



# NoC Dimensioning from Mathematical Models

Virginie Fresse<sup>1</sup>, Catherine Combes<sup>1</sup>, Matthieu Payet<sup>1,2</sup> and Frédéric Rousseau<sup>2</sup>

<sup>1</sup>Hubert Curien Laboratory UMR CNRS 5516, Jean Monnet University – University of Lyon, PRES Lyon- 42000, Saint-Étienne, France

<sup>2</sup>University of Grenoble Alpes, TIMA Laboratory, CNRS, F-38031 Grenoble, France

Received 20 July 2015, Revised 9 January 2016, Accepted 17 January 2016, Published 1 March 2016

**Abstract:** NoCs (Network-on-Chip) have emerged as efficient scalable and low power communication structures for SoC (System-On-Chip). Two main challenges are pointed out when prototyping a SoC on a reconfigurable chip such as FPGA (Field Programmable Gate Array). The first challenge is to tune a NoC according to the application requirements by exploring all design solutions. The second challenge is to dimension the FPGA resources regarding the previously selected appropriate solution. Usually, dimensioning of FPGA resources is done by several runs of automatic synthesis processes to evaluate if the number of resources fits to the selected FPGA device. Finding the most appropriate solution and FPGA dimensioning are time consuming and the design space exploration is not fully done in order to decrease the exploration time. A more appropriate solution would be analytic models of the NoC on a FPGA device. Mathematical modelling consists in identifying links between the NoC parameters and the FPGA resources using a database and in extracting relations between them.

In this paper, we present a methodological framework to estimate the number of resources required for a given communication architecture and the task graph of the application. The framework contains 4 steps: the design or the selection of the NoC, the data collection, the data analysis from which a model is deduced. The database obtained in the data collection step contains the synthesized results of each NoC configuration. Two NoCs, with a mesh topology and different characteristics, are used to provide two databases. The methodological framework provides the most appropriate models that are identified using predictive modelling. The evaluation of each model shows that the relative error is less than 5% in most cases. It is therefore possible to tune the most appropriate NoC and to estimate the required resources in a short exploration time without the synthesis steps.

**Keywords:** NoC on FPGA, mathematical models, resource dimensioning, methodological framework, area estimation

## 1. INTRODUCTION

Systems-on-Chip (SoC) using Network-on-Chip (NoC) are the most appropriate systems for real time embedded applications. The SoC is a set of hardware or software IPs (Intellectual Properties) connected to the NoC. NoCs are emerging communication structures as they provide high bandwidth and high scalability with low power. The NoC structure is more extensible and parallelizable than traditional buses but NoC based systems require a reliable methodology for better design space exploration. Indeed, the designer has to parameterize all the parameters of the NoC according to the application to optimize communication times between cores. Exploring all appropriate solutions is an intensive time process because of the sheer number of parameters required for the NoC.

The Field Programmable Gate Array (FPGA) devices are widely used for prototyping systems. FPGA can then be used to emulate the timing performances of NoC for

each set of parameters in a rapid design space exploration. Emulation provides precise timing and power evaluations in a shorter cycle than simulation [1]. But before emulation, each NoC candidate has to be synthesized then placed and routed to obtain the number of resources. Resources can be obtained either after the synthesis or after the place and route process. With large FPGAs, this process can take several hours for one NoC candidate. Usually, the designer selects the NoC parameters without exploring all the candidates, thereby saving a significant amount of time in the development process but the chosen solution is not always optimal. Area estimation is important in order to find architectural solutions that: 1) suit the target FPGA concerned 2) correspond to the requirements of the application 3) ensure efficient timing.

In this paper, we propose a methodological framework to extract the most appropriate mathematical model according to the selected NoC for FPGA resources dimensioning. In this framework validated by an expert,



the design can define the parameters of the NoC structure according to the algorithm in a short exploration time.

The paper is organized into 6 sections. Section 2 presents the motivation of works. The section 3 describes the methodological framework and the four steps of this framework are detailed. The validation of the framework is given in section 4. An analysis of results and a discussion are explained in section 5. Section 6 concludes the paper.

## 2. MOTIVATION OF WORKS

NoCs have emerged as efficient scalable and low power communication structures for many-core SoC (System On Chip including several hundred or thousands of cores). Many NoCs are designed for FPGA devices [20][21][22] and application-specific NoC design flows are proposed [23]. In the design flows, the application described as a task graph is mapped to the topology graph. The topology graph is the NoC structure with all parameters already specified. These existing design flows do not explore the design space of the NoC. Many works are based on task mapping exploration as mapping plays an important role in performance of NoC design [7][8][9][13].

The design space exploration of NoCs is commonly formulated as constrained optimization problem. Design Space Exploration (DSE) refers to the activity of exploring design alternatives prior to implementation [29]. The challenge of DSE is to explore the sheer size of the design space and to find the best candidates. Typically a large system has billions of possibilities as parameters of NoCs are abundant. The designer must select the topology, the number of nodes, the size of flits, the commutation mode, the routing algorithm, the size of buffers and many other parameters. Enumerating every point of the design space is prohibitive and is time consuming [28].

Most of existing works based on NoC design space exploration are proposed to explore the energy consumption or the timing performances.

Design space explorations for the NoC are mainly based on power consumption and timing on ASIC (Application Specific Integrated Circuit) [1][21][24][3][6]. Power models at different abstraction levels have also been proposed for a variety of networks in the past [10][11][12]. ORION is a tool designed for fast and accurate power and area model on Integrated Circuit (IC) [5]. The tool explores the area occupied ( $\text{mm}^2$ ) and the power (mW) by the NoC used on 65 nm chips when routers and links increase. The 3D Tezzaron design flow also explores the 3D NoC to reduce the area of the chip and interconnects on ASIC to optimize power [13]. A system level approach is proposed to explore the NoC design space with an

objective to minimize the energy consumption and link bandwidth (timing). These works concern power and timing evaluation for ASIC. A power model is formulated for links and switches for the Nostrum NoC using Synopsys Power Compiler in [15]. In [16] authors propose a performance analysis and dimensioning method for NoC based platforms (NoC-ADM). A software tool model NoC based systems is developed to determine Bounds on delay, buffer size and throughput and verify timing constraints. Components exploration for multimedia applications is simulated on the Nostrum NoC.

