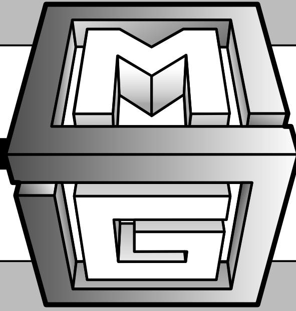

Université Libre de Bruxelles

CoDE - SMG

An MOO-based methodology for designing 3D-stacked integrated circuits

CoDE-SMG – Technical Report Series

N.A.V. Doan, D. Milojevic, F. Robert, Y. De Smet

CoDE-SMG – Technical Report Series

Technical Report No.
TR/SMG/2011-005

December 2011

CoDE-SMG – Technical Report Series
ISSN 2030-6296

Published by:
CoDE-SMG, CP 210/01
Université Libre de Bruxelles
Bvd du Triomphe
1050 Ixelles, Belgium

Technical report number TR/SMG/2011-005

The information provided is the sole responsibility of the authors and does not necessarily reflect the opinion of the members of CoDE-SMG. The authors take full responsability for any copyright breaches that may result from publication of this paper in the CoDE-SMG – Technical Report Series. CoDE-SMG is not responsible for any use that might be made of data appearing in this publication.

An MOO-based methodology for designing 3D-stacked integrated circuits

CoDE-SMG – Technical Report Series

Anh Vu Doan nguyen.anh.vu.doan@ulb.ac.be
Dragomir Milojevic dragomir.milojevic@ulb.ac.be
Frdric Robert frederic.robert@ulb.ac.be
Yves De Smet yves.de.smet@ulb.ac.be
CoDE-SMG, Université Libre de Bruxelles, Brussels, Belgium

August 2012

JOURNAL OF MULTI-CRITERIA DECISION ANALYSIS
J. Multi-Crit. Decis. Anal. **ISSUE**: PAGE1-PAGE2 (2012)
Published online XX Month 2012 in Wiley Online Library
(wileyonlinelibrary.com) DOI: 10.1002/mcda.XXX

A MOO-based methodology for designing 3D-stacked integrated circuits

N.A.V. Doan[*] [†], D. Milojevic[†], F. Robert[†] and Y. De Smet[*]
[*]CoDE-SMG - Brussels Faculty of Engineering
[†]BEAMS - Brussels Faculty of Engineering
Université Libre de Bruxelles, Av. F.D. Roosevelt 50, 1050 Brussels, Belgium

ABSTRACT
In the past decades, the microelectronic industry has been following the Moore's law to improve the performance of integrated circuits (IC). However, it will probably be impossible to follow this law in the future due to physical limitations appearing with the miniaturization of the transistors below a certain threshold. In order to overcome this problem, new technologies have emerged, and among them the 3D-Stacked Integrated Circuits (3D-SIC) have been proposed to keep the Moore's momentum alive. 3D-SICs can bring numerous advantages in the design of future ICs but at the cost of additional design complexity due to their highly combinatorial nature, and requiring the optimization of several conflicting criteria. In this paper, we present a first study of tools that can help the design of 3D-SICs, using mutiobjective optimization (MOO). Our study has targeted one of the main issues in the design of 3D-SICs: the floorplanning. This work shows that the use of MOO can provide relevant and objective analysis of the problem that may not be feasible with the current design methods. MOO can allow a quick design space exploration and this could improve the current design flows. Also, with its flexibility, MOO can cope with the multiple degrees of freedom of 3D-SICs, which enables more design possibilities. We believe that these promising results will help designers to overcome the main difficulties of designing 3D-SICs.

KEY WORDS: Multi-objective optimization, Microelectronic design, 3D-stacked integrated circuits

1 Introduction

In order to continuously improve the performance of integrated circuits (IC), technologists deploy enormous efforts to produce IC manufacturing process that is compelling to follow the well-known Moore's Law. This empirical law predicts a doubling of the transistors' integration each 18 months and therefore increasing logic capacity of the circuit per unit area. However, the development of future ICs will probably be affected by the physical limitations of the silicon — quantum effects due to miniaturization [1], and certainly the price involved with development of advanced technology nodes beyond $20nm$ [2].

Recently, new technologies have been emerged to overcome these limitations such as the carbon nanotubes [3], the nanowire transistors [4], the single-electron transistors [5], but also the 3D-Stacked Integrated Circuits (3D-SIC) proposed by the academic and industrial communities. The later has been often cited as the most prominent one [6].

Most of the current ICs are designed with electronic

Received: October 2011
Accepted: not yet

Copyright ©2011 John Wiley & Sons, Ltd.

components (i.e. transistors) that are planar (although multi-gate transistors, such as finFETs tends to extend in the 3rd dimension) interconnected using up to maximum of 12 (also planar) wiring (metal) layers per circuit. Those conventional ICs can thus be considered to be two-dimensional (2D)-ICs since the interconnections are predominantly made in a planar fashion [7, 8]. As a major evolution of 2D-ICs, 3D-SICs are designed with multiple traditional 2D-ICs (that are manufactured independently, using standard CMOS technology) that are assembled (stacked) vertically in 3D-tiers. Different 2D circuits communicate between tiers using vertical interconnections that need to connect front side of the chip and the backside, i.e. they need to traverse bulk silicon. These connections, known as – Through Silicon Vias (TSV), can be today manufactured with satisfactory geometrical properties, namely their diameter, pitch and height, allowing efficient integration of real-world systems [9, 10]. This is shown in Fig. 1, where 2 dies, oriented face down are connected. An active component (i.e. logic gate) of the T1 is connected to the T2 using a TSV, back side metallization layer (to enable TSV placement anywhere in the T1 die), and micro-bump on the top layer of the T2, that is then connected, through a series of metal layers of the T2, to the active component of the top tier (T2). Numerous advantages offered by using 3D-SICs have been pointed out in the literature the past years and will be discussed in Section 2.1.

Fast evolution of IC manufacturing technologies makes even the design of 2D-ICs a complex and tedious task with the growing number of design choices at the system level (e.g. number and type of functional units and memories, type and topology of the interconnection system, etc.) and physical level (respecting area/timing/power constraints). Using 3D-SICs introduces even more degrees of freedom: number of tiers, choices for manufacturing technology (e.g. full 3D integration, silicon interposer, face-to-face, back-to-face, etc.), 3D partitioning and placement strategies etc. These new degrees of freedom will contribute to the combinatorial explosion of already huge design spaces. Moreover, practice and 2D design experience cannot be fully exploited with 3D technology, since 3D-SICs change considerably the way ICs are implemented. The current design flows, which already showed their limits with conventional 2D-ICs, may thus need improvements to be able to deal with the increased complexity of emerging 3D-SICs [2,11].

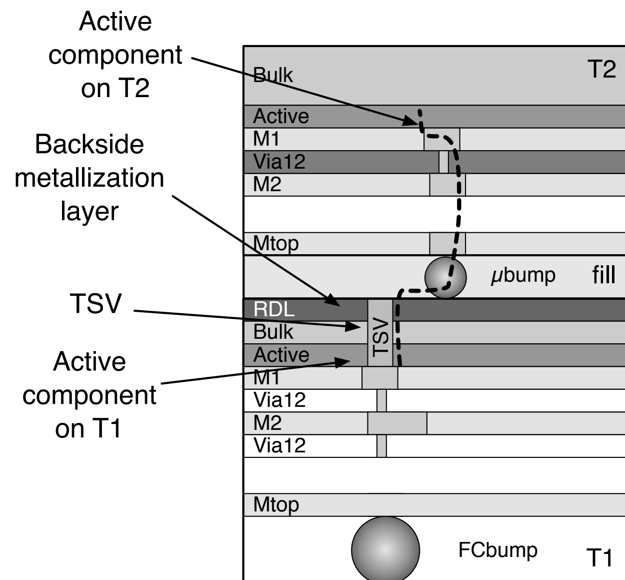

Figure 1: Illustration of the wiring properties of a 3D-SIC

One of the solutions to face this problem is to develop high-level tools which can quickly explore design spaces and give early and reasonably accurate performance estimations based on physical prototyping of the 3D circuits [2]. The tools from the operational research field have shown their abilities in solving similar problems in other fields, which also have a large solution space and applying metaheuristics have shown interesting results [12]. Due to the nature of the criteria (which will be explained later), there are few hopes to be able to apply exact methods. As a first approach, we thus propose to use metaheuristics which are commonly used to quickly explore huge solution spaces.

In addition, performance estimation/optimization and the selection of the most-suitable solutions usually implies to take several objectives in account (e.g. maximization of the performance, minimization of the cost, minimization of the package size, etc.).

Currently, the design tools can be considered to follow a uni-criterion paradigm. Indeed, they have sequential development steps and each criterion is opti-

Copyright ©2011 John Wiley & Sons, Ltd.

DOI: xxx/xxx

mized without considering the impact on other criteria. This can lead to several rollbacks in the design flow since the achievement of the requirements can be time consuming (typical design iterations are measured in weeks).

A more recent method consists in a multi-objective approach where all the criteria are simultaneously optimized. Designing 3D-SICs inherently implies a huge design space and numerous degrees of freedom and criteria, so the use of MOO seems to be suitable since it allows to quickly explore the design space while optimizing multiple criteria simultaneously. These criteria will be defined in Section 3.1.

The aim of this contribution is to perform a fast design space exploration and to assess/optimize the performance of 3D-SICs by using a metaheuristic. This paper is an extended version of the proceeding [13]. Previously, we have shown as proof of concept that MOO can be useful for IC design since it can provide information that would not be available with current tools. The main purpose of that work was to show that the use of multi-objective optimization could give relevant information to a designer. The analyses were done with only three criteria and were limited in term of degrees of freedom, and no analysis were done about the algorithm itself. The conclusion made in the proceeding mainly focused on a trade-off analysis and showed that a designer can have information on how (and in which direction) a 3D-SIC could be optimized.

