FPGA Area Time Power Estimation for DSP Applications

S. Bilavarn, G.Gogniat, J. L. Philippe
Université de Bretagne Sud
Laboratoire d'Électronique des Systèmes Temps Réels (LESTER)
Rue Saint Maude - 56325 Lorient - FRANCE
e-mail : *bilavarn@iuplo.univ-ubs.fr*

1 INTRODUCTION

Fast prototyping of complex applications such as digital signal processing systems over Field Programmable Gate Array technology has become a very important issue due to the increasing number of FPGA components and their recent architecture evolutions. A rapid estimation of FPGA utilisation rate, time execution and power consumption (ATP in the following) at the behavioral level is necessary to allow an efficient exploration of a large design space and to select the best implementation component. The method described here aims at guiding the design process of heterogeneous architectures such as in the codesign issue, so the main quality of the performance estimator is to be fast and accurate enough in order to explore a wide design space and guaranty the choice of the best implementation solution. A lot of estimation techniques are described in the literature and many of them come from the High Level Synthesis (HLS) research. We can find methods estimating the processing unit [1][3][4], from the behavoral until the layout level, control or memory units [2] for one or two sort of constraints, area and time most of the time. But few techniques targets FPGA and to our knowledge, there is no global estimation methods (i.e. estimating the ATP cost of the whole system) starting from a behavioral specification. In [5], the method is set up on the estimation of area and timing values starting from a Register Transfert Level netlist resulting from a HLS tool. This approach aims at speed up the HLS flow by getting rid of the Partitionning, Placement, Routing and Timing optimization steps, and estimating the physical design characteristics instead. The method is very accurate but has two main drawbacks caused by the use of a HLS tool : it might be too slow in the case of the exploration of a wide design space such as in codesign, and the estimation process is strongly dependant of the technology used. Our estimation process allows both to be fastest (to the detriment of accuracy) and far less dependant of the technology through the characterization of the target FPGA in a component library. Furthermore, FPGA power consumption estimation is included in order to give a complete characterization of the design.
The outline of this paper is as follow : In section 2, an overview of the estimation flow is presented. Then section 3 explains the specification model and resources allocation algorithm whereas section 4 describes the technology caracterization file and ATP mapping process. Finally, preliminary results and a conclusion are given.

2 Estimation Flow

In the codesign issue, the system can be can be seen as a set of several functionalities with different characteristics. The design space exploration step must garanty to rapidly test the performances of those functions for several target technologies (ASICs, FPGAs, DSPs, ...) in order to find the best implementa-

tion solution. This paper deals with the problem of FPGA estimation problem. In the method described in [5], there is a strong dependance to the technology trough the use of the Synthesis tool. The estimation for another FPGA technology will need the use of another synthesis tool. That's the reason why

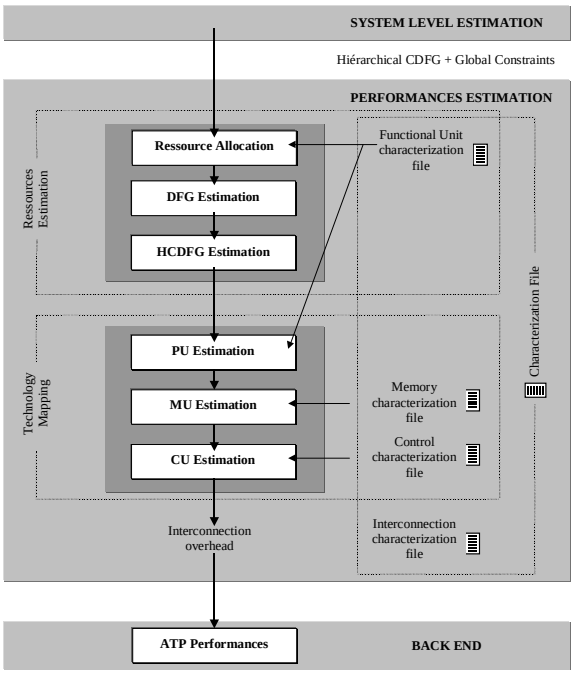

Figure 1: The Estimation Flow

we use a component library in wich each target FPGA is described in a characterization file, so we can evaluate the system's performances over several components without the need of any sysnthesis tool. Furthermore, the performances are characterized in a dynamic way to obtain a significant idea of the system's feasibility : area (FPGA utilization rate) and power consumption vs. variable time constraint (see section 5). This allow to take into account the variable parallelism within the design and to give the designer more realistic information about the system's feasibility. The estimation flow (figure 1) can be divided into two main parts. First, the resource allocation algorithm estimates the resource requirements starting from the behavioral specification. At this step of th e flow, time is given in number of clock cycles. Then, the ATP characteristics of those resources are projected on the target FPGA through the cahracterization file in order to obtain physical values of the performances.