The Exploration of the Design Space for NoC on FPGA has not been fully explored yet. FPGA are devices with a number of available resources to be used. The exploration should consider these resources as the maximal number of resources to be used. Models for resource dimensioning are required for exploration of NoC on FPGA. A power area analysis of NoCs in FPGAs has been proposed in [10]. The analysis is based on the analysis of power and area of the router for the  $4 \times 4$  torus topology. This work only considers the routing blocks, no any others blocks or routing. The number of links varies according to the position of the router so that it has a huge impact of the total number of resources. It is necessary to analyse the global structure to obtain a precise model depending on the topology.

NOCDEX is a tool to evaluate the impact of various options on area, number of cycles and execution time on FPGA [6]. The tool evaluated the number of cycles according to the number of slices and the maximum frequency for a cascade NoC with 4 masters and 4 slaves.

A resource usage model for NoC enables the optimization of design parameters under resource constraints. The size and parameters of the NoC should be decided according to the size of the task graph and the available FPGA resources. Today, NoC mapping heuristics do not cope with FPGA resources constraints. NoC dimensions are determined before the mapping step. Using mathematical models enables to quickly evaluate the impact of NoC parameters on FPGA resource usage as far as estimation errors are minimized.

The contribution of this paper is to propose models to explore the design space of NoCs to estimate the area metric on FPGA. Explorations are constrained by the target FPGA and the data flow graph from the application.

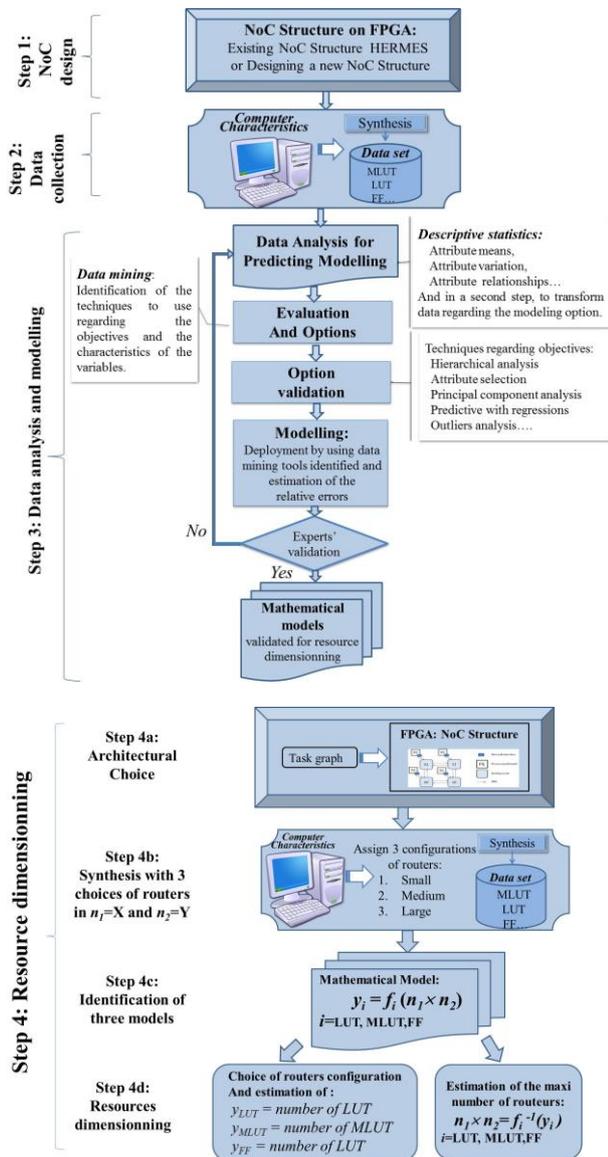


Figure 1. Methodological Framework

### 3. METHODOLOGICAL FRAMEWORK

A model for system dimensioning requires accurate estimations of the resources required by the application. First, we describe the characteristics of the FPGA structures explored using our approach. Second, we present the methodological approach for resources dimensioning. The methodological framework is illustrated in figure 1. The methodological framework is split into four steps:

- 1) Selecting the appropriate NoC structure,
- 2) Collecting the data required for analysis,
- 3) Analyzing the data to identify the most appropriate mathematical models,

4) Exploring the models to define the most appropriate parameters of the NoC according to the algorithm requirements.

The principle of the framework is to select or design the parameterized NoC structure used to connect the IP cores of the application. Then the data are analyzed (i.e. the FPGA resources according to the varying parameters of the NoC) using the analysis and modeling steps. The data analysis is used to find the most appropriate models and the associated variables to be used. Models are then validated by experts at the end of the step 3. When models are validated, the designer can define these models in the step 4 according to the FPGA target and software development tool. Using these models the application designer can tune the NoC with the resources dimensioning using the task graph of the application from the models provided by a restricted number of synthesis processes. The resource dimensioning is faster and few synthesis processes are required to define the models, accelerating significantly the exploration time.

#### A. Step 1: selecting the NoC

The first step is selecting the appropriate NoC structure. Any NoC structure designed in HDL (Hardware Description Language) can be used. Our experiments were conducted on two NoCs, one designed especially for the purpose (AdOCNet – Adjustable On-Chip Network) and one existing NoC that is, often used in the research community (Hermes) [2].

The NoC structure (topology, flow control, virtual channel, scheduling and routing algorithm) is fixed for a model. Table I shows the characteristics of the two NoCs used for resource modeling.

TABLE I. NOC CHARACTERISTICS

	Topology	Flow control	Virtual channel	Scheduling	Routing algorithm
Hermes	2D Mesh	Credit based	2	Round robin	XY
AdOCNet	2D Mesh	Handshake	No	Round robin	XY

A 2D mesh topology is commonly used since it fits best to the FPGA topology. The two NoCs differ in their flow control and the number of virtual channels. Such differences have a significant impact on FPGA resource usage.

#### B. Step 2: Data collection

The data were the FPGA resources in the post synthesis report in the VIVADO 2012 development tool (integrating the Xilinx synthesis tool) according to the NoC parameters [30]. These resources may differ for other software, but are basically of the same type. From each synthesis processes, the results stored in a database are:



- $n_1$ : the number of routers in the X-axis.
- $n_2$ : the number of routers in the Y-axis.
- $n_3$ : the depth of the buffer.
- $n_4$ : the size of the flit.
- LUT: the number of Look Up Table used.
- MLUT: the number of Memory LUT used.
- FF: the number of Flip Flop used.

### C. Step 3: Data Analysis

The database of observed results was analyzed to identify links between variables and LUT, MLUT, FF regarding the variables  $n_1$ ,  $n_2$ ,  $n_3$  and  $n_4$ . The aim was to mathematically model the relationships between the input configuration of the NoC and material resources used without going through the synthesis step.