In this paper, we extend the results in two ways:

- We show that the flexibility of MOO can easily allow a designer to consider more degrees of freedom and that provide even more interesting information about the design space. Previously, the possible degrees of freedom were limited to the geometrical positions of the components in a 3D-SIC. In this work, we include a new design possibility (the aspect ratio of a component, explained in Section 4) and show that this brings richer information that would not be available with current tools. In order to be able to visualize the improvements, this is still done with three criteria.

- We then analyze the algorithm we have used for five criteria, in order to show that, even with such a complex problem (discrete and continuous variables, linear and non-linear criteria), it is possible to evaluate interesting performance properties for the metaheuristics and the solution space (e.g. convergence, spread, density, etc.), and show that MOO in general can be helpful for the design of 3D-SICs.

In Section 2, we will present some related work in 3D-SIC integration and in multi-objective optimization. Then, in Section 3 we will define the problem and the criteria. This will be followed in Sections 4 and 5 by the design methodology and a case study as well as its implementation. Finally, the results of the study will be shown in Section 6 as well as the performance evaluation of the algorithm.

2 Related work

2.1 3D Integration

The benefits of using 3D-ICs are numerous and have been pointed out in the literature very often over the past few years [6]. The number of functions in the system can be extended beyond the capabilities of traditional 2D scaling because of the increased IC packaging density. 3D-SICs are expected to have much better performance.

First, by adding vertical dimension to the construction of the physical IC we can increase the IC packaging density. This means more components for the same circuit footprint, that is much higher functional complexity of the final circuit for the same packaging volume. Secondly, the 3D-SICs are expected to have much better computing/power dissipation ratio. The integration in the 3rd dimension allows the design of circuits with different parts being closer to each other, resulting in less and shorter wires [14]. The wiring properties of a 3D-SIC is illustrated in Fig. 1.

Lowering wire delays and allowing higher operating frequencies will result in increased bandwidths between nodes satisfying data hungry applications. Also, less and shorter wires mean lower total parasitic capacitance and inductance of the circuit, resulting in lower power dissipation and related buffering cost. Finally, the 3D-SICs will enable the design of really heterogeneous systems, embedding not only traditional digi-

Copyright ©2011 John Wiley & Sons, Ltd.

DOI: xxx/xxx

tal circuits such as processors and memories, but also analogue circuits such as sensors, antennas and power supplies [15]. The Fig. 2 shows a schematic view of such a heterogeneous circuit.

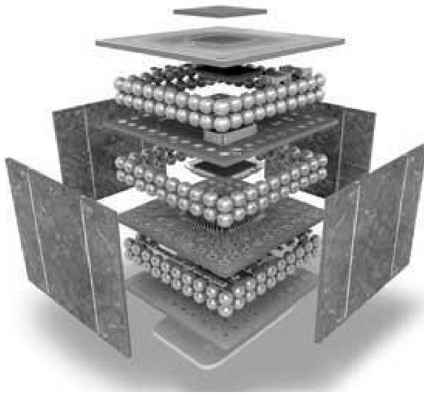

Figure 2: Artistic illustration of a 3D-SIC (proposed by IMEC) integrating sensors, antennas, power supplies, beside the digital circuits [16]

By using dedicated technologies for various functions, performance/power/cost metrics can be significantly improved. Besides classical system-level design challenges, 3D design thus involves additional degrees of freedom:

1. Partitioning of the design across multiple tiers
2. Selection of process technology per tier
3. Choice of 3D technology to interconnect the tiers

To understand the performance/power benefits of above design options, their physical design implications should be estimated. Today, no practical solution exists to assess physical design during the system exploration. The current methodology for 2D design is based only on data about the area, performance, power estimations, etc. of each component separately. This information is however insufficient for 3D-SIC since the whole physical design will define the performance and the cost of the circuit. The current alternative solution is to perform a full trial design, but it is not a practical solution since it would take too much time for a system exploration. For instance, it may take from several hours up to several days to evaluate the performance of one single design with the current tools and it is even worse if 3D-SICs are considered.

Currently different technologies for manufacturing of 3D-SICs have been proposed in the literature. Proposed methods have been used for the implementation of complete systems going far beyond simple proof-of-concept or feasibility demonstrators. One can mention the implementation of a processor and memory in a single 3D chip dedicated for video coding applications [17] and a processor with multiple levels of memory hierarchy dedicated for high-throughput server applications [18]. Finally, the first commercial 3D-SIC products have been already announced by IBM [19] and companies specialized in 3D semiconductor industry such as Tezzaron [20]. 3D Integration is taken into account in the roadmaps of almost all key players in the field of integrated circuit design and manufacturing.

As we can see, 3D-SIC design includes many options that have to be taken into account and the design space is thus huge. Indeed, if we take a really simplified example where we have 5 blocks of 2x2 mm to place in a surface of 10x10 mm with 25 fixed positions, there are 6 375 600 possible solutions. For a 3D-SIC with 3 tiers, there are 2 071 126 800 possibilities. If we consider a more complex design with 10 blocks, there are more than 3.10^{18} solutions.

The current design paradigm have already shown its limits with the increased complexity of such circuits. In addition, new problems appear (e.g. thermal dissipation) and it is more and more complicated to produce good designs with the current tools. We thus propose to improve the design flow in order to overcome these difficulties by using multi-objective optimization.

2.2 Multi-criteria paradigm and multi-objective optimization

In this section, we will briefly describe the multi-criteria paradigm and justify our choice to use multi-objective optimization.

As explained previously, the 3D technology can offer new perspectives but designing 3D-SICs includes two major characteristics: several criteria and a huge

Copyright ©2011 John Wiley & Sons, Ltd.

solution space. When facing optimization problems, two main methods exist: the uni-criterion approach and the multi-criteria approach. The problems encountered in the field of IC design — and in numerous other fields in the industry — often imply several conflicting criteria.

For instance, when designing ICs, a manufacturer will try to simultaneously maximize the performance while minimize the cost of the circuit. However, we can already guess that those two objectives are conflicting. Also, producing high-end ICs can be subject to more difficulties in thermal dissipation. In addition, a criterion based on ecological standards may have impacts on the cost and the performance of an IC. In order to deal with the several objectives of a problem, another approach consists in taking into account all the criteria simultaneously.

The first step is to identify the Pareto optimal frontier. Two approaches can be used to establish this set [21]:

- *Exact methods* which aims to compute the Pareto frontier directly [22, 23].
- *Approximate methods* which are based on metaheuristics to quickly explore the solution space and approach as best as possible the Pareto optimal frontier [12].

As explained, designing 3D-SICs includes a huge solution space to deal with in the optimization process. Since the solution is unknown and an exhaustive research would take a prohibitive time. Also, due to the nature of the criteria (discrete and continuous variables, linear and non-linear criteria) such as the thermal dissipation that can require finite elements, we have few hope to be able to develop an exact method. The definition of the criteria will be explained in Section 3 to illustrate that fact. For those reasons, we choose to use an approximate method.

We will thus use a multi-objective metaheuristic in order to establish an approximate Pareto front of our problem. Several algorithms have been developed and interested readers can find more information about them in reference books such as [12, 22, 24]. As popular metaheuristics, we can cite simulated annealing, genetic algorithms, ant colony optimization, scatter search, tabu search, etc.

Since our aim is to optimize sets of solutions, we choose to use a genetic algorithm (GA) for the multi-objective optimization. GAs belong to the class of evolutionary algorithm (EA) which is a population-based metaheuristic. In this work, we will use one of the most known approaches in EAs called NSGA-II (Non-dominated Sorting Genetic Algorithm-II) which is an extension of (the original) NSGA. More information about this algorithm can be found in [25]. We will give details about our implementation in Section 5.

Since it is a widely-used algorithm for prototyping [26], the choice of NSGA-II has been made as a first approach work, the main aim of this work being to show that multi-objective optimization in general can be helpful for the design of 3D-SICs. Also, NSGA-II allows both continuous and discrete variables [27], which is suitable to our problem.

2.3 On the use of MOO/MCDA in the field of IC design

Many high-level tools have been developed to optimize particular aspects of embedded systems. To the best of our knowledge, they are based on a uni-criterion paradigm (and thus a mono-objective optimization with integrated (meta)heuristics). Several works have been done to apply metaheuristics to 2D-IC design [28–32].

As stated in the introduction, a uni-criterion paradigm can show its limit while designing integrated circuits. Tools that include an overall design space exploration at the architeture level using multi-objective optimization have appeared in the past decade with the urgent need to deal with the growing complexity of designing ICs [28, 29, 33–35]. In particular, we can also cite the MILAN framework [36] in 2002, the PISA platform [37] in 2003, the SoC Architecture Explorer tool [38] in 2005 or the MULTICUBE project [39] in 2008.

Those works have been developed for 2D-ICs. 3D-SIC dedicated tools have only appeared recently [40–43]. To our knowledge, they use a uni-criterion approach or deal with a limited set of criteria while performing only trade-off analyses from a Pareto front. The goal of this paper is to show that a more multicriteria-oriented optimization could be more suit-

Copyright ©2011 John Wiley & Sons, Ltd.

able to take into account the many aspects of a design and provide more information.

3 Problem definition and proposed model

When designing a circuit, several degrees of freedom that characterize a system have to be taken into account. In the following, we will refer a system definition as a "scenario" and the components will be referred as "blocks". A "scenario" is thus a design which contains a number of "blocks". In order to meet the requirements for a circuit, a designer has to make choice at the physical level first:

- Targeted architecture, e.g. ASIC, FPGA
- Number of functional units
- Number of memories and their size
- The general layout
- Communication infrastructure, e.g. bus, Network-on-Chip
- ...

Since the 3D-SICs are based on conventional circuits, the options and degrees of freedom coming from 2D-ICs are still present:

- Process technology, e.g. 180 nm to 22 nm CMOS
- Memories technology, e.g. SRAM, DRAM, FLASH
- ...

In addition to those options and degrees of freedom coming from 2D-ICs, there are also numerous 3D-SIC's parameters:

- Number of tiers to use
- Place and route of the functional units between the tiers
- Technology to use per tier (heterogeneity)
- Interconnection and geometry between tiers
- 3D-SIC integration technology
- 3D-SIC assembly technology
- ...

