

## Algorithms for the Partitioning of Applications containing variable duration tasks on reconfigurable architectures

**Fakhreddine GHAFARI**

I3S, University of Nice Sophia  
Antipolis, CNRS  
Les Algorithmes/ Euclide B,2000  
route des Lucioles BP 121, 06903  
Sophia-Antipolis Cedex  
ghaffari@i3s.unice.fr

**Maher BENJEMAA**

Research Unit GMS.  
Ecole Nationale d'Ingénieurs de  
Sfax BPW 3038, Tunisie  
Maher.Benjemmaa@enis.rnu.tn

**Michel AUGUIN**

I3S, University of Nice Sophia  
Antipolis, CNRS  
Les Algorithmes/ Euclide B,2000  
route des Lucioles BP 121, 06903  
Sophia-Antipolis Cedex  
auguin@i3s.unice.fr

### Abstract

*Many applications, especially in image processing, have variable execution times according to the nature of data to process. The implementation of low level processings in image processing (for example the detection of contours, labeling) in embedded architectures often uses specialized systems. We consider here an architecture composed of a RISC processor linked to a dynamically reconfigurable circuit through a memory interface. To help to distribute processings with variable execution time on this architecture we present a partitioning approach based on a genetic algorithm.*

### Keywords

Partitioning, reconfigurable architecture, variable execution time, motion detection, genetic algorithm, conditioned DFG.

### 1. INTRODUCTION

In the embedded system domain of control, communication and processing of the signal, applications are often submitted to real time constraints that impose adapted powers of calculation. To implant a great power of calculation in an embedded system implies generally the use of parallel heterogeneous architectures to research the best performances / size /consumption trade-off. Furthermore, the fact that the models used in algorithms of such applications are more and more precise, leads an increasing of the processing complexity. Paradoxically, advanced technology drive them equally to increase the complexity of the design work because there exists increasingly possibilities to allocate application processing on software or hardware targets.

Several studies have targeted reconfigurable architectures formed by systems of FPGAs (Field Programmable Gate Array) by processing the problem of the application tasks placements on the reconfigurable units. An algorithm using the method of simulated annealing has been used in [1] and [2]. An approach based on a genetic algorithm [3] help to the spatial partitioning of operations on cards FPGAs networks. Others works have been undertaken to help the designer in the application functionalities partitioning on heterogeneous software/hardware architecture. We can

quote for example the approach developed in [4] that presents a method of allocation/scheduling that consists in minimizes the execution time and the silicon surface. The COSYMA system [5] uses a software oriented approach, this meaning that leaving from a entirely software specification, it seeks to migrate functions to the Hardware to satisfy real time constraints. On the contrary the VULCAN system [6] uses another approach that, an initial entirely hardware specification, consists to allocate non-critical parts to a software realization to decrease the surface cost. A method based on a genetic approach is considered in [7]. A formulation of the problem following a linear program in whole numbers has equally been used in [8] and [9]. Nevertheless, all these approaches consider that functions of the application possess constant calculation times, for a fixed number of resources. Well, but many applications, especially in images processing show variable calculation costs in function of images to process.

Tasks with vague duration of different nature have already been studied, we quoted the model with multi representations [10] [11], that associates to a task two implementations, one corresponding to an optimal service quality, but of uncertain duration, and the other, known duration, but correspondent to a lesser quality service and the incremental [12] model where a task is decomposed in two parts, an essential part that has absolutely to be executed, and a secondary part, allowing to refine results produced, that will be executed if the available time is sufficient. The approach that we adopt is totally different, since we consider variations of duration linked to the nature of data to process such as the case of image processing applications that shows variable calculation costs in function of images to process.

The motion detection on a fixed image background 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 tasks execution times during the hardware/software

partitioning.

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 advanced in new architecture based on programmable components. The power of calculation produced by FPGAs circuits comes from their specialization towards the 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<sup>1</sup> 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 processing 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 [13].

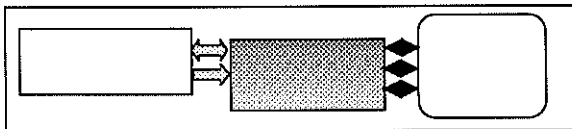


Figure 1 : The target architecture

## 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 processing and arcs of the graph represent data dependences. For instance, we can model the application: motion detection on a fixed image background by the Data Flow Graphs as shown in figure 2.