Data mining is used to automatically locate useful information among large quantities of data. Data mining can be both predictive and descriptive: when predictive, the aim is to predict the value of a particular attribute given existing data, when descriptive; the aim is to derive patterns that summarize underlying relationships in the data. Data mining is thus an integral part of knowledge discovery, which is the overall process of converting raw data into knowledge by obtaining selecting useful information from the data. Five core data mining tasks have been identified [17][19]: 1) predictive modelling, 2) attribute selection, 3) association analysis, 4) cluster analysis, 5) anomaly detection.

In our approach, the most appropriate data mining task is predictive modelling. The task is to build a predictive model for a target variable, based on explanatory variables. Classification and regression are ways of predicting a discrete or continuous outcome (e.g. whether or not somebody will do something) c.q. an extrapolation of continuous output (e.g. what the future value of a measurement will be). Regression analysis is a statistical tool for the investigation of relationships between variables. Usually, the investigator is looking for the causal effect of one variable on another.

First, we checked linear relationships between couples of variables using Pearson's correlation. It is also possible to cluster variables in terms of their correlations. Two variables have a pair of values for each sample, and measures of distance and dissimilarity between these two column vectors can be considered. The similarity between variables is measured: this can be in the form of correlation coefficients or other measures of associations. The result of a cluster analysis is a binary tree, or dendrogram, with  $n-1$  nodes. The branches of this tree are cut at a level of similarities obtained in our case using the correlation between all the variables.

Second, for predictive modelling, when the outcome or class is numeric and all the attributes are numeric, linear regression is the logical choice. The standard way of dealing with continuous prediction is writing the

outcome as a linear sum of attribute values with appropriate weights such that:

$$y_{\text{pure}} = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n$$

where:  $y_{\text{pure}}$  is the class,  $x_1, x_2, \dots, x_n$  are the attribute values of the variable  $X_1, X_2, \dots$  and  $X_n$ ,  $w_1, w_2, \dots, w_n$  are the weights of each variable  $X_1, X_2, \dots$  and  $X_n$ .

In this way, we obtained a regression equation to be used to determine the corresponding weights for each variable, a well-known procedure in statistics. The weights were calculated from the training data, the model minimizes this sum of squares by choosing the appropriate coefficients. Linear regression is an excellent, simple method for numeric prediction that has been widely used in statistical applications but is very sensitive to outliers.

The difference between the observed value of the dependent variable ( $y_{\text{obs}}$ ) and the predicted value ( $\hat{y}_{\text{pure}}$ ) is called the residual ( $e$ ). Each data point has one residual.

$$\text{Residual} = \text{Observed value} - \text{Predicted value} \quad (1)$$

$$e = y_{\text{obs}} - \hat{y}_{\text{pure}} \quad (2)$$

Both the sum and the mean of the residuals are equal to zero. That is,  $\sum e = 0$  and  $e = 0$ .

Tables II and III list the respective Pearson's correlations for the AdOcNet (29 synthesis) and Hermes (152 Synthesis) [18].

TABLE II. PEARSON'S CORRELATION FOR ADOCNET

	$n_3$	$n_4$	$n_2$	$n_1$	FF	LUT	MLUT
$n_4$	0.033						
$n_2$	-0.045	-0.045					
$n_1$	-0.071	-0.071	0.551				
FF	-0.122	-0.124	0.731	0.954			
LUT	-0.093	-0.127	0.721	0.959	0.998		
MLUT	-0.340	-0.292	0.596	0.789	0.876	0.850	
$n_1 \times n_2$	-0.089	-0.089	0.733	0.959	0.997	0.999	0.834

TABLE III. PEARSON'S CORRELATION FOR HERMES

	$n_4$	$n_2$	$n_1$	FF	LUT	MLUT
$n_2$	0.170					
$n_1$	-0.157	-0.237				
FF	0.364	0.509	0.531			
LUT	0.460	0.503	0.467	0.992		
MLUT	0.627	0.463	0.339	0.932	0.971	
$n_1 \times n_2$	-0.056	0.468	0.680	0.880	0.812	0.649

When Pearson's coefficient is greater than 0.7, the closer the points are located to one another on the line (a perfect correlation is 1, indicating that all points fall directly on a line). Concerning the two NoCs, there is a strong positive correlation between the observed resources. When similarities between groups were analyzed, the strongest correlation was found between FFs and LUTs (respectively 0.992 for Hermes and 0.998 for AdOcNet). Strong correlations were also found between MLUT and LUT, and between FF and MLUT. There is also a strong correlation (lesser but significant) between  $n_1 \times n_2$  and the number of FF and LUT. One main difference between both NoCs is the use of resources



according to  $n_3$  parameter. AdOcNet only uses MUTs, LUTs and FFs for the NoC structure whereas the Hermes NoC uses BRAM (Block RAM) to implement buffer (the number of BRAM does not change for the varying size of buffer).  $n_3$  is respectively considered as a variable for AdOcNet and a constant for Hermes. Therefore  $n_3$  is not considered in the Person's correlation and hierarchical analysis for Hermes. Next, we identified strongly correlated group of variables by applying hierarchical analysis on the variables [17]. Fig 2 presents the dendograms respectively AdOcNet and Hermes. A strong correlation indicates a high degree of similarity. A weak correlation indicates a low degree of similarity.

For AdOcNet, a group contains five variables with a similarity value equal to 97.96 ( $n_1$ , FF, LUT, MLUT,

$n_1 \times n_2$ ). For Hermes, a group contains four variables with a similarity value equal to 94.00 (FF, LUT, MLUT,  $n_1 \times n_2$ ). In conclusion, there are strong linear links between the variables (see Table IV).

For each NoC, FF, LUT, MLUT are strongly correlated with  $n_1 \times n_2$ . Our objective was to identify possible linear relationships between these resources and  $n_1 \times n_2$ . Figure 3 validates the linear relationship between the number of MLUT and  $n_1 \times n_2$ . Figure 4 validates the linear relationship between the number of FF and  $n_1 \times n_2$ . Figure 5 validates the linear relationship between the number of LUT and  $n_1 \times n_2$ . We then checked it on the other resources for both NoCs.