The above mentioned parameters illustrate the numerous possibilities for designing a 3D-SIC. The main issue is therefore to choose the most efficient combination among all those options. This can thus be compared to a combinatorial optimization problem which makes the use of metaheuristics more than justified. In the next section, we will briefly define the different criteria that a designer can consider.

3.1 Model and criteria definition

Typically, the criteria that have to be optimized simultaneously can be the performance, the power consumption, the cost, the package size, the heat dissipation, etc. In this model, we will define six criteria which are among the most important parameters while designing a circuit [44]. Among the six presented criteria, the first five are implemented and the latter one is discussed:

1. *The interconnection global length*: this parameter can reflect the global performance of a system. The objective is to minimize it in order to have, for instance, a short delay and low power consumption. It will be calculated using the Manhattan distance:

$$d_{i,j} = \sqrt{|x_i - x_j| + |y_i - y_j|} \tag{1}$$

where (x_n, y_n) is the geometrical coordinates of the nth block. As a first approximation, the center point of each block will be selected as reference coordinate. Also, since it is more interesting to place close to each other two blocks that require a large bandwidth (BW) to communicate, we will balance the values as follows:

$$d'_{i,j} = \frac{d_{i,j}}{BW_{i,j}} \tag{2}$$

Copyright ©2011 John Wiley & Sons, Ltd.

DOI: xxx/xxx

So the global interconnection length D will be the sum of $d'_{i,j}$ for all communicating blocks:

$$D = \sum d'_{i,j} \quad (3)$$

2. *The cost*: an economical factor is obviously an important criteria for a design. This criteria has been estimated with the aid of an expert in 3D-SIC manufacturing. While a circuit can be more efficient with many layers, it will also be more expensive. This criteria has to be minimized. Due to the confidential nature of the cost of a 3D-SIC, we will consider a simplified model where the cost is proportional to the area and increasing exponentially with the number of tiers:

$$cost = a(tech).S + b(tech)^{layer\ number} \quad (4)$$

where $a(tech)$ and $b(tech)$ are coefficient depending on the technology assigned. Let us note that this criterion includes both discrete and continuous variables.

3. *The package volume*: this can be an important criteria when designing embedded circuits. The package volume is calculated as follows:

$$volume = largest\,layer\,size * stack\,thickness \quad (5)$$

A large appoximation of 200 µm will be made for the thickness of one tiers. Let us note that this criterion includes both discrete and continuous variables.

4. *The clock tree position*: in this model, we consider a synchronous system so the objective is to minimize the distance between each block and the clock tree in order to have a high frequency. We choose arbitrarily to approximate the reference point as a fixed point located at the upper left corner of the middle tier of the 3D-SIC.

5. *The thermal dissipation*: thermal dissipation is one of the major issue when designing 3D-SICs. It can be more appropriate to place two blocks underneath each other in successive tiers but a high heat dissipation may happen in intensive computational process. This criterion is a research topic on its own [45, 46]. Here we will use a simplified evaluation model with finite elements. This model will consider that the dissipated power, intra- or inter- tiers, is inversely proportional to the distance to the heat source:

$$P_{diss} = \sum_i \frac{1}{R_{th,i} r} \quad (6)$$

where r is the distance to the heat source and $R_{th,i}$ the thermal resistance depending on whether the dissipation is intra- or inter- tiers. This criteria is still on early development stage and we can generate thermal maps of a floorplan as shown in Fig. 3 but this is currently based on finite elements [47] which require quite a long computational time even for a simplified thermal model. This criterion in its current development stage is difficult to integrate to the exploration process, due to the computing time of finite elements. In the current work, we will simply compute the peak power of a circuit which can be done more quickly.

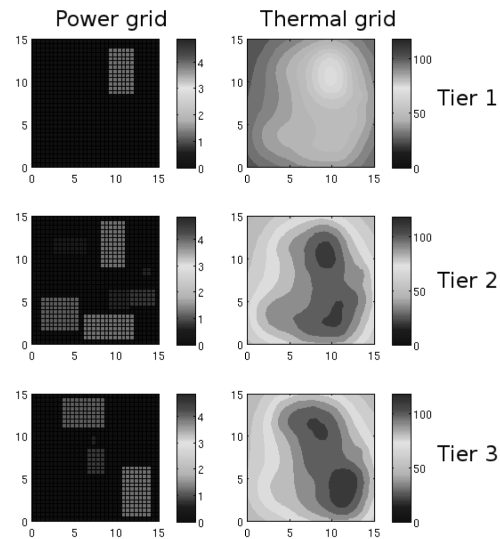

Figure 3: Power grid and thermal map of a floorplan (3 tiers)

6. *The power consumption*: the objective is to minimize the power consumption which can be a crucial criteria for embedded systems. Generally, the

Copyright ©2011 John Wiley & Sons, Ltd.

DOI: xxx/xxx

power consumption can be defined as the sum of a static (P_{stat}) (given data from the component datasheet) and a dynamic (P_{dyn}) consumption:

$$P = P_{stat} + P_{dyn} \quad (7)$$

$$P_{stat} = [given\ data] \quad (8)$$

$$P_{dyn} = \alpha.C.V_{dd}^2.f.[tech] \quad (9)$$

where α represents the toggle rate, C the capacitance, V_{dd} the voltage, f the frequency and $[tech]$ a rectification factor due to the technology assigned.

As already stressed in our previous work [13], we focused on the three first criteria in order to be able to have a visualization of the design space. This work was also limited in term of degrees of freedom. Here, we will go a step further. First, we will analyze what happens if we release a degree of freedom, in this case the aspect ratio of a component (explained in the next section). This will be done while considering the three same criteria, in order to keep a visualization of how the flexibility of MOO will improve the information. Then we will analyze our methodology with the five first criteria that have been presented. As stated, the sixth criterion is currently unused. Actually, it has been simulated and tested but we need data from the manufacturers which are not easily available.

4 Design methodology

In summary, the problem we are facing in this work is to place several blocks that have to be assigned in many tiers while considering the multiple conflicting criteria described previously. Now that our model of the problem and the involved criteria have been defined, we will present in this section a proposition of a new design methodology based on multi-objective optimization.

As explained previously, designing ICs implies numerous choices. At the moment, with this growing complexity, the current design flows can show their limits. For instance, most of the time, the designers will be likely to freeze a certain amount of choices on basis of their experience, and then begin the optimization process with the remaining parameters. This will therefore limit the exploration of the design space and good solutions may be ignored. Also, the fixed choices can be questionable since they are based on the designer's experience though they could also be based on more objective facts.

In order to enable an efficient design space exploration, we propose a design method in four steps based on MCDA which is illustrated in Fig. 4. The implementation will be briefly presented in the next section.

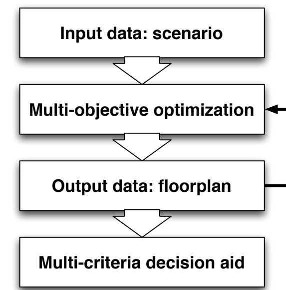

Figure 4: MCDA-based design methodology

For the problem we consider, the input data will contain the information about the scenario:

- Type and number of blocks: computational units, memories, etc.
- Size of the blocks: inherent to the block.
- Minimum aspect ratio: we consider a degree of freedom where a block can have its dimensions varying within an aspect ratio range. This means that a block does not have to be square, as shown in Fig. 5. This parameter can influence the delay in a block.
- Size variability of a block: we add this degree of freedom considering that the specified size of a block can be fixed by the designer but this fixed size can restrict the design space exploration. The variability of a block's size can have effect on the performance and the global footprint.

In addition, the bandwidth requirements are also needed as they will indicate which are the important interconnections and prevent two blocks that require a large bandwidth from being too far from each other.

Copyright ©2011 John Wiley & Sons, Ltd.

DOI: xxx/xxx

The available manufacturing technologies are as well useful to enable the design of heterogeneous systems.

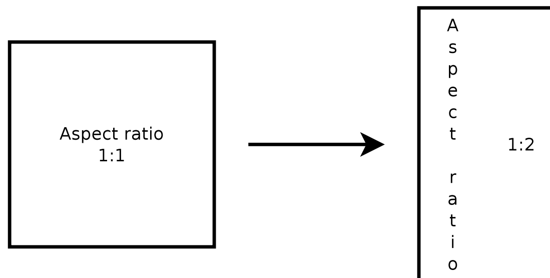

Figure 5: Example of aspect ratio degree of freedom

The combination of all the parameters described in the model are the possible alternatives for a 3D-SIC design and will provide output data after design space exploration. For a floorplanning problem the required output data are generally the geometrical layout of the circuit [44]:

- The geometrical coordinates for each block and the assigned layer.
- The size of each block (if it can vary from the specified size).
- The aspect ratio for each block.
- The technology assigned to each tier: this will reduce the size of each block. The size of a block will define the number of transistors inside using a given technology, for example 180 nm. For a constant number of transistors, if the block if manufactured with a smaller technology, let us say 45 nm, then its size will be divided by a $(180/45)^2$ factor, as shown in Fig. 6. Please note, that this factor is an approximation which is not always met with real physical design.

5 Case study and implementation

In this section, we will briefly explain the implementation of our design method and show some experimental results based on this first approach. In this work,

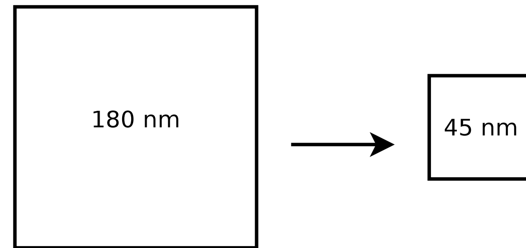

Figure 6: The use of different manufacturing technologies results in a size variation

we consider a case study based on the 3MF MPSoC platform developed at IMEC [48]. This case study has been implemented using Python (with NumPy) and all data were coded using matrices.

5.1 Description of the case study and modeling

The 3MF MPSoC is made of 13 blocks as shown on Fig. 7:

- 6 ADRES processors ([49])
- 2 data memories (L2D#)
- 2 instruction memories (L2Is#)
- 1 external memory interface (EMIF)
- 1 input/output processor (FIFO)
- 1 ARM processor (ARM)

Details about the area required for each component is given in Table 1 for a 90 nm technology. This table is also the input matrix required to specify the scenario.