3 Resources Estimation

The behavioral description of the application is a Hierarchical Control Data Flow Graph (SPF Model [6]) resulting from a high level language specification (given in the C language). The HCDFG model (figure 2 & 3) is composed of nodes and edges giving the direction of the control flow between the nodes. There are five types of nodes : CDFG nodes are hierarchical nodes containing other CDFG structures, control nodes are composed of control structures (loops, branches), DFG nodes contain straight forward code (without control and hierarchy), data nodes contain data used in the application (for memory requirement analysis) and processing nodes describes the processing operations of the graph. The Estimation Flow begins with the processing part of the application. The first step is to list all the operation types and data formats of the whole graph. Then, the most suited operator is selected among the component library for each type processing node. Once the re-

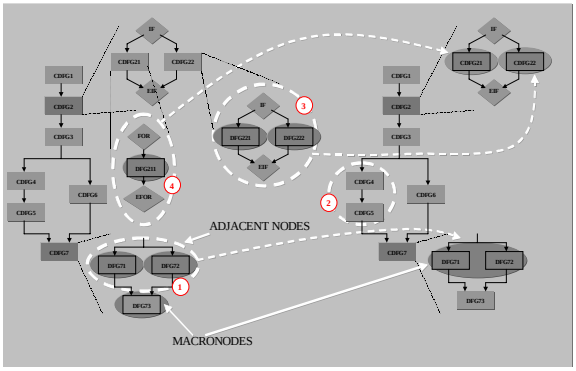

Figure 2: the SPF model

source allocation step has been performed, the clock period is set as the delay of the fastest functional unit. So each node can be assigned a number of clock cycles, in order to apply a classical DFG estimation method based on

the computation of the as soon as possible (ASAP) and as late as possible (ALAP) dates [1][4], or a scheduling of the graph, according to the specification complexity. Each DFG (wich is the basic block of the HCDFG model) is estimated in a dynamic way, i.e. for many time constraints (figure 3). Then, adjacent

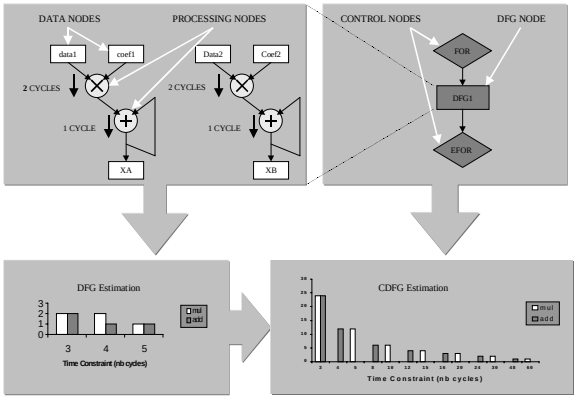

Figure 3: Example of CDFG estimation

node analysis is performed in order to go up trough the entire hierarchy (figure 4). There are four types of dependencies between two adjacent nodes :

- Parallel execution (fig 2.(1)) : the resulting execution time is the maximum of the two execution times, and the number of resources of type k is the sum of the number of resources of type k for each node (for low cost operators), or may be inferred from the analysis of the temporal use of the operator (for high cost operators, i.e. area and power consuming operators).
- Sequential execution (figure 2.(2)) : execution time is the sum of the two execution times and resource re-using is performed by taking the maximum number of resources for the two nodes.
- Branches (fig 2.(3)) : the estimation method is based on the Probability-based Analysis of Control Flow Graph Models [2] where each transition arc between the nodes has a probability of execution. The resources estimation results of the nodes are weighted by the transition arc probabilities.
- Loops (fig 2.(4)) : There are two kind of loops depending on whether the number of iterations is known or not. For deterministic loops, we can consider many configurations. For example, a 40 iteration loop can be implemented as 40 iterations of one branch, or 20 iterations of 2 branches, 10 iterations of 4 branches, ... Concerning non deterministic loops, the number of iterations may be user defined or determined by profiling. There can only be one configuration of one loop repeated N times in this case.