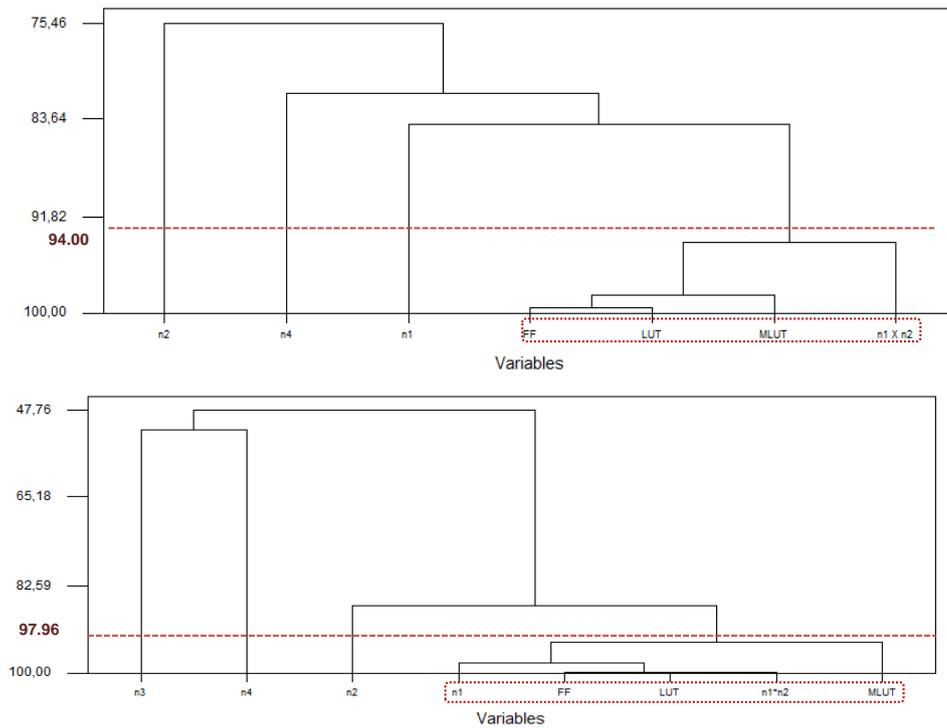


Figure 2. Hierarchical analysis of similarities between the variables (top: AdOcNet, bottom: Hermes)

TABLE IV. IDENTIFICATION OF THE HIGHLY CORRELATED VARIABLES FOR EACH NOC.

NoC	Highly correlated variables	Similarity Level
AdOcNet	FF LUT $n_1 \times n_2$ $n_1$ MLUT	97.96
Hermes	FF LUT MLUT $n_1 \times n_2$	94.00

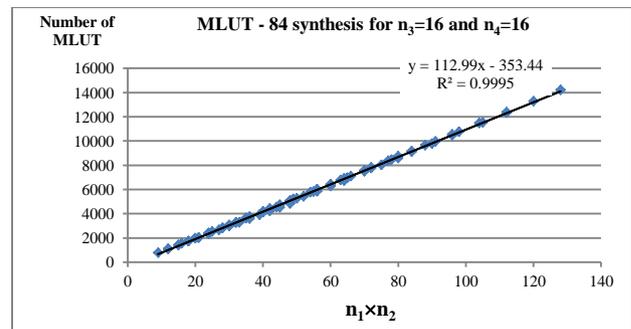
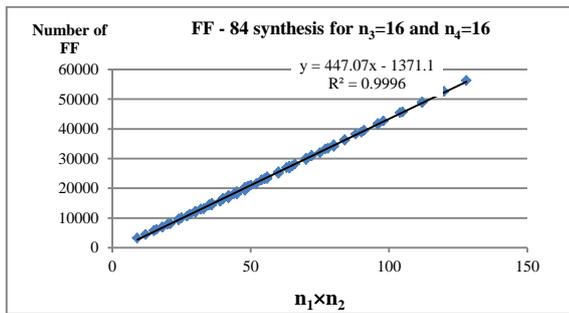
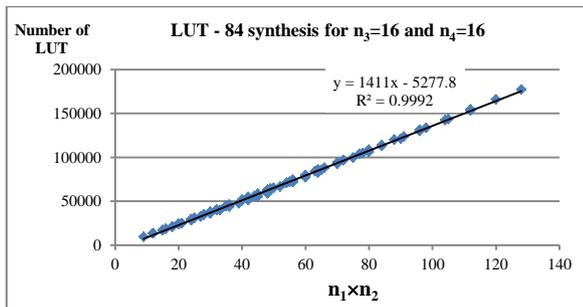


Figure 3. Number of MLUT regarding  $n_1 \times n_2$

Figure 4. Number of FF regarding  $n_1 \times n_2$ Figure 5. Number of LUT regarding  $n_1 \times n_2$ 

Graphical analysis is a very effective way to investigate the adequacy of the fit of a regression model and to check the underlying assumption. Residual plots allow us to examine the goodness-of-fit in regression. The Henry's test (Normal probability plot) and the histogram of residual values for FF are depicted in figure 6 and figure 7. The residual values for LUT and MLUT are similar to the FF's residual values. The Henry's straight line (figure 6) is a graphical method for adjusting a Gaussian distribution with that of a series of observations. The normal scores are not aligned, there are outliers amongst results. The histogram of residual value (figure 7), used to check the variances, is not normally distributed around zero (a symmetric well-shaped histogram evenly distributed around zero indicates that the normality assumption is likely to be true). We observe that the regression model does not completely fit the data. Even if the model does not perfectly fit the data the result is satisfactory and the linear model is approved.

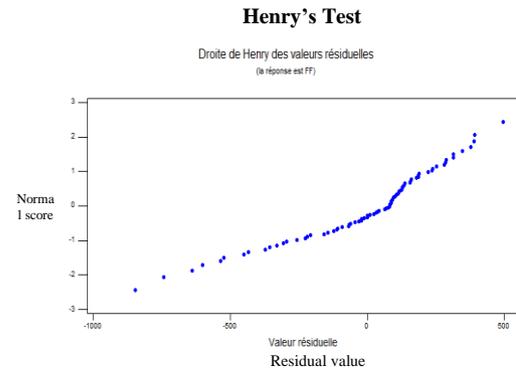


Figure 6. Henry's straight line test

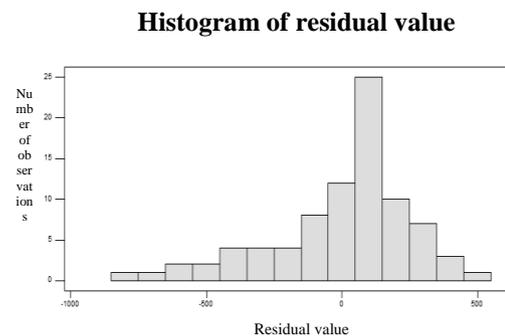


Figure 7. Histogram of residual value

Therefore, we looked at the model and a regression model seems to be the good solution. The linear regression with one variable ( $n_1 \times n_2$ ) is appropriate in our case. The multiple linear regressions are not necessary. The step 3 is completed and models are validated by expert. The designer can therefore use these linear models to dimension the parameters of the selected NoC in the step 4 of the framework.