The 3MF MPSoC can be configured for three use cases which have specific bandwidth requirements. This work is based on the ”data split scenario” configuration which possesses the communication specifications shown in Table 2. This information is implemented, as shown in Table 3, in an input matrix which is built by specifying the communication structure: the first column will contain the ID of the source block and

Copyright ©2011 John Wiley & Sons, Ltd.

DOI: xxx/xxx

Table 1: Scenario input matrix example

Component	ID	Size (90 nm)	Min aspect ratio	Size variability
ADRES 1~6	1~6	18.6 mm^2	0.5	0.1
FIFO	7	0.54 mm^2	0.5	0.1
L2D1-2	8-9	6.74 mm^2	0.5	0.1
L2Is1-1	10-11	6.62 mm^2	0.5	0.1
EMIF	12	0.66 mm^2	0.5	0.1
ARM	13	0.89 mm^2	0.5	0.1

ID: Component identification number

Table 2: ”Data split scenario” bandwidth requirements

Source	Target	Bandwidth (MB/s)
FIFO	EMIF	39.6
EMIF	ADRESi	6.6
L2D1	ADRESi	26.4
L2D2	L2D1	52.7
ADRESi	FIFO	1.2
ADRESi	L2D2	6.6
ADRESj	L2Is1	300
ADRESk	L2Is2	300

Index: $i, j, k \in \mathbb{N}^+$;
$1 \leq i \leq 6; 1 \leq j \leq 3; 4 \leq k \leq 6$

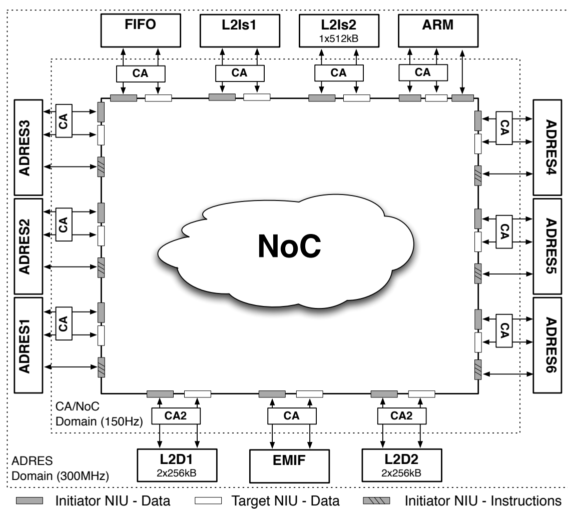

Figure 7: Architecture of the 3MF MPSoC platform [48]

each next pairs of columns will contain the ID of the target blocks and the bandwidth required.

The input data are thus shown in Tables 1 and 3. The available technologies are also needed to take advantage of the heterogeneity. An example matrix for this input data is given in Table 4. Since no information is given about the ARM unit bandwidth usage, we will simplify our problem and not include it in the implementation. We consider therefore 12 blocks to place.

In summary, the problem we consider is to place 12 blocks while taking into account several criteria. We will also consider a scenario where the blocks can be placed on 1 up to 5 tiers. The input data will be processed to generate floorplans. Those output data will be encoded using the matrix model following the example shown in Table 5. They will be generated through a multi-objective optimization.

As explained earlier, we will use a metaheuristic to approximate the Pareto optimal frontier. For that purpose, we choose to use NSGA-II [25]. The algorithm was run from a sample of 10 000 generated solutions from 1 up to 5 tiers. This size of random solutions is

Copyright ©2011 John Wiley & Sons, Ltd.

DOI: xxx/xxx

Table 3: Bandwidth input matrix

S	T	B	T	B	T	B	T	B	T	B	T	B
1	7	1.2	9	6.6	10	300	0	0	0	0	0	0
2	7	1.2	9	6.6	10	300	0	0	0	0	0	0
3	7	1.2	9	6.6	10	300	0	0	0	0	0	0
4	7	1.2	9	6.6	11	300	0	0	0	0	0	0
5	7	1.2	9	6.6	11	300	0	0	0	0	0	0
6	7	1.2	9	6.6	11	300	0	0	0	0	0	0
7	12	39.6	0	0	0	0	0	0	0	0	0	0
8	1	26.4	2	26.4	3	26.4	4	26.4	5	26.4	6	26.4
9	8	52.7	0	0	0	0	0	0	0	0	0	0
12	1	6.6	2	6.6	3	6.6	4	6.6	5	6.6	6	6.6

S: source block ID
(T, B): target block ID and required bandwidth

Table 4: Available technologies input matrix example

Technology (nm)				
90	60	45	32	22

Table 5: Output matrix template

ID	L	X	Y	S	AR	LX	LY	T
1	2	4.5	6	18.6	1	4.3128	4.3128	90
2	2	4	0.4	18.6	1	4.3128	4.3128	90
3	3	3.1	6.9	18.6	1	4.3128	4.3128	90
4	3	8.4	10.1	18.6	1	4.3128	4.3128	90
5	3	6.6	2.2	18.6	1	4.3128	4.3128	90
6	1	9	5.7	18.6	1	4.3128	4.3128	90
7	1	10	3.5	0.54	1	0.7348	0.7348	90
8	1	7.5	11	6.74	1	2.5962	2.5962	90
9	2	9	5	6.74	1	2.5962	2.5962	90
10	1	4.5	8	6.62	1	2.5729	2.5729	90
11	2	8.6	0.4	6.62	1	2.5729	2.5729	90
12	3	8.3	7.4	0.66	1	0.8124	0.8124	90

ID: component identification number; L: assigned layer;
(X, Y): geometrical coordinate; S: size (mm^2); AR: aspect ratio;
(LX, LY): length in X and Y axis; T: assigned technology for the layer

chosen arbitrarily since it is actually quite difficult to estimate the size of solution space, due to the heterogeneous nature of the criteria. As stated in Section 2, even for a really simplified problem, the solution space is huge (more than 10^{18}) so, taking 10 000, 100 000 or 1 000 000 randomly-generated solutions does not really imply any difference. Also, taking too few solutions (e.g. 100) is not interesting since we have empirically observed that our algorithm will take a longer time to begin to converge. 10 000 randomly-generated solutions seems to us a good compromise of time and workable solutions.

In the following section, we will present some details about the implementation of the NSGA-II algorithm.

5.2 Implementation of the exploration algorithm: NSGA-II

As shown in Table 5, we choose to encode our data in real or integer values, so that they can be used directly by design tools:

- The component identification number (ID) is a fixed integer value linked to the component.
- The assigned layer (L) is a discrete value ranging from 1 to 5 in the case study.
- The geometrical coordinates (X,Y) are real values that depends on the dimension of the circuits and

Copyright ©2011 John Wiley & Sons, Ltd.

DOI: xxx/xxx

the aspect ratio of a block, so that the component cannot be placed outside the chip.

- The size (S) is a fixed real value linked to the component.
- The aspect ratio (AR) is a real value ranging from AR_{min} to $1/AR_{min}$ where AR_{min} is given as a specification as explained in Section 4.
- The length in X and Y axis (LX, LY) are real values computed from the size and the aspect ratio.
- The assigned technology per layer is a discrete value taking one of the specified technology (see Table 4).

This matrix will be our full chromosome for the NSGA-II algorithm.

We implemented our design space exploration following the steps of the NSGA-II which can be summarized by the diagram shown in Fig. 8.

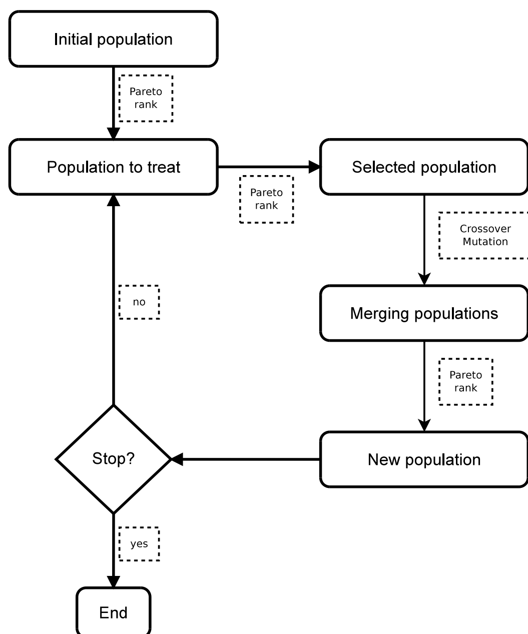

Figure 8: General NSGA-II steps

Initialization (the initial population)

We will work with a minimum size of population, namely 50, which is a common value in GAs [50]. The initial population will be a set of at least 50 solutions with the best Pareto ranks from a randomly-generated set of 10 000 solutions. The generated set places the blocks randomly (using a uniform distribution) and does not allow overlapping between the blocks. We could of course use a greedy algorithm as well as a more advanced method such as GRASP (Greedy Randomized Adaptive Search Procedure) [51]. This can be done as future work for comparison purposes.

Of course, having at least 50 Pareto solutions does not always happen. Actually, the selection is based on the Pareto rank so it does not include only the Pareto solutions (rank 1), but also the solutions with lower ranks:

1. First, the Pareto optimal solutions are taken.
2. If the size of the set is lower than 50, then all the solutions from the next Pareto rank are taken.
3. Step 2 is repeated until at least 50 solutions are in the set.

This will also be the rules for each generation but for at least 100 solutions.

The lower rank solutions are obtained as follows:

- To find the solutions of rank 2, the solutions of rank 1 are removed and we compute the Pareto front of the remaining set.
- To find the solutions of rank 3, the solutions of rank 2 are then removed and we compute the Pareto front of the remaining set.
- Repeat iteratively for the next ranks.

Selection/crossover

For the selection step, two solutions will be allowed to make a crossover depending on a roulette wheel where the probability is proportional to the normalized Euclidean distance between the solutions ordered by their

Copyright ©2011 John Wiley & Sons, Ltd.