Two adjacent nodes are clustered into macronodes and each point of the respective dynamic curves are analysed according to the kind of dependency in order to compute the resource requirement for the resulting macro node (nodes merging). Hence, dynamic curves are obtained at the end of each adjacent nodes combination, until the top of the hierarchy is reached. A global time constraint GTC_i is computed for each DFG_i of the graph and an exploration zone is defined in order to reduce the dynamic estimation algorithm complexity (figure 5). The number of exploration points around GTC_i can be user defined or obtained from a system level analysis, like the method described in [7].

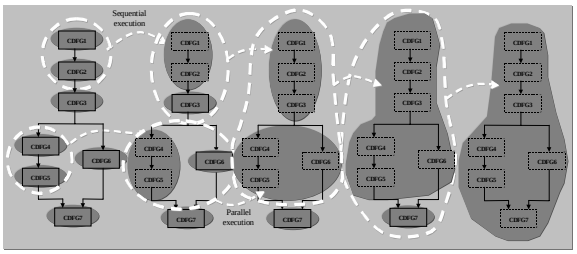

Figure 4: Adjacent nodes combination

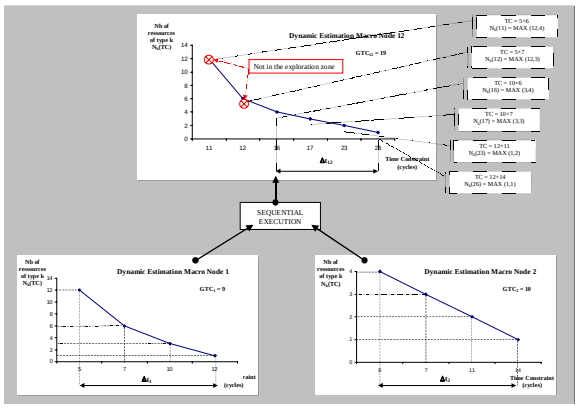

Figure 5: Nodes merging example (case of sequential execution)

4 Technology Mapping

Each FPGA technology has a charaterization file who contains all the characteristics of the architecture (interconnection resources, memory resources and control logic like the Embedded System Blocks in the ALTERA APEX family) and the basic functional units performances (arithmetic and logic operators). Area and time characterization is operated by synthesizing each basic functional unit for several data formats. The synthesis tool gives the delay (critical path) and ressource utilization of the component (number of logic cells, ...). Concerning power consumption, it is obtained by measuring the supply current of the FPGA with a precision ampere meter. Power characterization is performed through two parameters : static power is the power consummed when the FPGA is programmed and there is no clock activity, dynamic power is the power consummed when data are sent through the inputs and is parameterized by the clock frequency. Then, the average dynamic power consumption per access is obtained as follow : $ADP_{op_k} = U * I/F_h$ where U is the supply voltage, I is the measured current and F_h is the running frequency. After the step of Resources Estimation, we know the number of functional units of each type in function of a variable time constraint. The area, delay and power consumption of each functional unit is characterized in the component library (figure **??**). Time execution values are obtained by projecting the clock period value with the time constraint defined in number of clock cycles. We compute the total area of the whole application by adding the contribution of each functional unit, knowing the area of each of them. Concerning

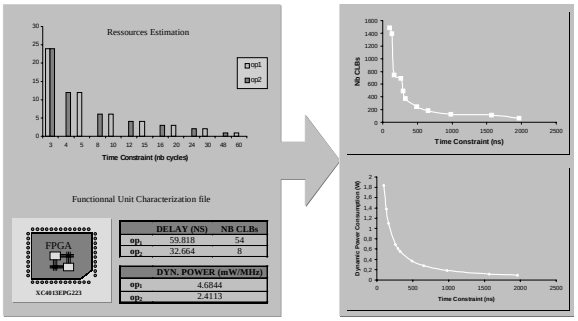

Figure 6: Technology mapping

power estimation, the number of use of each functional unit is computed. Then, knowing the average power consumption by access makes total power consumption easy to compute : $P_{op_k} = ADP_{op_k} * N_k^{operations}/T_{ex}$, where $N_k^{operations}$ is the number of operations of type k in the graph and T_{ex} is the execution time of the application.