#### D. Step 4: Resource dimensioning

As linear regression model is adopted, the resource dimensioning can be done using 3 synthesis processes (extracting 3 points). The objective is to define the resources used according to the algorithm, the NoC selected and the target FPGA. The algorithm is described as a task graph and the designer identifies the number of tasks to be connected to the NoC. As task and data parallelism can be exploited, a minimal and maximal number of tasks are set. From a range of nodes  $n_1 \times n_2$  connected to the NoC (local router), the designer dimension the FPGA resources according.

Figures 8 and figure 9 show respectively the regression models of LUT usage for Hermes and AdOcNet. Figures 10 and figure 11 show the regression models of MLUT usage for Hermes and AdOcNet. Figures 12 and figure 13 show respectively the regression models of FF usage for Hermes and AdOcNet. For



Hermes, the size of flits has a significant impact on all the resources. There are three linear regression models for each resource. All these three are significantly different for LUT, LMUT and FF. The AdOcNet is a more regular structure as linear regressions are very closed whatever the size of buffer and the size of flits.

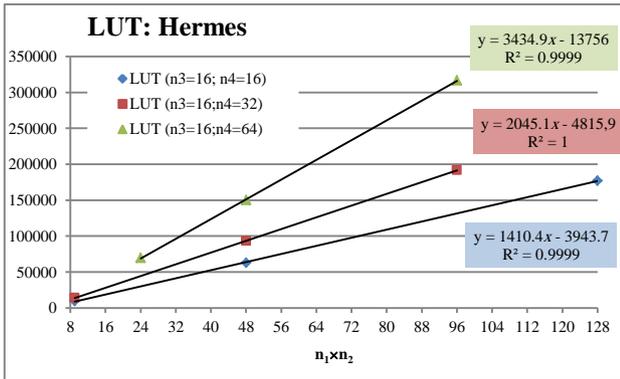


Figure 8. Regression models for LUT for Hermes (x in the equations corresponds to  $n_1 \times n_2$ )

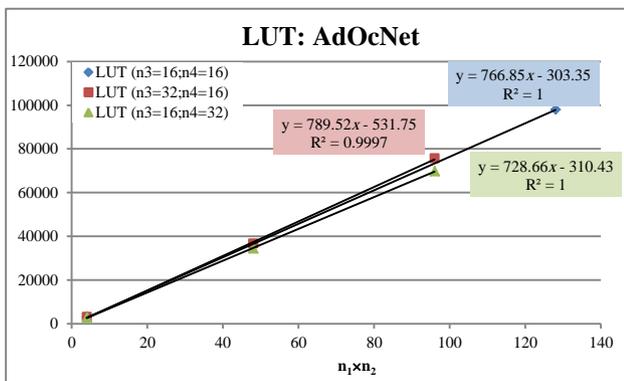


Figure 9. Regression models for LUT for AdOcNet (x in the equations corresponds to  $n_1 \times n_2$ )

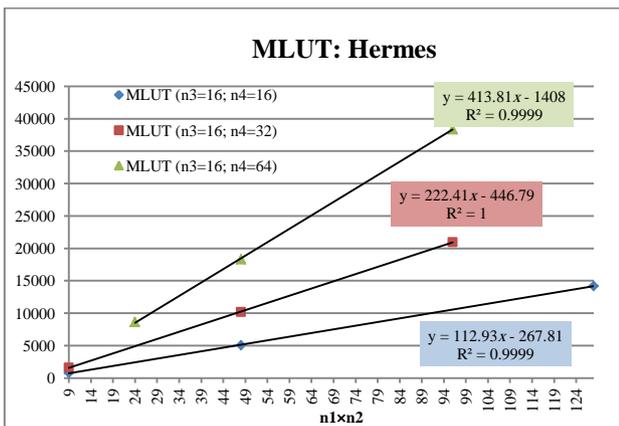


Figure 10. Regression models for MLUT for Hermes (x in the equations corresponds to  $n_1 \times n_2$ ).

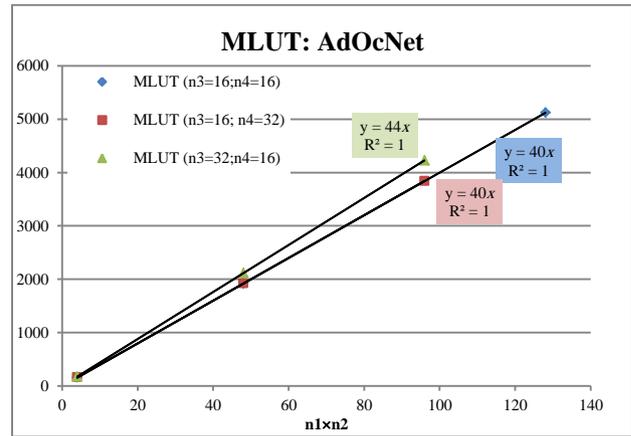


Figure 11. Regression models for MLUT for AdOcNet (x in the equations corresponds to  $n_1 \times n_2$ )

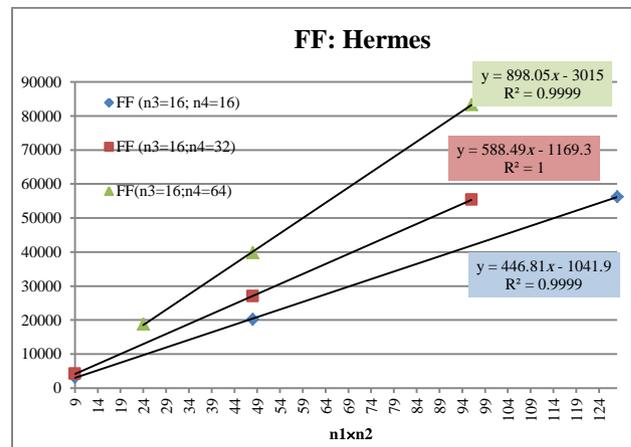


Figure 12. Regression models for FF for Hermes (x in the equations corresponds to  $n_1 \times n_2$ )

