Chapter 7
Optimization Methods and Applications to Microelectronics CAD

Salvatore Rinaudo, Valeria Cinnera Martino, Franco Fiorante, Giovanni Stracquadanio, and Giuseppe Nicosia

Abstract In many areas of research and design, simulators are a crucial tool for optimizing the relevant features of devices and for determining the effect of parameter variations on the output of a given system. Many commercially available simulation tools in the microelectronics industry are endowed with optimization programs, usually based on Least Squares Methods coupled with a numerical solver for minimizing, such as the normal equations or gradient methods. However these optimization codes are strictly linked to the whole commercial simulation package and cannot be easily adapted to the various requirements of an industrial environment. This chapter aims at introducing the most important algorithms and methods which could be of interest to CAD engineers working in universities or in several microelectronics companies. The examples which will be illustrated are real industrial cases and the results obtained with the cascade of simulators used within STMicroelectronics will be presented.

7.1 Motivation

In many areas of research and design, simulators are a crucial tool for optimizing the relevant features of devices and for determining the effect of parameter variations on the output of a given system.

The simulators are used to replace a large amount of experiments and measurements which are necessary to take into consideration the deterministic and stochastic behaviour of the manufacturing processes of electronic devices.

S. Rinaudo (✉) • V. Cinnera Martino • F. Fiorante
STMicroelectronics, Stradale Primosole 50, 95121, Catania, Italy
e-mail: salvatore.rinaudo@st.com; valeria.cinnera@st.com; franco.fiorante@st.com

G. Nicosia • G. Stracquadanio
Department of Mathematics and Computer Science, University of Catania, V.le A. Doria 6, 95125, Catania, Italy
e-mail: nicosia@dmi.unict.it; stracquadanio@dmi.unict.it

M. Günther (ed.), *Coupled Multiscale Simulation and Optimization in Nanoelectronics*, Mathematics in Industry 21,
DOI 10.1007/978-3-662-46672-8_7

A very important application of optimization is the *parameter extraction*, by which the parameters characterizing a given device within a given mathematical model can be obtained from the measurements of some characteristics of the device.

For instance the parameters required to model the behaviour of a device are related to physical quantities such as mobility, recombination and so on. When a device is described by an equivalent circuit, as for example the Gummel-Poon model, used by many circuit simulators (e.g. SPICE [36]), to perform the characterization of the electrical behaviour of bipolar transistors, the parameters are related to the electrical components of the circuit.

Many commercially available simulation tools in the EDA field of microelectronics industry are endowed with optimization programs, usually based on Least Squares Methods coupled with a numerical solver for minimizing, such as the normal equations or gradient methods. *However these optimization codes are strictly linked to the whole commercial simulation package and cannot be easily adapted to the various requirements of an industrial environment.*

For instance, in some industrial applications, a designer would like to have a global optimizer, which, being computationally expensive, is usually not available in the commercial package. However the greater computational cost of global optimization could be tolerable in an industrial context if efficient use be made of the computing power of a given design unit (e.g. by an appropriate use of a cluster of multivendors workstations). Another important example of great interest is the optimization of a cost function which is computed by using several simulation codes which are not integrated in a single software package and have been provided by different software vendors. Still another example would be the need to add more functions to the optimization (or parameter extraction) which are not usually found in commercial optimization software.

Because of these various demands for customized optimization and tolerance analysis in a CAD/CAM unit, several research groups in the microelectronics industry have developed their own optimization packages.

In particular the TCAD unit of STMicroelectronics has developed a post processing package called *EXEMPLAR* [33, 34] which encompasses global optimization and advanced statistical sensitivity analysis for the cascade of commercial and in house simulators, which are widely used by the design engineers throughout the company. The activity, started two decades ago in ST with the development of Exemplar, has been extended in the COMSON project.