5 Results

We present here the estimation results coming from a speech coding application (G.722 recommendation). The example is a simple filter based on a 12 iteration loop whith basic multiplication addition pattern. The target FPGA is a Xilinx XC4000. Figure 7 shows the Configurable Logic Blocks (CLBs) utilization rate, for the processing unit, compared to the synthesis results. There is slight deviation between the estimated time constraints (X axis) increasing when the design area (Y axis). This is due to the interconnection delays whose influence gets bigger whith the FPGA utilisation rate. Figure 8 shows the average power consumption estimation for a

25 MHz clock frequency running compared to the Xilinx estimation method. We can notice

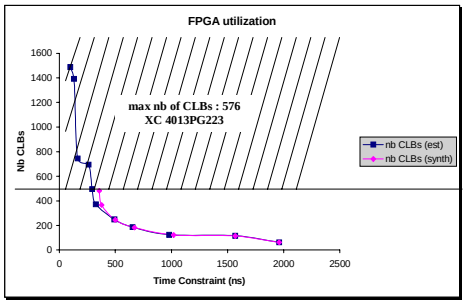

Figure 7: Area estimation results.

the same deviation concerning temporal estimation for low time constraint values. The

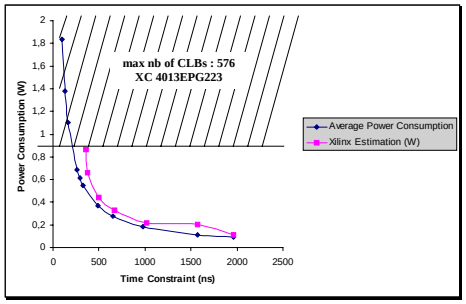

Figure 8: Power estimation results.

analysis of the power estimation curves shows a great difference between the two estimation tools, and will be compared to measurements in a future work. The first results presented here show that the estimator is quite acurate, with a maximum deviation of 3.31% for CLB estimation and 17.46% for time constraint estimation (average error 7.9%). The next step of our work is to include the interconnection delays to correct the temporal estimation error [5]. Then, the memory requirements (size, number of ports, address generator, ...) will be analysed. Finally, for the control part of the application, a same approach can be done by estimating the combinational logic (number of logic gates for the control unit estimation, depending on the number of states, number of functional units, number of memory ports, [2]).

6 Conclusion and perspectives

We have presented a new performance estimation technique for FPGA based systems. The estimator gives realistic area vs. time estimation values through dynamic characterization and includes power consumption. This methodology can be applied to the estimation of ASIC implemented functions by defining a new characterization file and interconnection cost model. So a global performance estimator for codesign systems can be developped from this work by including interdependency analysis (communications between the different untits of the architecture, memory sharing, ...)

References

[1] J.P. Diguet, *Estimation de Complexité et Transformations d'Algorithmes de Traitement du Signal pour la Conception de Circuits VLSI*, PhD Thesis, Université de Rennes I, 1996.

[2] S. Narayan and D.D. Gajski, *Area and Performance Estimation from System-Level Specifications*, Technical Report, University of California, 1992.

[3] S.Y. Ohm, F.J. Kurdahi, N.Dutt, M. Xu, *A Comprehensive Estimation Technique for High-Level Synthsis*, Proceedings of the 8th IEEE International Symposium on System Synthesis (ISSS'95), Cannes, France, September 1995.

[4] A. Sharma and R. Jain, *Estimating Architectural Resources and Performance for High-Level Synthesis Applications*, IEEE Trans. on VLSI Systems, vol 1, no 2, June 1993.

[5] M. Xu, F.J. Kurdahi, *Area and Timing Estimation for Lookup Table Based FPGAs*, Proceedings of ED&TC, March 1996.

[6] J.P. Diguet, G. Gogniat, P. Danielo, M. Auguin, J.L. Philippe, *The SPF Model*, accepted at FDL, Tübingen, Germany, September 2000.

[7] H. Thomas, J.P. Diguet, J.L. Philippe, *A methodology for an Application Profiling at a System Level*, SiPS, Taïwan, October 1999.