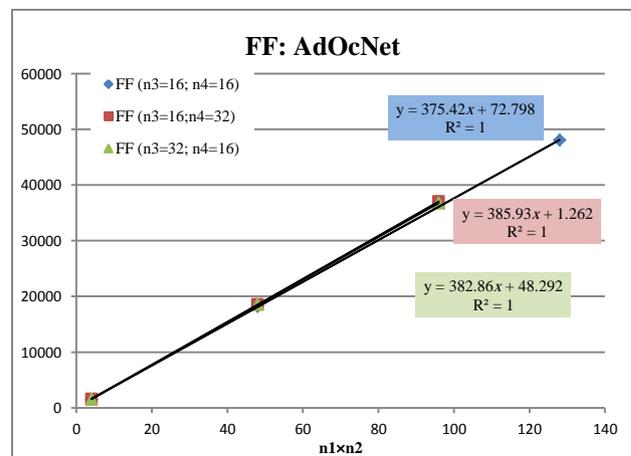


Figure 13. Regression models for FF for AdOcNet (x in the equations corresponds to  $n_1 \times n_2$ )

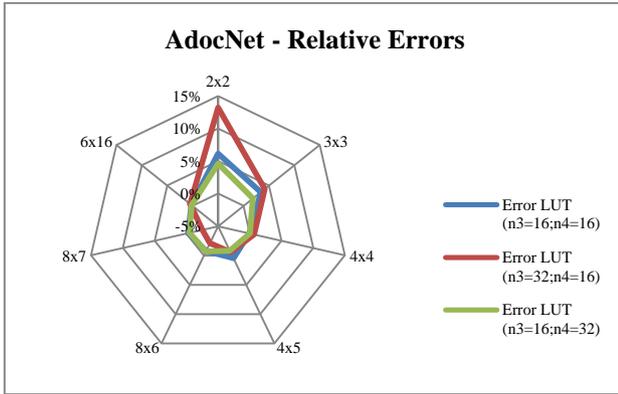


Figure 14. Relative error rate for AdOcNet.

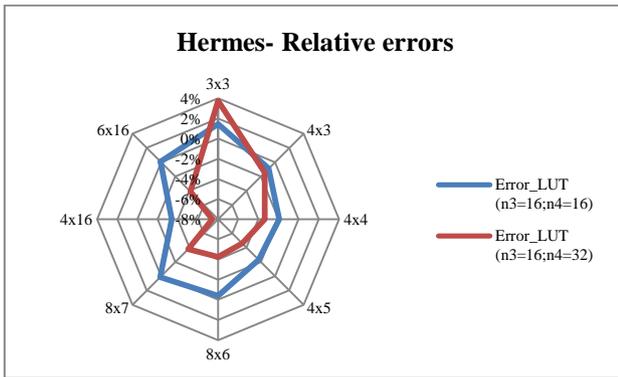


Figure 15. Relative error rate for Hermes.

Therefore, the number of synthesis processes is higher for the Hermes than AdOcNet. It is necessary to extract 3 points (3 synthesis processes) for each size of flits for Hermes. It is possible to only extract three points for the AdOcNet models whatever the size of flits and the size of buffers.

#### 4. VALIDATION

The validation shows that dimensioning provides good resource results. From new synthesis from the NoC, the relative errors for LUT, MLUT and FF are calculated between the synthesized results and the resources obtained from models. The negative error rate indicates that the resources of the models are above the synthesized resources. The positive error rate indicates that the resources of the models are below the synthesized resources. The model overestimates the resources for the first case and underestimates the resources for the second case. The relative error of LUT for Hermes and AdOcNet are depicted in figures 14 and 15.

For Hermes, the error rate is from -6% to +2% for  $n_3=16$  and  $n_4=16$ . The error rate is from -8% to 4% for  $n_3=16$  and  $n_4=32$ . The min values are for the  $4 \times 16$  NoC and the max values are for the  $3 \times 3$  NoC. The error rates are similar for MLUT and FF for Hermes and AdOcNet. As

the relative error is positive, the number of estimated resources is below the number of FPGA resources used. The error rate decreased to less than  $\pm 2.5\%$  for bigger sizes of NoC. This indicates that the analytically estimated results are a little bigger than the results obtained after synthesis (synthesized results) for small sizes of NoC. If the  $(2 \times 2)$  or  $(3 \times 3)$  router sizes of NoCs are not taken into account, the intervals of relative errors are  $[-7.90\%; 6.90\%]$  for Hermes and  $[-2.52\%; 6.14\%]$  for AdOcNet.

Estimations for small sizes ( $2 \times 2$  or  $3 \times 3$ ) of NoCs can be replaced by direct synthesis because the synthesis time is not prohibitive, in that cases. However, buses or point to point communication can be better suited for small designs.

Therefore, the mathematical models can be obtained from only 3 synthesis of the NoC. The selected sizes should be over  $3 \times 3$ .

#### 5. ANALYSIS AND DISCUSSION

In this section, three points are discussed:

- The high error rate for small NoCs,
- The impact of the synthesis options on the models,
- The impact of changes on the NoC structure on the models and the linearity.

##### A. Resources dimensioning for small NoCs

We showed in the previous section that the models for small NoC provide the number of estimated resources a little different from the synthesized resources (considering than  $\pm 10\%$  should be the maximal relative error to dimension the resources). The exploration of resources for small NoCs can be achieved with two ways. The first way is to give the resource models for small NoCs by selecting appropriate points. In this case, points from the synthesis processes are extracted from small NoC. Another way is to evaluate the number of resources directly with synthesis process for each configuration. Figure 16 shows the synthesis time according to the NoC parameters (variables  $n_1 \times n_2$ ,  $n_3$  and  $n_4$ ). The synthesis time depends on the PC used and mainly on  $n_1 \times n_2$ . Estimations for small sizes ( $2 \times 2$  or  $3 \times 3$ ) of NoCs can be replaced by direct synthesis process because the synthesis time is not prohibitive, in that cases.

The models are used to speed up the dimensioning of the NoC ( $n_1 \times n_2$ ). Small sizes of NoC can be synthesized quickly (less than 5 minutes). Therefore having a higher error rate for small size of NoC does not significantly affect the exploration time.

##### B. Synthesis options on models

Models obtained for the FPGA resources depend on the synthesis options used. Options can be used according to the type of dedicated SoC. In the previous experiments and analysis, the default options are set in the Xilinx



Synthesis Tool (XST). In such configuration, XST can use all available resources of the target FPGA. This can be used for heterogeneous platform using hardware IP or software IP (embedded processor). For some cases, some resources can be reserved for dedicated IPs: LUT, FF for hardware IPs (using also DSP blocks) and MLUT for software IP (for instruction and data memories). Therefore, models depend on the type of dedicated SoC and the FPGA resources. Another major resource is the RAM blocks that can be used by the NoC.