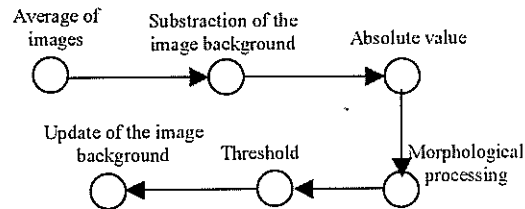


Figure 2 : An example of Data Flow Graph

## 4. Partitioning methodology

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 dynamically variable execution times. To develop a partitioning method (Figure 3) of applications with 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 processed data.

The correlation parameter is identified by a preliminary

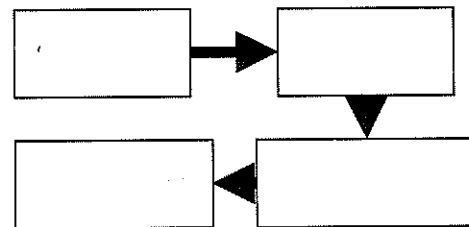


Figure 3 : Partitioning approach graph

study of the concerned task, followed by a confirmation with practice tests [14]. For instance, in the application of motion detection on a fixed image background, we can define correlation parameters for tasks that undergo with variable execution times as indicated on the following table:

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

By undertaking several measures of execution time on the processor and on the FPGA, we can characterize these tasks according to correlation parameters. As an example, the task of Moving test [15], has an execution time that depends on the number of objects in motion in the image. If

<sup>1</sup> 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.

there are  $n$  objects in the image and if we measure the execution time of the task on this image, we obtain a set of points as indicated in figure 4.

From this set of points, we can classify images into different categories or subsets (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 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 single maximum execution time per function. The constraint on this classification is that thresholds defined on a correlation parameter are identical for the processor and the FPGA.

In order that our approach does not drive to a combinatorial

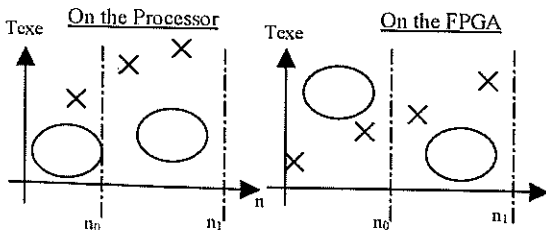


Figure 4 : Execution time as a function of the number of objects in image

explosion of possible configurations, we have considered a level of granularity corresponding to a compromise between the precision of executions time values chosen for thresholds ( $n_0$  and  $n_1$  on the figure 4) and the number of configurations for partitioning [16]. We can then construct a conditioned data flow graph in which each task, with variable execution time, is duplicated as many times that there are categories identified for this task (figure 5).

From this conditioned graph we can define the set of possi-

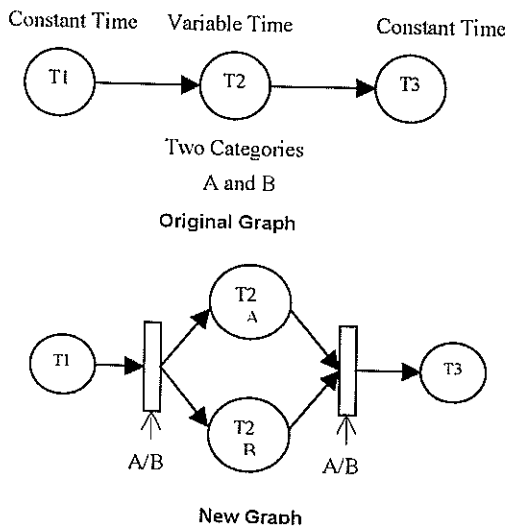


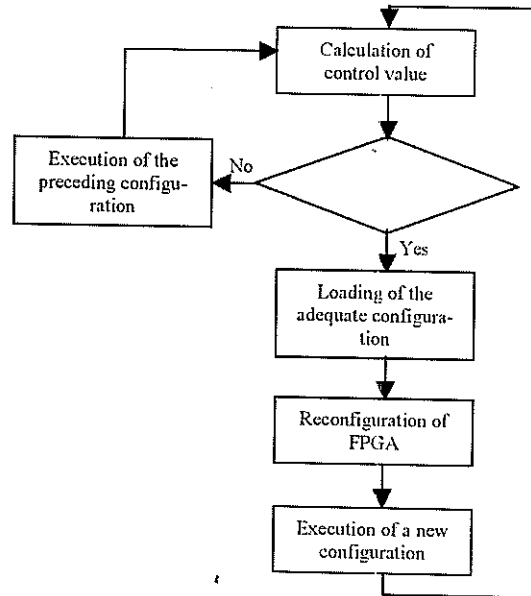
Figure 5 : Construction of conditioned graph

ble configurations that correspond to unconditioned DFGs [17].