Pareto rank in the objective space. The normalization is done as follows :

$$\frac{g_i(A_j)}{\max_A g_i(A)} \quad (10)$$

where $g_i(A_j)$ is the evaluation of the solution A_j on the criterion i.

If two solutions are close to each other, they will have more chance to reproduce than if they are distant. This is to ensure the intensification properties of our algorithm. Since we choose to have a crossover probability proportional to the Euclidean distance between the solutions, we will have to specify a lower and an upper bound for the probabilities. The lower bound is set for the solutions which are the furthest to each other while the upper bound is for those which are the closest. In between, the probability will vary linearly.

These values will be fixed as $[P_{c,min} = 0.6; P_{c,max} = 1.0]$ since these seem to be common values [50].

Crossover

Let us now see how does the crossover occur. First, let us remark that it does not have limitations for the exploration process since the information contained in the matrix spans the whole circuit.

Second, we have to analyze how the chromosome is coded in order to see how we will apply the crossover step. For instance, let us choose the Layer (L) column as indicator for the crossover. If we order the matrix in Table 5 following this column, we will have the Table 6 and the Table 7 for another solution that we will use for the crossover.

Now, without loss of generality, let us suppose that the crossover happens (randomly) on line 7. One of the child will be the Table 8 and we see that the original scenario is not preserved since the first column (in bold) contains the same ID several times.

We observe that the only possible indicator for the crossover step is the ID column. Indeed if we order the two parents following the ID column, we have the Tables 9 and 10. If we still consider that the crossover occurs on line 7, we can have the child shown in Table 11. We see that there is no inconsistency since the scenario is still respected.

Table 6: First parent, ordered by L column; the line specifies the crossover cut

ID	L	X	Y	S	AR	LX	LY	T
6	1	9	5.7	18.6	1	4.3128	4.3128	90
7	1	10	3.5	0.54	1	0.7348	0.7348	90
8	1	7.5	11	6.74	1	2.5962	2.5962	90
10	1	4.5	8	6.62	1	2.5729	2.5729	90
1	2	4.5	6	18.6	1	4.3128	4.3128	90
2	2	4	0.4	18.6	1	4.3128	4.3128	90
9	2	9	5	6.74	1	2.5962	2.5962	90
11	2	8.6	0.4	6.62	1	2.5729	2.5729	90
3	3	3.1	6.9	18.6	1	4.3128	4.3128	90
4	3	8.4	10.1	18.6	1	4.3128	4.3128	90
5	3	6.6	2.2	18.6	1	4.3128	4.3128	90
12	3	8.3	7.4	0.66	1	0.8124	0.8124	90

ID: component identification number; L: assigned layer;
(X, Y): geometrical coordinate; S: size (mm^2); AR: aspect ratio;
(LX, LY): length in X and Y axis; T: assigned technology for the layer

Table 7: Second parent, ordered by L column; the line specifies the crossover cut

ID	L	X	Y	S	AR	LX	LY	T
4	1	1.6	8.5	18.6	1	4.3128	4.3128	90
5	1	1.2	1.3	18.6	1	4.3128	4.3128	90
6	1	0.6	4.7	18.6	1	4.3128	4.3128	90
3	2	5.9	4	18.6	1	4.3128	4.3128	90
9	2	5.4	8	6.74	1	2.5962	2.5962	90
10	2	8.5	8.1	6.62	1	2.5729	2.5729	90
11	2	2.8	4.6	6.62	1	2.5729	2.5729	90
1	3	7	6.3	18.6	1	4.3128	4.3128	90
2	3	7.4	9.8	18.6	1	4.3128	4.3128	90
7	3	5.6	5.5	0.54	1	0.7348	0.7348	90
8	3	2.8	5.5	6.74	1	2.5962	2.5962	90
12	3	5.7	7.5	0.66	1	0.8124	0.8124	90

ID: component identification number; L: assigned layer;
(X, Y): geometrical coordinate; S: size (mm^2); AR: aspect ratio;
(LX, LY): length in X and Y axis; T: assigned technology for the layer

Copyright ©2011 John Wiley & Sons, Ltd.

Table 8: Possible child, ordered by L column

ID	L	X	Y	S	AR	LX	LY	T
6	1	9	5.7	18.6	1	4.3128	4.3128	90
7	1	10	3.5	0.54	1	0.7348	0.7348	90
8	1	7.5	11	6.74	1	2.5962	2.5962	90
10	1	4.5	8	6.62	1	2.5729	2.5729	90
1	2	4.5	6	18.6	1	4.3128	4.3128	90
2	2	4	0.4	18.6	1	4.3128	4.3128	90
9	2	9	5	6.74	1	2.5962	2.5962	90
1	3	7	6.3	18.6	1	4.3128	4.3128	90
2	3	7.4	9.8	18.6	1	4.3128	4.3128	90
7	3	5.6	5.5	0.54	1	0.7348	0.7348	90
8	3	2.8	5.5	6.74	1	2.5962	2.5962	90
12	3	5.7	7.5	0.66	1	0.8124	0.8124	90

ID: component identification number; L: assigned layer;
(X, Y): geometrical coordinate; S: size (mm^2); AR: aspect ratio;
(LX, LY): length in X and Y axis; T: assigned technology for the layer

Table 10: Second parent, ordered by ID column; the line specifies the crossover cut

ID	L	X	Y	S	AR	LX	LY	T
1	3	7	6.3	18.6	1	4.3128	4.3128	90
2	3	7.4	9.8	18.6	1	4.3128	4.3128	90
3	2	5.9	4	18.6	1	4.3128	4.3128	90
4	1	1.6	8.5	18.6	1	4.3128	4.3128	90
5	1	1.2	1.3	18.6	1	4.3128	4.3128	90
6	1	0.6	4.7	18.6	1	4.3128	4.3128	90
7	3	5.6	5.5	0.54	1	0.7348	0.73485	90
8	3	2.8	5.5	6.74	1	2.5962	2.5962	90
9	2	5.4	8	6.74	1	2.5962	2.5962	90
10	2	8.5	8.1	6.62	1	2.5729	2.5729	90
11	2	2.8	4.6	6.62	1	2.5729	2.5729	90
12	3	5.7	7.5	0.66	1	0.8124	0.8124	90

ID: component identification number; L: assigned layer;
(X, Y): geometrical coordinate; S: size (mm^2); AR: aspect ratio;
(LX, LY): length in X and Y axis; T: assigned technology for the layer

Table 9: First parent, ordered by ID column; the line specifies the crossover cut

ID	L	X	Y	S	AR	LX	LY	T
1	2	4.5	6	18.6	1	4.3128	4.3128	90
2	2	4	0.4	18.6	1	4.3128	4.3128	90
3	3	3.1	6.9	18.6	1	4.3128	4.3128	90
4	3	8.4	10.1	18.6	1	4.3128	4.3128	90
5	3	6.6	2.2	18.6	1	4.3128	4.3128	90
6	1	9	5.7	18.6	1	4.3128	4.3128	90
7	1	10	3.5	0.54	1	0.7348	0.7348	90
8	1	7.5	11	6.74	1	2.5962	2.5962	90
9	2	9	5	6.74	1	2.5962	2.5962	90
10	1	4.5	8	6.62	1	2.5729	2.5729	90
11	2	8.6	0.4	6.62	1	2.5729	2.5729	90
12	3	8.3	7.4	0.66	1	0.8124	0.8124	90

ID: component identification number; L: assigned layer;
(X, Y): geometrical coordinate; S: size (mm^2); AR: aspect ratio;
(LX, LY): length in X and Y axis; T: assigned technology for the layer

Table 11: Possible child, ordered by ID column

ID	L	X	Y	S	AR	LX	LY	T
1	2	4.5	6	18.6	1	4.3128	4.3128	90
2	2	4	0.4	18.6	1	4.3128	4.3128	90
3	3	3.1	6.9	18.6	1	4.3128	4.3128	90
4	3	8.4	10.1	18.6	1	4.3128	4.3128	90
5	3	6.6	2.2	18.6	1	4.3128	4.3128	90
6	1	9	5.7	18.6	1	4.3128	4.3128	90
7	1	10	3.5	0.54	1	0.7348	0.7348	90
8	3	2.8	5.5	6.74	1	2.5962	2.5962	90
9	2	5.4	8	6.74	1	2.5962	2.5962	90
10	2	8.5	8.1	6.62	1	2.5729	2.5729	90
11	2	2.8	4.6	6.62	1	2.5729	2.5729	90
12	3	5.7	7.5	0.66	1	0.8124	0.8124	90

ID: component identification number; L: assigned layer;
(X, Y): geometrical coordinate; S: size (mm^2); AR: aspect ratio;
(LX, LY): length in X and Y axis; T: assigned technology for the layer

Copyright ©2011 John Wiley & Sons, Ltd.

Mutation

A mutation cannot happen anywhere in the matrix. Indeed, if we take the conclusion about the choice of the crossover row indicator, all the elements except the ID column can mutate.

The mutation used is a random uniform distribution $U([a, b])$, where $[a, b]$ is the interval of values that the mutating element can take. For the discrete values, we use equidistributed discrete probabilities. The mutation probability of a child will be set as $P_m = 0.3$. Empirical observations have shown that smaller mutation probability can easily lead to a local optimum. This can be explained by the fact that we choose that only a single element of a line can mutate instead of the whole line. If a child is allowed to mutate, then one randomly-chosen value of the whole matrix will mutate within the range of values it is allowed to take.

A Gaussian mutation is also a common operator but it has not been chosen since it will produce a solution which is not far from the original one. This is not really interesting to have similar solutions when exploring the design space for integrated circuits. Of course, a large standard deviation value can be chosen but this will be likely to produce solution which are out of the feasible bounds.

Consistency test

Of course, infeasible solutions may appear after the crossover/mutation step, since these operations are made with randomness. In order to verify that, we do a test on each new solution to check if there is overlapping between the blocks. Currently, the solutions which are infeasible will be discarded. Of course, it is possible to apply some repair mechanism but it is to be investigated as future work even if we currently have feasible solutions.

Termination (stop conditions)

Three stop conditions have been implemented and are based on what is commonly used:

- Maximum number of iterations, set to 100.
- Maximum elapsed time, set to 1 hour.
- Maximum number of iterations with an unchanged population, set to 10.