The impact of these RAM blocks is discussed in the following analysis. Two synthesis options are considered when synthesizing the AdOcNet:

- Opt\_BRAM\_0: XST cannot use the RAM blocks (BRAM) as they are already used by soft IP. Such configuration should be considered when using MPSoC,
- Opt\_BRAM\_1: XST can use as many RAM as required as IP do not require many RAM blocks, they can be used for the communication structure

Linear models for LUT, FF for both synthesis options in the AdOcNet are respectively presented below:

Model 1: the number of LUT with Opt\_BRAM\_0:

$$y_1 = 390x \quad (4)$$

Model 2: the number of LUT with Opt\_BRAM\_1:

$$y_2 = 330x \quad (5)$$

Model 3: the number of FF with Opt\_BRAM\_0:

$$y_3 = 720.6x - 83.4 \quad (6)$$

Model 4: the number of FF with Opt\_BRAM\_1:

$$y_4 = 687x - 32 \quad (7)$$

Model 5: the number of MLUT with Opt\_BRAM\_0 (the number of BRAM is 0):

$$y_5 = 44x \quad (8)$$

Model 6: the number of BRAM with Opt\_BRAM\_1 (the number of MLUT is 0):

$$y_6 = 3x \quad (9)$$

Where  $x = n_1 \times n_2$  for all models

The linear models for FF and LUT and MLUT/BRAM are different according both XST options.

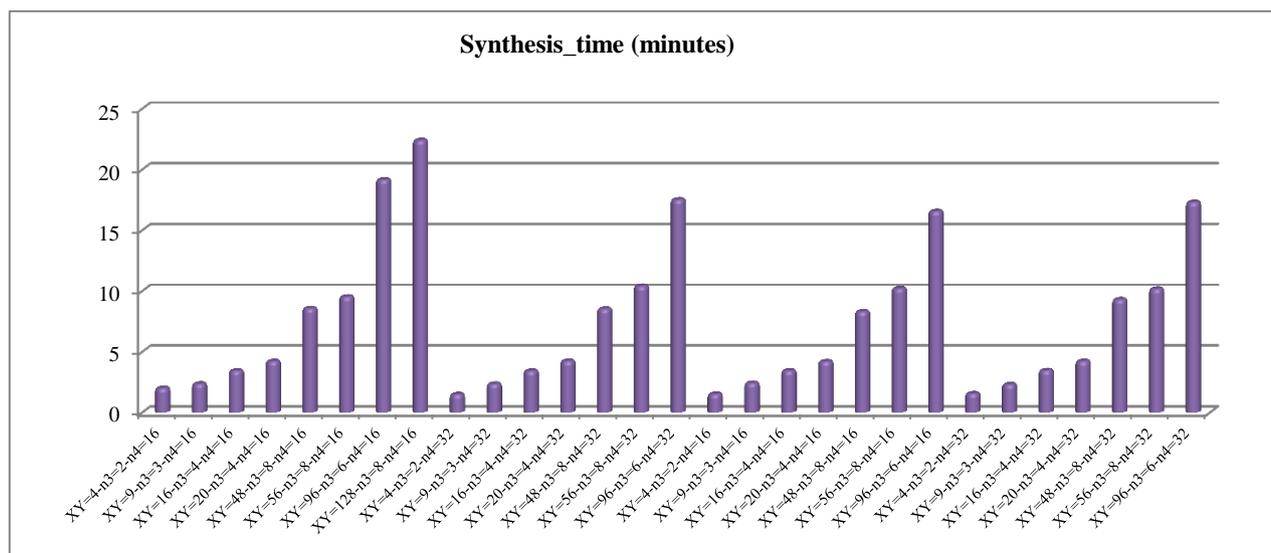


Figure 16. Synthesis times for several parameters ( $n_1 \times n_2$ ,  $n_3$  and  $n_4$ ) of the AdOcNet.

### C. Changes of NoC structure on models

Adding or replacing components in the NoC structure has an impact on the linearity of models. The models change but are still linear for the following cases:

- Changing the routing algorithm by another determinist and semi-adaptive algorithm [26],

- Replacing the type of control flow (credit based, handshake, adding virtual channels...) [26],
- Specifying other synthesis options.

Some significant changes in the structure can remove the linearity of the models. In [27], a management structure is added to Hermes. The structure is based either on a random access scheduling (BackOff) or a scheduled



access scheduling (Weighted Round Robin). The linearity is kept unchanged for the first structure but the models are not linear for the second structure.

## 6. CONCLUSION

In this paper, we showed the feasibility of identifying a mathematical model for NoC dimensioning on FPGA. This model can be extracted from only 3 synthesis processes done using the same FPGA with one computer and with the same synthesis options. Extracting 3 points is enough to provide a model of each FPGA resource (for each couple size of buffer and size of flits) according to the total number of routers. The resources estimated for Hermes are  $\pm 9\%$  of the obtained resources. The resources estimated for AdOcNet are  $\pm 7\%$  of the obtained resources. The AdOcNet structure is more regular than Hermes as the model is more precise. The model guarantees a reduction in the time needed for design space exploration (DSE) of NoCs. This model was validated by a comprehensive set of experimentations for a NoC using less than 10% of FPGA resources.