This chapter aims at introducing for the most important algorithms and methods which are part of the optimization framework Exemplar which could be of interest to CAD engineers working in universities (graduate students) or in several microelectronics companies.

The examples which will be illustrated are *real industrial cases* and the results obtained with the cascade of simulators used within STMicroelectronics will be presented. This textbook is addressed to:

- Researchers in the microelectronics industry working in the CAD area. In particular those who must keep and update the commercial simulation software

used by CAD engineers; and must also integrate different commercial simulation software within optimization environments.
- PhD or advanced students in electrical engineering, computer science and applied mathematics.

7.2 The Optimization Problem

Optimization is the process by which one finds that value x that maximizes or minimizes a given function $f(x)$. The function f is called *objective function*.

Except in linear case, optimization proceeds by iteration, that is, starting from an approximate trial solution, a good algorithm gradually refines the research space until a predetermined level of precision has been reached. An extremum of f (maximum or minimum point) can be either global or local.

Generally, the global extremum is required, even if to distinguish a local extremum from a global extremum is not so simple. A technique to determine the global minimum could be to vary the initial point and take as extremum the one among all that results to be the minimum or maximum in absolute (if they are not all equal). If necessary, a high number of initial points can be generated in a random way. Another technique could be to perturb a local extremum to verify if the algorithm gives again the same extremum. Relatively recent techniques such Simulated Annealing and Genetic Algorithm are designed to minimize functions that are not smooth and that may have many local minima. Simulated Annealing algorithms introduce a random element into the iteration process, giving the algorithm a change to escape from a local extremum. Genetic Algorithms carry information about multiple candidates for the global extremum that are simultaneously refined as iteration proceeds.

In the optimization field it is necessary to make another difference between **constrained optimization** and **unconstrained** optimization; we talk about constrained optimization when the x value which minimizes or maximizes f has to satisfy a priori one or more constraints.

Having established the optimization as unconstrained, we must choose an optimization method. First of all, we must choose between methods that need only evaluations of the function to be minimized and methods that also require evaluations of the derivatives of f. In the multidimensional case, this derivative is the gradient (∇f), that is the vector whose components are the partial derivatives of f with respect to x_i for i $= 1, \dots,$ n.

Algorithms using the derivative are somewhat more powerful than those using only the function, but not always enough so as to compensate for the additional calculations of derivatives. Another criterion to be taken into account, to choose an optimization method, is related to the quantity of memory it requires. We must choose between methods that require storage of order n^2 and those that require only of order n, where n is the number of dimensions. For moderate values of n and reasonable memory sizes this is not a serious constraint. There will be, however, the occasional application where storage may be critical. Among the methods which does not require the derivatives calculation and require a storage of order n^2, we have to focus more attention on **Nelder and Mead's Simplex Method** and on the **Powell's Method**. For as much as regards the algorithms which require the first derivatives calculation we can consider two major families of algorithms: **Conjugate Gradient Method** and **Quasi-Newton Methods**. The former requires only of order n storage, while the Quasi-Newton Methods require of order n^2 storage. Both families require a one-dimensional minimization sub-algorithm, which can itself either use, or not use, the derivatives calculation.

The *EXEMPLAR* optimizer widely used by ST design engineers, is mainly based on optimization methods without derivatives. Some of these methods are: the Simplex method [28], the Powell [31], the CRS (*Controlled Random Search*) [7, 32] and the Direct method [26].

In the present chapter we describe the integration of ST PAN modeling flow [4, 5] with the EXEMPLAR framework in which the innovative *Discretized Immune Algorithm* (DIA) is encapsulated (Fig. 7.4).