On each unconditioned DFG, we apply a partitioning algorithm to obtain the contexts that are memorized in the architecture to program the processor and the FPGA according to images processed.

In our approach, we use a genetic algorithm associated with a clustering approach [18] to undertake the partitioning. Once the different contexts are obtained from the identified configurations, it is possible to construct the control flow of the application on the architecture. This flow of control has to activate sequences processing that corresponds to contexts defined by the partitioning.

During the execution of the application, we use therefore as control values the correlation parameters previously quoted to be able to identify the graph of tasks to consider and therefore the context that is required to apply on the FPGA and on the processor. A control value can be the parameter of correlation itself or it can be a combination of several parameters. In normal processing we can calculate the control value for the image ( $i$ ) from image ( $i - 1$ ) already processed or by a balancing on parameters of previous images as shown in figure 5.



## 5. Experimental Results

Figure 7 shows an example of function which have an execution time that depend with the number of objects in image.

The dynamic reconfiguration of the FPGA allows the architecture to accept several contexts of reconfiguration as results of partitioning of all possible configurations. To

each configuration, corresponds a set of contexts allowing reconfiguration of the architecture. Comparing our approach with classic approaches working with partitioning on the worst case execution times, we notice that we have been able to gain on two levels:

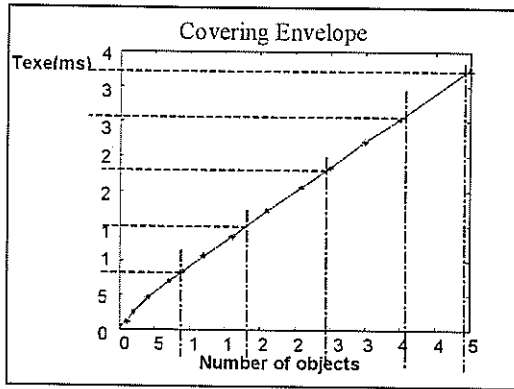


Figure 7 : Example of function with variable execution time

- **Total execution time:** results show differences between the partitioning of all configurations (figure 8). When images do not correspond to a worst case (greatest values of parameters of correlation) fewer resources are necessary to obtain execution times that verify the throughput of images. In these cases, free resources can be used for others types of processing as for instance a function that improves the quality of the image. This is possible since in our approach it is easy to quantify these free resources by configuration.

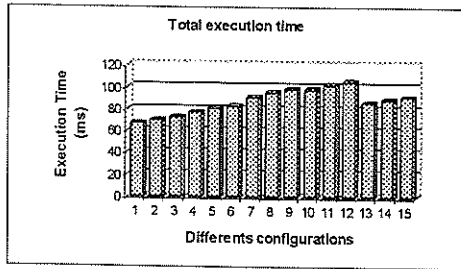


Figure 8 : Total execution time of all configuration

- **Communication Time:** results of the partitioning of configurations do not produce the same total communication time (figure 9). Reducing the communication time between units in an architecture, induces a diminution of the consumption of the system because memory access operations and data transfers through the bus are « greedy» in term of consumption.

These results show the efficiency of the partitioning approach to optimize the the utilization of resources according to the processing needs.

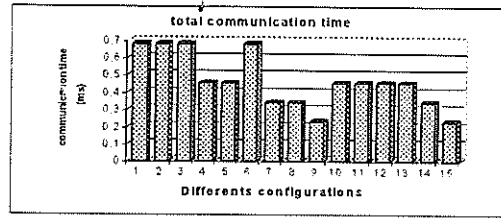


Figure 9 : Total communication time of all configuration

## 6. Future Work

Our future work consists to exploit the result of the previous partitioning in the next partitioning configuration.

Our goals is to minimize the reconfiguration time with saving the hardware tasks allocation. Figure 9 shows the conditioned DFG of " the motion detection on a fixed image background" application including all the possibilities of configurations. The two cases as depicted in the figure (dashed line, straight line) are very close and the partitioning result of the first one will help the partitioning of the second.

