments
Threshold
Update of the bottom of image

*Figure 2 : An example of Data Flow Graph*

## 4. Partitioning Approach

Many operations in image processing have an execution time strongly correlated to the content of the image. We can therefore, consider two types of tasks: tasks that keep a constant execution time for the different parcels of data and those that have a variable execution time. To develop a partitioning method of applications containing tasks of this last type, it is necessary first of all to define the nature of the correlation between the time execution and the characteristics of the treated data. The correlation parameter is identified by a preliminary study of the concerned task, followed by a confirmation with practice tests [7]. For instance, in the application of detection of movement on a fixed bottom of image, we can define correlation parameters for tasks that possess variable execution times as indicated on the table following:

| Task | Correlation parameter |
|---|---|
| Average of images | Number of images |
| Reconstruction | Total Size of objects |
| Labeling | Total Size of objects |
| Covering Envelope | Number of objects |
| Centre of gravity | Number and Size of objects |
| Moving Test | Number of objects |

[1] This project is supported by the French Ministry of Research and Education through the *Réseau National des Technologies Logicielles*. The partners of the project are CEA, Thales, Esterel Technologies, LESTER – Université de Bretagne Sud and I3S – Université de Nice Sophia Antipolis/CNRS.

By undertaking several measures of execution time on the processor and on FPGA, we can characterize these tasks according to correlation parameters. As an example, the task of Moving test [8], possesses a time of execution that depends on the number of objects in movement in the image. If we designate by n the number of objects in the image and if we measure the time of execution of the task on this image, we obtain a points as indicated on the figure 3.

From these totalities of points, we can classify images in different categories (for example A and B). For each category, we attribute an execution time that is the same for all its elements. This execution time is the maximum of times of the totality of points that belong to the same category. This has for purpose to obtain executions that satisfy constraints of time without forgetting to reduce the pessimism linked to the choice of a one maximum by function. The constraint on this classification is that thresholds defined on a correlation parameter are identical for the processor and the FPGA.

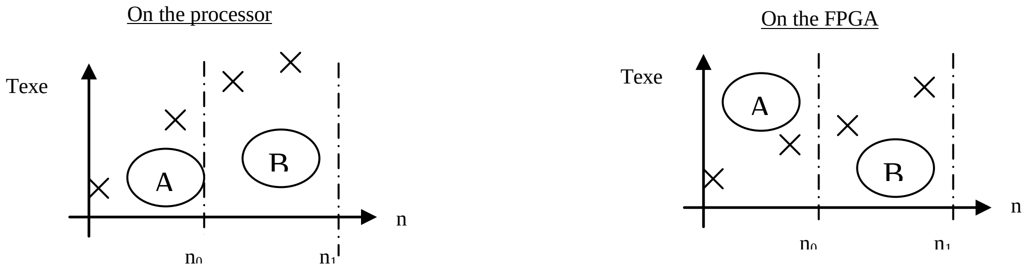

*Figure 3 : Execution time as a function of the number of objects in image*

In order that our approach does not drive to a combinative explosion of possible configurations, we have considered a level of granularity enough high and we sought a compromises between the precision of executions time values chosen for thresholds (n0 and n1 on the figure 3) and the number of configurations for partitioning [9]. We can then construct a conditioned data flow graph in which each task, with variable execution time, is duplicated so much time that there are categories identified for this task (figure 4).

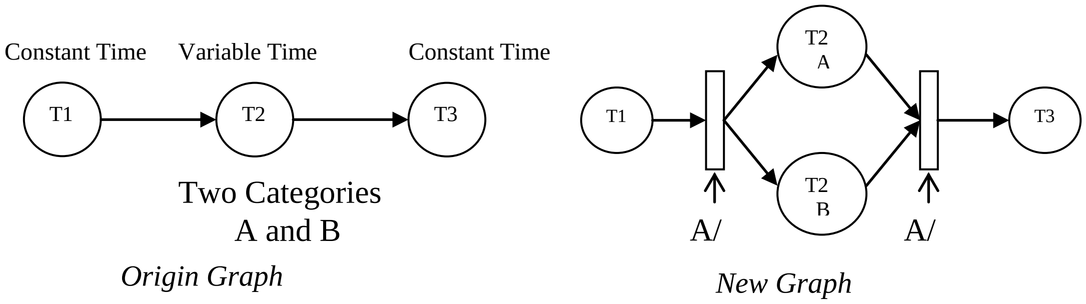

*Figure 4 : Construction of conditioned graph*

From this conditioned graph we can define the totality of possible configurations that are so much not conditioned DFGs [10].

On each graph thus obtained, we apply an algorithm of partitioning to obtain so many contexts that are memorized in the architecture to program the processor and the FPGA according to images treated.