The maximum elapsed time has been chosen arbitrarily for quick testing purposes. Having a simulation time of a few hours would not be a problem either. Indeed, in practice, the optimization of one single architecture can take from several hours to several days with the current design tools.

6 Results and their use for a designer

In [13] we have shown that the analysis provided by a multi-objective optimization can give relevant information to a designer that would not be available with classical design tools. The optimization was done for three criteria (so that we can visualize the design space) and the main results are given in Fig. 9 (3D plot) and Fig. 10 (interconnection length-cost projection). In summary, two conclusions can be drawn from that figure:

- The [10; 20] range values for the IL criteria: a small enhancement of the IL value leads to a large increase of the cost so the interest for a design with more than 4 tiers seems low.
- The [260; 280] range values for the cost criteria: a small increase of the price can give a large enhancement of the performance. A designer might consider accepting a slightly higher price for a sensitively better performance, knowing that this information can be quantified with an accurate model. Indeed, with the estimate model that we propose, a small 10% increase of the cost can decrease the IL by 60%.

These results from [13] did not take into account the degree of freedom of aspect ratio. If we go further by allowing varying aspect ratios, we can have the Pareto front shown in Fig. 11. This figure shows the Pareto front from Fig. 10 (without aspect ratio, symbol: ·) alongside with a new Pareto front (with aspect ratio, symbol: +).

Copyright ©2011 John Wiley & Sons, Ltd.

DOI: xxx/xxx

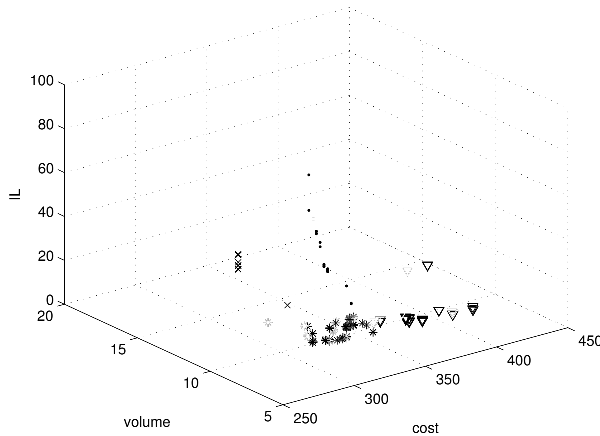

Figure 9: 3D view (interconnection length-volume-cost) of the Pareto frontier

$\cdot$: 1 tier; $\times$: 2 tiers; + : 3 tiers; * : 4 tiers; $\triangledown$: 5 tiers

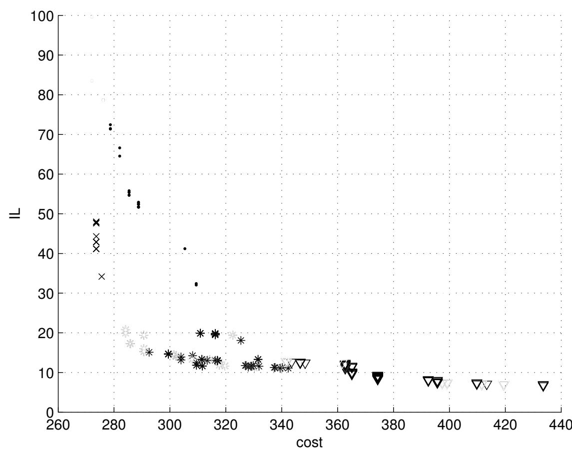

Figure 10: IL-cost projection view of the Pareto frontier

$\cdot$: 1 tier; $\times$: 2 tiers; + : 3 tiers; * : 4 tiers; $\triangledown$: 5 tiers

As expected, the Pareto front given when considering varying aspect ratios is globally better. Furthermore, by comparing the two graphs, we can see an interesting area where the two frontiers begin to merge at the cost value 350. This means that, in that area, it is not necessary to take the aspect ratio into account. Once again, these kind of information can be important in the design of an IC and yet they would not be available with the current design flows since only a small number of possibilities are explored. Indeed, due to the sequential nature of the current design flows, such degrees of freedom are not even tried since they dramatically increase the duration of each optimization loop.

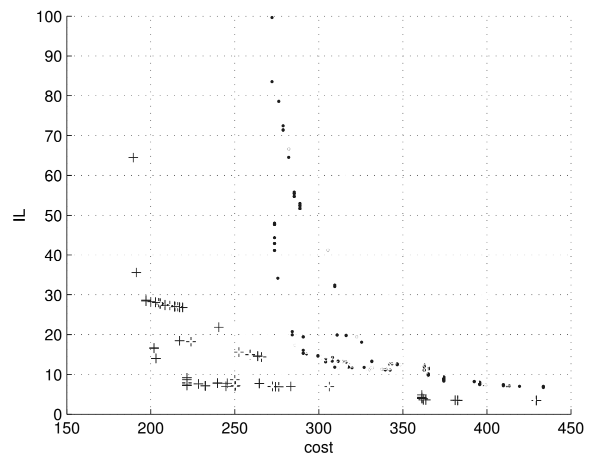

Figure 11: IL-cost projection view of the Pareto frontier (with and without aspect ratio)

$\cdot$: Pareto front without aspect ratio; + : Pareto front with aspect ratio

The results of this research have thus shown interesting results that can be relevant for a designer. First, using a multi-objective optimization methodology does not only consider all the criteria at the same time but also proceed to an extensive design space exploration which is rarely done with current tools. Second, the qualitative results shown here can give relevant information to the designer and they can be quantified with a more accurate model. Third, the flexibility of MOO allows to easily consider new degrees of freedom without having to change the paradigm. Finally, this methodology and the associated algorithms has shown positive indicators of convergence and robustness as it will be shown in the next section.

Copyright ©2011 John Wiley & Sons, Ltd.

7 Performance evaluation and Pareto front structure

Let us now take a deeper look at the results of the multi-objective optimization. We will thus analyze the properties of the design space in order to have the view over the convergence and the robustness of our methodology. We will use the classical indicators such as the contribution, the spread, the convexity which are presented in [12]. These results have been obtained with an Intel Core 2 Duo 2.0 GHz, 1.5 GB DDR2 SDRAM for 5 independent experiments.

7.1 Convergence-based indicators

> "The convergence metrics evaluate the effectiveness of the solutions in terms of the closeness to the optimal Pareto front." [12]

Contribution

As stated, the contribution is a convergence-based binary indicator. The contribution of an approximation PO_1 relatively to another approximation PO_2 is the ratio of non-dominated solutions produced by PO_1 in PO^*, which is the set of Pareto solutions of $PO_1 \cup PO_2$:

$$Cont(PO_1/PO_2) = \frac{\frac{\|PO\|}{2} + \|W_1\| + \|N_1\|}{\|PO^*\|} \quad (11)$$

where PO is the set of solutions in $PO_1 \cap PO_2$, W_1 the set of solutions in PO_1 that dominate some solutions of PO_2 and N_1 the set of non-comparable solutions of PO_1. This value has to be greater than 0.5 to indicate that PO_1 is better than PO_2 in terms of convergence to the Pareto front.

The Table 12 and the Fig. 12 show the evolution of the contribution indicator of the iterations of an experiment (the same observations apply for the other runs). We see that for the first iterations, $Cont(PO_i/PO_{i-1})$ is greater than 0.5, which means that the algorithm does indeed improve the solutions, then for the last iterations, the indicators are lower than 0.5 which means that there is a convergence.

Iteration	$Cont(PO_i/PO_{i-1})$
1	0.8080
2	0.8549
3	0.8916
4	0.8734
5	0.8451
...	...
48	0.4566
49	0.2741
50	0.3692

Table 12: Evolution of the contribution indicator

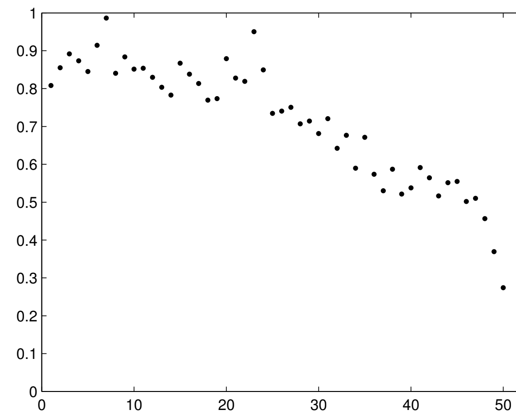

Figure 12: Evolution of the contribution indicator

Binary ϵ-indicator

A binary ϵ-indicator gives the quality of a solution front in comparison with a another set, with regards to all objectives. Consider a minimization problem with n positive objectives. An objective vector $f^1 = (z_1^1, z_2^1, \ldots, z_n^1)$ is said to ϵ-dominate another objective vector $f^2 = (z_1^2, z_2^2, \ldots, z_n^2)$ if $\forall 1 \leq i \leq n$: $z_i^1 \leq \epsilon \cdot z_i^2$, for a given $\epsilon > 0$. A binary ϵ-indicator $I_\epsilon(A, B)$ gives the factor ϵ such that for any solution in B there is at least one solution in A that is not worse by a factor of ϵ in all objectives. $I_\epsilon(A, B)$ can be calculated as follows [52]:

$$I_\epsilon(A, B) = \max_{z^2 \in B} \min_{z^1 \in A} \max_{1 \leq i \leq n} \frac{z_i^1}{z_i^2} \quad (12)$$

Copyright ©2011 John Wiley & Sons, Ltd.

A comparison of the binary ϵ-indicators for each experiment is given in Table 13. We can see that $I_\epsilon(A, B) > 1$ and $I_\epsilon(B, A) > 1$ which indicates that neither A weakly dominates B nor B weakly dominates S. This means that the generated front is consistent from one experiment to another.

Also, in Table 14 are given the ϵ-indicator between iterations of an experiment (the same observations apply for the other runs). We can see that in the first iterations, the front is always improved ($I_\epsilon(A_i, A_{i-1}) > 1$ and $I_\epsilon(A_{i-1}, A_i) \leq 1$) while in the last iterations, it begins to converge ($I_\epsilon(A_i, A_{i-1}) > 1$ and $I_\epsilon(A_{i-1}, A_i) > 1$).