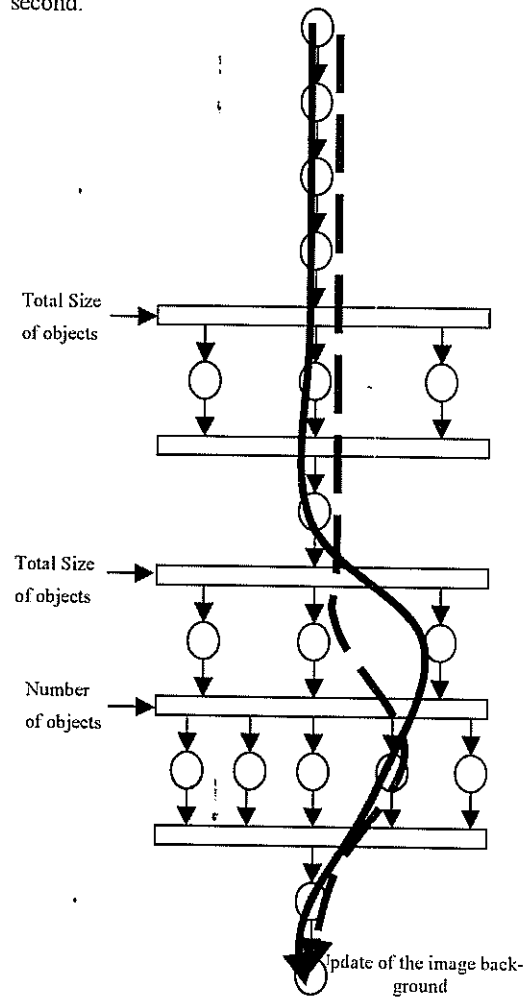


Figure 9 : DFG with variable execution time tasks

## 7. Conclusion

The approach of partitioning presented above is based on a genetic algorithm with a clustering heuristic. It allows to construct an architecture and an adapted real execution time scheduler of the application by taking account of characteristics of data evaluated during the execution. To optimize furthermore partitioning results, it is possible to exploit the results of the partitioning of a configuration to partition the configuration that possesses the greatest probability to execute next. Indeed, in the case of a fixed 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 previous state of the reconfigurable (FPGA). This allows reducing times of reconfiguration between relative images processing that belongs to different categories.

## References

- [1] K.Bazargan and M.Sarrafzadeh. Fast Online Placement for Reconfigurable Computing Systems. In Proceedings of IEEE Symposium on FPGAs for Custom Computing Machines, 1999.
- [2] T.J. Callahan, P.Chong, A.DeHon, and J.Wawrzynek. Fast Module Mapping and Placement for Datapaths in FPGAs. In International ACM/SIGDA Symposium on Field Programmable Gate Arrays, February 1998.
- [3] I.Ouaiss, S. Govindarajan, V. Srinivasan, M. Kaul, and R. Vemuri. An Integrated Partitioning and Synthesis System for Dynamically Reconfigurable Multi-FPGA Architectures. Fifth Reconfigurable Architectures Workshop, 1998.
- [4] 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.
- [5] 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.
- [6] R.GUPTA, G.DE MICHELI, "Hardware-Software Cosynthesis for Digital Systems", IEEE Journal Design and Test of Computers, pp 29-41, September, 1993.
- [7] R.P.Dick and N.K.Jha. MOGAC: A Multiobjective Genetic Algorithm for the Co-Synthesis of Hardware-Software Embedded Systems, IEEE Trans. Computer-Aided Design Integrated Circuits & Syst, vol.17, no.10, pp.920-935, Oct. 1998.
- [8] R.Niemann, P.Marwedel. Hardware/Software partitioning using integer programming. European Design and Test conference 1996.
- [9] M.Kaul, R.Vemuri, S.Govindarajan, I.Ouaiss. An automated temporel partitioning and loop fission approach for FPGA based reconfigurable synthesis of DSP applications Design Automation Conference, New Orleans, LA, 1999.
- [10] R.H. CAMPBELL - K.H. HURTON, G.G. BELFORD, "Simulations of fault-tolerant real-time deadline mechanisms", Proceedings of FCTS'9, pp. 95-101, January 1979
- [11] H. CHETTO, M. CHETTO, "An adaptive scheduling algorithm for fault-tolerant real-time systems" , Software Engineering Journal, pp. 179-186, Mai 1991.
- [12] J.Y. CHUNG, J.W.S LIU, "Algorithms for scheduling periodic jobs to minimize average error", 9th IEEE RTSS, pp. 142-152, December 1988.
- [13] Nios Embedded Processor Getting Started Version 1.1 user Guide, ALTERA March 2001.
- [14] 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.
- [15] E.Duchesene training report in CEA LIST: "Détection de mouvement sur un fond d'image fixe" , 2001.
- [16] F.GHAFFARI, "Etude du Partitionnement Logiciel/Matériel d'applications à distribution variable de charge de calcul", training report of DEA worked in I3S Informatique Signal et Systèmes de Sophia-Antipolis with ENIS Ecole Nationale d'Ingénieurs de Sfax, June 2002.
- [17] 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, pp. 139-148, 19-21 juin, 2000.
- [18] 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, July 2001.