7.3 Parameter Extraction for Compact Circuit Models

High-Voltage discrete *power MOSFETs* robustness is hardly tied to the topology of the layout device. Weakness in layout designs could produce, for example, during a classic UIS (Unclamped Inductive Switching), current focusing known as "hot spots" [12, 27] that could compromise the integrity of the entire device. It will be shown that an automatic optimization of a *power MOSFET*. layout can reduce the peak currents focused in hot spot areas during an UIS. The device is modeled using the innovative Power Analyzer (*PAN*) technique [4, 5] based on an accurate extraction of a spice-like model of the power device starting from its physical layout representation. The optimization is based on a framework fully integrated within the PAN modeling flow, which utilizes many of the most used and effective *state-of-art* optimization algorithms [23].

7.3.1 Automatic Physical Layout Optimization of Discrete Power MOSFETs for Reducing the Effects of Current Density

Discrete power MOSFET device represents nowadays a class of power devices highly requested in the field of SMPS (Switched-Mode Power Supply) for Servers, Solar & Desktop, AC/DC Converters, Battery Chargers, etc. due to their minimized gate charge, high speed switching and lowest $R_{DS}(ON)$ (Static drain-source on resistance).

The basic internal structure of a power MOSFET is made up of several elementary transistor cells connected in parallel rows in order to achieve the current handling capability required by the design application. Each row, and hence each single cell, is powered by a gate metal path that branches from the gate pad and stretching itself across the entire device. In the layout, there are cells displaced in areas far and very near to the gate pad; the result is that the gate impedance of each cell seen towards the gate pad could vary in relation to its distance to the gate pad [2, 6]. As a consequence of this non uniformity, it is possible to observe during the turn-on or turn-off switching that some cells will receive or loose the gate signal at different times causing a different behavior in terms of current carried. For example, at high switching frequencies the time required by the gate signal to reach the farthest elementary cell may be comparable to the switching times of the input signal. Therefore, this provides fast switching times for cells near to the gate pad and increasing delays for those located furthest away. The fast turn off for only portions of the device forces the remainder to drive large amounts of current during switching [2]. The result is a dramatic current density increase in the slowest parts of the device causing what we call a "hot spot" . Hot spots are restricted areas where probable thermal failures can make devices less robust. Also, in a UIS turn-off, peaks of current forced by the inductor load will be carried only by those slower cells, therefore, a better distribution of the cells is necessary to minimize the peak current which is a constraint that the designer should consider in order to develop a more robust devices [8, 24, 25]. This target could be reached by modifying the distribution of the gate metal across the device but paying attention not to vary the gate impedance of the whole device. This is strongly tied to the displacement of the metal path which is often a given parameter requested by the customer.

A new optimization framework flow will be demonstrated using new optimization algorithms that will modify the geometry of the layout. Moving element positions in the layout can cause an increase or decrease of the current density in the device depending on where elements are placed. Power Mos is made up of elements such as metal gate, fingers, gate pad, source windows, etc. which is shown in Fig. 7.1.

For example, moving the metal fingers closer together can cause current to reach some elementary MOS gates quicker and slower for others because the path traveled may be decreased or increased. The number of relative placements for elements can be enormous, therefore, a new optimization tool is designed to aid this process.

7.3.2 Modeling Approach

In order to better understand how the whole flow works, we will focus our attention on a spice-like netlist model used in the flow. This model is produced by an extraction starting from a CAD *(Computer Aided Design)* layout view of the device. Since, the layout usually has many different CAD layers that are not useful during