$I_\epsilon(A, B)$	Run 1	Run 2	Run 3	Run 4	Run 5
Run 1	1	1.7234	1.3772	1.8880	1.2152
Run 2	1.5305	1	1.3548	1.8333	1.3488
Run 3	1.4409	1.9285	1	1.8305	1.2952
Run 4	1.3709	1.4048	1.2654	1	1.2310
Run 5	1.4554	1.8533	1.4400	1.7868	1

Table 13: Comparison of the binary ϵ-indicators for each experiment

Iteration	$I_\epsilon(A_i, A_{i-1})$	$I_\epsilon(A_{i-1}, A_i)$
1	1.6674	1.1053
2	1.7223	1
3	2.4439	1
4	1.7477	1
5	2.0577	1
...	...	...
48	1.8788	1.4916
49	1.5344	1.8065
50	1.9862	1.6609

Table 14: Comparison of the binary ϵ-indicators between iterations of the same experiment

7.2 Diversity-based indicator

> "Diversity indicators measure the uniformity of distribution of the obtained solutions in terms of dispersion and extension. In general, the diversity is researched in the objective space." [12]

Spread

The spread indicator I_s combines the distribution and cardinality to measure the dispersion of the approximated set A:

$$I_s = \frac{\sum_{u \in A} |\{u' \in A : \|F(u) - F(u')\| > \sigma\}|}{|A| - 1} \tag{13}$$

where $F(u)$ is a fitness function and $\sigma > 0$ a neighborhood parameter. The closer is the measure to 1, the better is the spread of the approximated set A.

The Fig. 13 shows the results of the spread indicator I_s function of the neighborhood indicator σ (all the 5 experiments share the same graph shape). We see that the Pareto front is well spread ($I_s \geq 0.9$) for $\sigma < 0.39$ in average for the 5 runs (normalized values), so we can consider that the algorithm produces a well-spread approximation of the Pareto front.

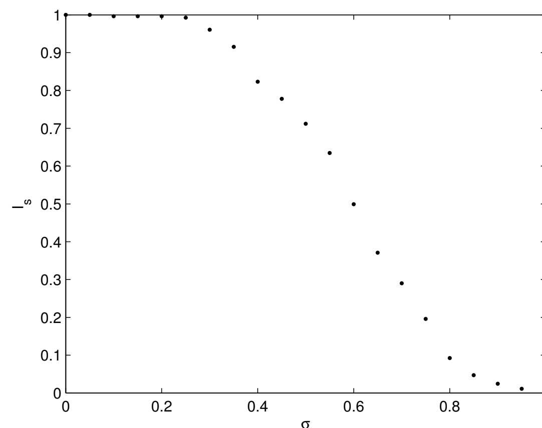

Figure 13: Spread indicator I_s function of the neighborhood parameter σ

7.3 Density of the Pareto front - gaps in the frontier

Another indicator of the Pareto front structure is its density. Here we will measure the density by find-

Copyright ©2011 John Wiley & Sons, Ltd.

DOI: xxx/xxx

ing gaps in the frontier. This will be done by counting the number of solutions in the neighborhood of another solution. Since the extreme distance between two solutions is 450.364 in average for the 5 runs (non-normalized), we consider that an acceptable neighborhood is twice the distance between two solutions if all the solutions were equidistant. We have thus a neighborhood of about 2. This test has shown that there was always at least one solution near another one, even for a neighborhood of 1, meaning that the algorithm can produce a sufficiently dense frontier.

7.4 Hybrid indicators

Some quality indicators combine convergence and diversity measure.

Unary hypervolume indicator

Since we already used a binary indicator (epsilon), we will use the hypervolume indicator I_H in its unary form. I_H, associated with an approximation set A is given by the volume of the space portion that is weakly dominated by the set A [12]. The result for each experiment is given in Table 15. As we can see, the values are rather consistent from one run to another.

Run	I_H
1	0.084617
2	0.084566
3	0.094522
4	0.102438
5	0.101421

Table 15: Hypervolume for each experiment

7.5 General properties of the algorithm

With the several indicators that we have analyzed, we can conclude that the algorithm we used can show good properties of convergence, spread and density. The methodology can thus be considered as robust even if the problem is not homogeneous (heterogeneous nature of the criteria) and this proves that a multi-criteria paradigm can be suitable for the design of 3D-SICs.

Also, analyses have been performed to determine the shape of the Pareto front. Globally, the Pareto front is not convex, as one may expect since the heterogeneous nature of the criteria. This is probably due to some correlation between the criteria but this is still to be investigated more deeply.

8 Conclusion and future works

This work mainly aimed at showing a preliminary overview of a MOO-based method for designing 3D-SICs. Given the growing complexity of designing conventional 2D-ICs, current design flows can already exhibit their limits and we have shown that with the complexity of 3D-SICs, making right system level and physical design choices has become more difficult.

We therefore propose a new methodology based on the use of a multi-criteria paradigm which allow a designer to dispose of objective information about the design space, in particular the establishment of the Pareto frontier and the decision aid. We have proposed a model of 3D-SIC in order to apply a multi-objective optimization by using an NSGA-II algorithm for which we have analyzed the properties. The results obtained can give relevant information over the design space that would not be available with the current tools and some degrees of freedom are not even tried since the time it would take with the current design flows.

Also, it seems that applying MOO to explore the design space of ICs at the architecture level is quite innovative for the field, especially for 3D-SIC design and to our best knowledge, this is the first approach that adopts a multi-criteria paradigm at the architecture/system level design. The main aim of this work was thus to show that MOO can be helpful for the design of 3D-SIC.

Of course, other algorithms than NSGA-II could be used and might produce better results. Some improvement can be made on the exploration process. As stated earlier, a GRASP algorithm can be developed as comparison purposes.

This methodology has also proved to be robust even if the problem contains criteria of heterogeneous na-

Copyright ©2011 John Wiley & Sons, Ltd.

DOI: xxx/xxx

ture. With these promising results, we believe that a multi-criteria paradigm can be applied to the design of 3D-SICs in order to produce them more efficiently than we can be done with the current tools.

Globally, our methodology includes two steps:

- Fast design space exploration and performance estimation/optimization using metaheuristics
- Ex-post exploration for multi-criteria decision analysis and decision over the most-suitable solutions

This work has focused on the first step. As future works, the second step is still to be done. Indeed, now that we can generate a Pareto front, we have to fully analyze how the information provided by the frontier can be used by a designer with an ex-post exploration. As for the criteria, they will be improved continuously in order to be able to have more accurate estimation.

On another side, we have done this work for a real case study but it will also be interesting to see how this methodology can be used on other structures, especially when there is a scaling effect on the number of components. Finally, this research will be validated by integration to existing design flow such as PathFinding [44]. This will allow us to work with a full flow, from architecture/system-level to physical design and to compare this methodology with the current classical one.

Acknowledgement

This work has been supported by the ARC Project META-X funded by *La Communauté française de Belgique*

References

[1] V. Zhirnov, I. Cavin, R.K., J. Hutchby, and G. Bourianoff, "Limits to binary logic switch scaling - a gedanken model," *Proceedings of the IEEE*, vol. 91, no. 11, pp. 1934 – 1939, nov 2003.

[2] D. Milojevic, R. Varadarajan, D. Seynhaeve, and P. Marchal, *PathFinding and TechTuning in Three Dimensional System Integration*, A. Papanikolaou, D. Soudris, and R. Radojcic, Eds. Springer, Nov 2011. [Online]. Available: http://www.springer.com/architecture+%26+design/architecture/book/978-1-4419-0961-9

[3] S. Tans, A. Verschueren, and C. Dekker, "Room-temperature transistor based on a single carbon nanotube," *Nature*, vol. 393, no. 6680, pp. 49–52, 1998.

[4] Y. Cui, Z. Zhong, D. Wang, W. U. Wang, and C. M. Lieber, "High performance silicon nanowire field effect transistors," *Nano Letters*, vol. 3, no. 2, pp. 149–152, 2003. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/nl0258751

[5] D. Goldhaber-Gordon, H. Shtrikman, D. Mahalu, D. Abusch-Magder, U. Meirav, and M. A. Kastner, "Kondo effect in a single-electron transistor," *Nature*, vol. 391, no. 6663, pp. 156–159, Jan. 1998.

[6] S. Al-Sarawi, D. Abbott, and P. Franzon, "A review of 3-d packaging technology," *Components, Packaging, and Manufacturing Technology, Part B: Advanced Packaging, IEEE Transactions on*, vol. 21, no. 1, pp. 2 –14, feb 1998.

[7] E. Sicard and S. Dhia, "An illustration of 90nm CMOS layout on pc," in *Devices, Circuits and Systems, 2004. Proceedings of the Fifth IEEE International Caracas Conference on*, vol. 1, 3-5 2004, pp. 315 – 318.

[8] F. Microelectronics, "2008.1 product guide, assp, memory, asic," 2008.

[9] K. Takahashi and M. Sekiguchi, "Through silicon via and 3-d wafer/chip stacking technology," in *VLSI Circuits, 2006. Digest of Technical Papers. 2006 Symposium on*, 0-0 2006, pp. 89 –92.

[10] J.-S. Kim, C. S. Oh, H. Lee, D. Lee, H.-R. Hwang, S. Hwang, B. Na, J. Moon, J.-G. Kim, H. Park, J.-W. Ryu, K. Park, S.-K. Kang, S.-Y. Kim, H. Kim, J.-M. Bang, H. Cho, M. Jang, C. Han, J.-B. Lee, K. Kyung, J.-S. Choi, and

Copyright ©2011 John Wiley & Sons, Ltd.

Y.-H. Jun, “A 1.2v 12.8gb/s 2gb mobile wide-i/o dram with 4 #x00d7;128 i/os using tsv-based stacking,” in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2011 IEEE International*, feb. 2011, pp. 496 –498.