In our approach, we use a genetic algorithm associated to an approach of clustering [11] to undertake the partitioning. Once the different contexts obtained from identified configurations, it is possible to construct the control flow of the application on the architecture. This flow of control has to activate treatment that corresponds to contexts defined by the partitioning.

At the moment of the execution of the application, we use therefore as controls value correlation parameters previously quoted to be able to identify the graph of tasks to consider and thereafter the context that it is necessary to apply on the FPGA and to the processor. This criterion of control can be the parameter of correlation himself as it can be a combination of several parameters. In permanent regime we can calculate the criterion of control for the image (i) from the image (i - 1) already treated or by a balancing on parameters of images that it precede.

## 5. Experimental Results

Results of our approach show the efficiency of the adaptation of partitioning to needs of treatment.

The dynamics reconfiguration of the FPGA allows the architecture to accept several contexts of reconfiguration as results of partitioning of all possible configurations. To each configuration, adapted to a category of images (a power of calculation), corresponds a totality of contexts allowing reconfiguration of the architecture. By comparing our approach with classic approaches working with partitioning of the worst case execution time, we notice that we have been able to gain on two levels:

- **Total execution time:** results show differences between the partitioning of all configurations. When images do not correspond to a worst case (greatest values of parameters of correlation) fewer resources are necessary to obtain times of execution that verify the debit of images. In these cases, free resources can be used for others types of treatment as for instance a function that improves the quality of the image. This is rendered possible because in our approach it is easily to quantify these free resources by configuration.

- **Time of communication:** results of the partitioning of configurations do not produce the same total communication time. To decrease the time of communication between units of an architecture, provokes a diminution of the consumption of the system because operations of memory access and transfer of data through the bus are « greedy» in term of consumption.

## 6. Conclusion

The approach of partitioning presented above is based on a genetic algorithm with a heuristic clustering. It allows to construct an architecture and an adapted real execution time diagram of the application by taking account characteristics of data evaluated at the moment of the execution.

To optimize favors results of the partitioning, it is possible to exploit results of the partitioning of a configuration to partition the configuration that possesses the greatest probability to execute then. Indeed, in the case of a permanent camera, the evolution of the number of mobile objects in the scene is in general relatively slow. It follows that an order of partitioning of configurations can be defined for the purpose to exploit at best for a given configuration the anterior state of the reconfigurable (FPGA). This allows reducing times of reconfiguration between relative treatments of images that belong to different categories.

## References

[1] A. KALAVADE, E.A. LEE, "A Hardware-Software Codesign Methodology for DSP Applications", IEEE Design & Test of Computers, Vol. 10, N° 3, pp. 16-28, September 1993.

[2] R.GUPTA, G.DE MICHELI, "Hardware-Software Cosynthesis for Digital Systems", IEEE Journal Design and Test of Computers, pp 29-41, September, 1993.

[3] Z. PENG, and K. KUCHCINKI, "An Algorithm for Partitioning of Application Specific Systems", Proc. European Design & Test Conference (EDAC-ETC-EuroASIC), IEEE CS Press, February 1993.

[4] R. ERNST, J. HENKEL, T. BENNER, "Hardware-Software Cosynthesis for Microcontrollers", IEEE Journal Design and Test of Computers, Vol. 10, N° 4, pp. 64-75, December 1993.

[5] B. DAVE, G. LAKSHMINARAYANA, N. LHA, "COSYN: hardware/software co-synthesis of embedded systems", Design Automation Conference, Anaheim, 1997.

[6] Nios Embedded Processor Getting Started Version 1.1 user Guide, ALTERA March 2001.

[7] F.GHAFFARI, M.AUGUIN, M.BEN JEMAA, "Etude du partitionnement logiciel/matériel d'applications à distribution variable de charge de calcul". Renpar'14 /ASF/SYMPA, Hammamet, TUNISIE, pp. 334–338, 10 – 13 Avril 2002.

[8] E.Duchesene training report in CEA LIST: "Détection de mouvement sur un fond d'image fixe" , 2001.

[9] F.GHAFFARI, Etude du Partitionnement Logiciel/Matériel d'applications à distribution variable de charge de calcul, training report of DEA worked in I3S université de Nice Sophia-Antipolis, 2001/2002.

[10] M.Auguin, L.Bianco, L.Capella, E.Gresset. "Conception de systèmes embarqués par partitionnement de spécifications flots de données conditionnel", Conférence Architectures Nouvelles de Machines, Sympa'6, Besançon, 19-21 juin,2000.

[11]K.Ben Chehida, Partitionnement Logiciel/Matériel pour des architectures reconfigurables utilisant une approche génétique, Training report of DEA worked in I3S université de Nice Sophia-Antipolis, 2000/2001.