## REFERENCES

- [1] N. Genko, D. Atienza, G. De Micheli, J.M. Mendias, R. Hermida, F. Catthoor : "A Complete Network-on-Chip Emulation Framework". Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, Page(s):246-251 Vol. 1, 2005.
- [2] F. Moraes, N. Calazans, A. Mello : "HERMES: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip", Integration, the VLSI Journal, vol. 38, no. 1. pp 69–93. Oct. 2004.
- [3] A. Banerjee, R. Mullins, S. Moore : "A power and energy exploration of Network-on-Chip architecture". Proceedings NoC, pp163-172, 2007.
- [4] S. Bhat "Energy models for Network-on-Chip Components". MS thesis, department of mathematics and computer science, Royal Institute of Technology, Eindhoven, 2005.
- [5] A. B. Kahng, B. Lin, L.S. Peh and K. Samadi : "ORION 2.0: a fast and accurate NoC power and area model for early stage design space exploration". Design Automation and Test in Europe Conference and Exhibition, April 2009.
- [6] X. Li, O Hammami : "NOCDEX: Network-on-Chip Design Space Exploration Through Direct Execution and Options Selection Through Principal Component Analysis". International Symposium on Industrial Embedded Systems IES October 2006, France.
- [7] L. Tang and S. Kumar : "A Two Genetic Algorithm for Mapping Task Graphs to a Network-on-Chip Architecture," in Proc. of DSD'03, Antalya, Turkey, 2003, pp. 180–187.
- [8] K. Srinivasan and K. S. Chatha, and G. Konjevod : "Linear programming based techniques for synthesis of Network-on-Chip architectures," IEEE Transactions on VLSI Systems, vol. 14, no. 4, pp. 407–420, 2006.
- [9] J. Hu and R. Marculescu : "Exploiting the Routing Flexibility for Energy/Performance Aware Mapping of Regular NoC architecture," in Proc. of DAT'03, Munich, Germany, 2003, pp. 1068–1093.
- [10] M. Binesh Marvasti, T.H. Szymanski : "A power-area analysis of NoCs in FPGAs". IEEE International of SoC Conference, September 2012.
- [11] A. Banerjee, R. Mullins, S. Moore : "A Power and Energy Exploration of Network-on-Chip Architectures", Proceedings of NoCs, 2007, pp.163-172.[2]
- [12] N. Banerjee, P. Vellanki and K. S. Chatha : "A Power and Performance Model for Network-on-Chip Architectures", Proc. DATE , 2004, pp. 1250-1255
- [13] R.K. Jena : "System Level approach to NoC Design Space exploration". International Journal of Information and Electronics Engineering, vol 2, n° 2, march 2013.
- [14] J. Chan, S. Parameswaran : "NoCEE : Energy Macro-Model Extraction Methodology for Network-on-Chip Routers". IEEE IEEE/ACM International Conference on Computer Aided Design (ICCAD), 2005.
- [15] A. Penolazzi, A. Jantsch: "A High Level Power Model for the Nostrum NoC". 9<sup>th</sup> Euromicro conference on Digital System Design architectures, Methods and Tools, DSD, 2006, Dubrovnik.
- [16] K. Dannmann: "Performance analysis on Network-on-Chip Platforms based on Network Calculus". 1<sup>st</sup> International Workshop on Performance Analysis of Embedded and Networked Systems, October 2009, Sweden.
- [17] R. O. Duda : P. E. Hart, D. G. Stork, "Pattern Classification", 2nd Edition, ISBN: 978-0-471-05669-0, 680 pages, November 2000
- [18] K. Pearson: "Notes on the history of correlation". Biometrika, 1920, 13: 25–45.
- [19] P. Tan : "Introduction To Data Mining. Pearson Education", 2007.
- [20] F. Moraes, N. Calazans, A. Mello: "HERMES: an Infrastructure for Low Area Overhead Packet-switching Networks on Chip", Integration, the VLSI Journal, vol. 38, no. 1. pp 69–93. Oct. 2004.
- [21] C. A. Zeferino, A. A. Susin : "SoCIN: A Parametric and Scalable Network-on-Chip", Proc. 16th Symposium On Integrated Circuits and System Designs, pp 169-174. 2003.
- [22] C. Hilton, B. Nelson: "PNoC: a flexible circuit-switched NoC for FPGA-based Systems", in Field Programmable Logic, Aug. 2005. pp 24–26.

- [23] P. K. Sahu, A. Chattopadhyay : “A survey on application mapping strategies for Network-on-Chip design”. Journal of Systems architecture, Elsevier, 59, pp 60-76, 2013.
- [24] C. Hilton, B. Nelson: “PNoC: a flexible circuit-switched NoC for FPGA-based Systems”, in Field Programmable Logic, FPL, pp 24–26 August 2005.
- [25] R.K. Jean : “ABC Based multi-objectives approach for NoC design Space exploration”. International Journal of Information and Computation Technology, vol. 4, number 13, pp1263-1273, 2014.
- [26] V. Fresse, C. Combes, H. Belhassen : “Mathematical models applied to on-chip network on FPGA for resource estimation”. 17th IEEE International Conference on Computational Science and Engineering (CSE 2014), 19-21 December 2014, Chengdu, China.
- [27] A. Dorai, V. Fresse, C. Combes, E. Bourennane, A. Mtibaa : “A collision management structure for NoC deployment on multi-FPGA ” submitted in Elsevier Journal Computers and Electrical Engineering: On Chip Parallel and Network-based Systems, April 2015.
- [28] F. G. Villamon : “Design Space Exploration for Networks on Chip”. Phd report, University of Valencia, Spain, 2011.
- [29] E. Kang, E. Jackson, W. Schulte : “An approach for effective Design Space Exploration”. FOCS’10 Proceedings of the 16th Monterey conference on Foundations of computer software: modelling, development and verification of adaptive systems, pp33-54, 2010.
- [30] Xilinx: [www.xilinx.com](http://www.xilinx.com)



**Dr Virginie Fresse** received her PhD in Electrical Engineering at INSA Rennes, France, in 2001. Dr She worked as a post doc in the University of Strathclyde, Glasgow, from 2001 to 2003. She is currently associate professor at Jean Monnet University in Saint Etienne, France. Her research interests include NoC on FPGA, emulation platforms, design space exploration and real time systems for image processing applications.



**Dr Catherine Combes** is Assistant Professor at the University Institute for Technology of Saint-Etienne (France). She is member of the Machine Learning group at the Hubert Curien Laboratory. She received her PhD in computer science from Blaise Pascal University (Clermont-Ferrand - France) in 1994. Her main interests include machine learning, data mining, and discrete event simulation.



**Matthieu Payet** received the Master degree (with honors) in industrial computing and electronics from Grenoble University, Grenoble in 2011. He is currently working toward the Ph.D. degree at University of Saint-Etienne, France.

After completing his master degree, he joined Adacsys, an innovating startup, as Hardware Engineer, where he researched and developed FPGA based architectures for the acceleration, the test and the verification of circuits. His research interests include high-level synthesis CAD, reconfigurable computing architecture and Network-on-Chip architectures.



**Pr Frédéric Rousseau** received the Engineer degree in computer science and electrical engineering from the University of Grenoble in 1991 and a Ph.D. in computer science in 1997 from the University of Evry – France.

He has held an assistant professor position at the University of Grenoble since October 1999 and a professor position since 2007. He is researcher in TIMA lab. His research interest concerns Systems-on-Chip design and architecture, prototyping of hardware/software systems and high level synthesis for embedded systems