[11] A. V. Biest, D. Milojevic, and F. Robert, “Key enablers for next generation system-level design in microelectronics,” in *Proc. 10th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI), Orlando (Florida), 16-19 July 2006*, 2006.

[12] E.-G. Talbi, *Metaheuristics : from design to implementation*. John Wiley & Sons, 2009.

[13] N. Doan, F. Robert, Y. D. Smet, and D. Milojevic, “Mcda-based methodology for efficient 3d-design space exploration and decision,” *International Symposium on System-on-Chip Proceedings (SoC 2010)*, pp. 76–83, September 2010.

[14] S. Das, A. Fan, K.-N. Chen, *et al.*, “Technology, performance, and computer-aided design of three-dimensional integrated circuits,” pp. 108–115, 2004.

[15] E. Beyne and B. Swinnen, “3D System Integration Technologies,” *Integrated Circuit Design and Technology, 2007. ICICDT ’07. IEEE International Conference on*, pp. 1–3, May 2007.

[16] P. Pieters and E. Beyne, “3d wafer level packaging approach towards cost effective low loss high density 3d stacking,” in *Electronic Packaging Technology, 2006. ICEPT ’06. 7th International Conference on*, 26-29 2006, pp. 1 –4.

[17] K. Kumagai, C. Yang, H. Izumino, *et al.*, “System-in-silicon architecture and its application to H.264/AVC motion estimation for 1080HDTV,” *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, pp. 1706–1715, Feb. 2006.

[18] T. Kgil, S. D’Souza, A. Saidi, *et al.*, “PicoServer: using 3D stacking technology to enable a compact energy efficient chip multiprocessor,” *SIGPLAN Not.*, vol. 41, no. 11, pp. 117–128, 2006.

[19] A. W. Topol, J. D. C. La Tulipe, L. Shi, *et al.*, “Three-dimensional integrated circuits,” *IBM J. Res. Dev.*, vol. 50, no. 4/5, pp. 491–506, 2006.

[20] S. Gupta, M. Hilbert, S. Hong, *et al.*, “Techniques for Producing 3D ICs with High-Density Interconnect,” in *21st International VLSI Multilevel Interconnection Conference, Waikoloa Beach*, 2004.

[21] P. Vincke, *Multicriteria Decision-Aid*. J. Wiley, New York, 1992.

[22] M. Ehrgott and X. Gandibleux, *Multiple Criteria Optimization. State of the art annotated bibliographic surveys*. Kluwer Academic, Dordrecht, 2002.

[23] R. E. Steuer, *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley, New York, 546 pp, 1986.

[24] R. T. Marler and J. S. Arora, “Survey of multi-objective optimization methods for engineering,” *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, Apr. 2004. [Online]. Available: http://dx.doi.org/10.1007/s00158-003-0368-6

[25] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multi-objective genetic algorithm: NSGA-II,” 2000.

[26] N. Padhye and S. Kalia, “Rapid prototyping using evolutionary approaches: part 1,” in *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, ser. GECCO ’09. New York, NY, USA: ACM, 2009, pp. 2725–2728. [Online]. Available: http://doi.acm.org/10.1145/1570256.1570394

[27] C. Silvano, W. Fornaciari, and E. Villar, *Multi-Objective Design Space Exploration of Multiprocessor SoC Architectures: The Multicube Approach*. Springer, 2011. [Online]. Available: http://books.google.be/books?id=WCa7CUNLurMC

Copyright ©2011 John Wiley & Sons, Ltd.

[28] L. Thiele, S. Chakraborty, M. Gries, and S. Knzli, “Design space exploration of network processor architectures,” in *In Network Processor Design: Issues and Practices, Volume 1*. Morgan Kaufmann Publishers, 2002, pp. 30–41.

[29] L. Thiele, S. Chakraborty, M. Gries, and S. Künzli, “A framework for evaluating design tradeoffs in packet processing architectures,” in *Proceedings of the 39th annual Design Automation Conference*, ser. DAC ’02. New York, NY, USA: ACM, 2002, pp. 880–885. [Online]. Available: http://doi.acm.org/10.1145/513918.514136

[30] C. Sechen, *VLSI placement and global routing using simulated annealing*, ser. Kluwer international series in engineering and computer science. Kluwer Academic Publishers, 1988. [Online]. Available: http://books.google.be/books?id=WSpTAAAAMAAJ

[31] J. de Vicente, J. Lanchares, and R. Hermida, “Placement by thermodynamic simulated annealing,” *Physics Letters A*, vol. 317, no. 56, pp. 415 – 423, 2003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0375960103013653

[32] T.-Y. Sun, S.-T. Hsieh, H.-M. Wang, and C.-W. Lin, “Floorplanning based on particle swarm optimization,” in *Emerging VLSI Technologies and Architectures, 2006. IEEE Computer Society Annual Symposium on*, vol. 00, march 2006, p. 5 pp.

[33] J. Keinert, M. Streub&uhorbar;hr, T. Schlichter, J. Falk, J. Gladigau, C. Haubelt, J. Teich, and M. Meredith, “Systemcodesigner – an automatic esl synthesis approach by design space exploration and behavioral synthesis for streaming applications,” *ACM Trans. Des. Autom. Electron. Syst.*, vol. 14, no. 1, pp. 1:1–1:23, Jan. 2009. [Online]. Available: http://doi.acm.org/10.1145/1455229.1455230

[34] T. Schlichter, M. Lukasiewycz, C. Haubelt, and J. Teich, “Improving system level design space exploration by incorporating sat-solvers into multi-objective evolutionary algorithms,” in *Emerging VLSI Technologies and Architectures, 2006. IEEE Computer Society Annual Symposium on*, vol. 00, march 2006, p. 6 pp.

[35] C. Erbas, S. Cerav-Erbas, and A. Pimentel, “Multiobjective optimization and evolutionary algorithms for the application mapping problem in multiprocessor system-on-chip design,” *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 3, pp. 358 – 374, june 2006.

[36] S. Mohanty and V. K. Prasanna, “Rapid system-level performance evaluation and optimization for application mapping onto soc architectures,” in *in Proc. of the IEEE International ASIC/SOC Conference*, 2002.

[37] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler, “Pisa - a platform and programming language independent interface for search algorithms.” Springer, 2003, pp. 494–508.

[38] K. Ueda, K. Sakanushi, Y. Takeuchi, and M. Imai, “Architecture-level performance estimation method based on system-level profiling,” *Computers and Digital Techniques, IEEE Proceedings -*, vol. 152, no. 1, pp. 12–19, 2005.

[39] C. Silvano, G. Palermo, V. Zaccaria, W. Fornaciari, R. Zafalon, S. Bocchio, M. Martinez, M. Wouters, G. Vanmeerbeeck, P. Avasare, L. Onesti, C. Kavka, U. Bondi, G. Mariani, E. Villar, H. Posadas, C. Y. Q., F. Dongrui, and Z. Hao, “Multicube: Multi-objective design space exploration of multiprocessor architectures for embedded multimedia applications,” in *Proceedings of the DATE’09 workshop on Designing for Embedded Parallel Computing Platforms: Architectures, Design Tools, and Applications*, Nice, France, April 2009.

[40] D. Milojevic, R. Radojcic, R. Carpenter, and P. Marchal, “Pathfinding: A design methodology for fast exploration and optimisation of 3d-stacked integrated circuits,” in *System-on-Chip, 2009. SOC 2009. International Symposium on*, oct. 2009, pp. 118 –123.

Copyright ©2011 John Wiley & Sons, Ltd.

[41] C. Weis, N. Wehn, L. Igor, and L. Benini, "Design space exploration for 3d-stacked drams," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*, march 2011, pp. 1 –6.

[42] Y. Xie, G. H. Loh, B. Black, and K. Bernstein, "Design space exploration for 3d architectures," *J. Emerg. Technol. Comput. Syst.*, vol. 2, no. 2, pp. 65–103, Apr. 2006. [Online]. Available: http://doi.acm.org/10.1145/1148015.1148016

[43] Y. Xie and Y. Ma, "Design space exploration for 3d integrated circuits," in *Solid-State and Integrated-Circuit Technology, 2008. ICSICT 2008. 9th International Conference on*, oct. 2008, pp. 2317 –2320.

[44] D. Milojevic, T. Carlson, K. Croes, R. Radojcic, D. F. Ragett, D. Seynhaeve, F. Angiolini, G. V. der Plas, and P. Marchal, "Automated pathfinding tool chain for 3D-stacked integrated circuits: Practical case study." in *3DIC*. IEEE, 2009, pp. 1–6.

[45] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplanning algorithm for 3D ICs," in *ICCAD '04: Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 306–313.

[46] H. Yan, Q. Zhou, and X. Hong, "Thermal aware placement in 3D ICs using quadratic uniformity modeling approach," *Integr. VLSI J.*, vol. 42, no. 2, pp. 175–180, 2009.

[47] G. Pelosi, "The finite-element method, part i: R. l. courant [historical corner]," *Antennas and Propagation Magazine, IEEE*, vol. 49, no. 2, pp. 180 –182, april 2007.

[48] D. Milojevic, L. Montperrus, and D. Verkest, "Power dissipation of the network-on-chip in multi-processor system-on-chip dedicated for video coding applications," *Journal of Signal Processing Systems*, p. 15, June 2008.

[49] F.-J. Veredas, M. Scheppler, W. Moffat, and B. Mei, "Custom implementation of the coarse-grained reconfigurable adres architecture for multimedia purposes." in *FPL*, T. Rissa, S. J. E. Wilton, and P. H. W. Leong, Eds. IEEE, 2005, pp. 106–111.

[50] L. Davis, "Adapting operator probabilities in genetic algorithms," in *Proceedings of the third international conference on Genetic algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989, pp. 61–69. [Online]. Available: http://dl.acm.org/citation.cfm?id=93126.93146

[51] J. Hart and A. Shogan, "Semi-greedy heuristics: An empirical study," *Operations Research Letters*, vol. 6, pp. 107–114, 1987.

[52] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *Evolutionary Computation, IEEE Transactions on*, vol. 7, no. 2, pp. 117 – 132, april 2003.

Copyright ©2011 John Wiley & Sons, Ltd.