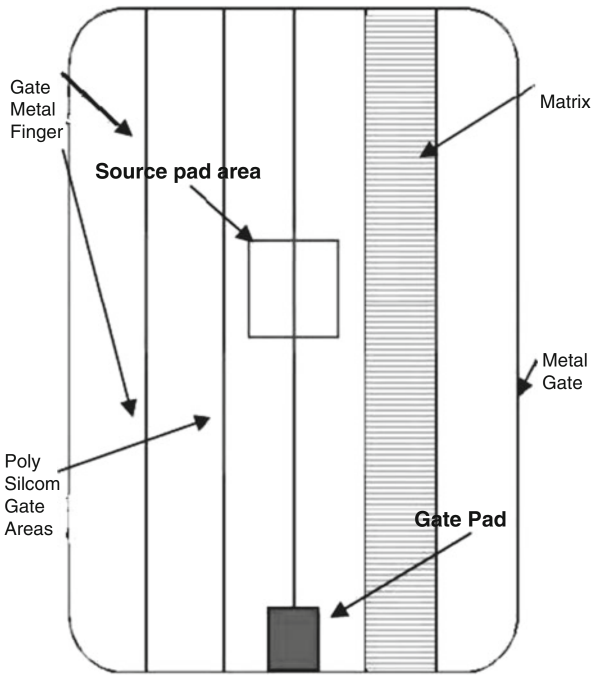

Fig. 7.1 Simplified layout of the Device under investigation

the final extraction. A simplified CAD view of the device is effectively used by the modeling tool as input.

The extracting tool, PAN is used to extract the netlist of the device in two steps from the simplified view.

The first step is to extract a numerical model which is strictly related to the layout topology. Each layer is mapped to a number, for example, an elementary cell is represented by the number 2. The second step starts from the numerical model which automatically produces the spice-like netlist model through indexes interpretation. The cell named as “MOS” corresponds to the model of a single elementary cell which must be supplied by the user and it could also be extracted with the aid of a TCAD *(Technology CAD tool)* simulation tool or by measurements on a wafer.

The peculiarity of this spice-like modeling flow is once the numerical model is produced, then the final netlist could be easily extracted starting from this data. This very important possibility releases the designer from the original layout so that many more analysis can be performed simply by modify the numerical model written in ASCII format which implies modification of the physical layout structure (es. number of fingers, pad size, position, etc.).

Since the position of elements has been discretized such that they are put on a 2D lattice with integer coordinates, the algorithms have to find the x and y coordinates for each element of the circuit under observation.

It is important to outline that any algorithm that is able to work with discrete variables are suitable for this problem, because it can be tackled as a *black box* *optimization* problem. In our work, we use, through the Exemplar optimizer, four algorithms that are the *state-of-art* in black box and circuit design optimization: in particular, we use Controlled Random Search (CRS), and Controlled Random Search Enhanched (CRS-E) [7]. The modelling methodology has been evaluated in several of its aspect and a U.S. Patent has also been deposited [5].

7.4 The Optimization Algorithm

Designing micro-electronic devices is a complex process, which takes into account increasing frequency and bandwidth ranges, small size factor, high reliability and low power consumption [3, 20, 29]. From a mathematical point of view, there are three major aspects to take into account; the formulation of a formal model of the system, the performance optimization and the robustness analysis. In this chapter, we focus on finding an optimal design for the power MOSFET. The space of solutions defined by the parameters of the power MOSFET is enormous and it is highly rugged. Moreover, the power MOSFET is a complex device which simulation requires $\sim$5 min; due to this expensive computational cost, an optimization algorithm has to find good solution using tight budget of simulations [3, 20, 29]. In order to tackle effectively this optimization problem, we design a new OPTIMIZATION IMMUNOLOGICAL ALGORITHM, called OPTIA [13, 18]. The OPTIA is a stochastic optimization algorithm based on the Human Clonal Selection Principle of the Immune System (IS) [1, 16, 17].

The IS is an excellent example of bottom-up optimization strategy through which adaptation operates at the local level of cells and molecules, and useful behavior emerges at the global level [15, 19]. In particular, the Clonal Selection theory shows that B and T lymphocytes, that are able to recognize the antigen, will start to proliferate by cloning upon recognition of such antigen. When a B cell is activated by binding an antigen, many clones are produced in response via the so called *clonal expansion*. The newly created cells can undergo to somatic hypermutation, creating offspring B cells with mutated receptors: the higher the affinity of a B-cell to the antigens, the more likely it will clone. This results in a Darwinian process of variation and selection, called *affinity maturation*. The increase in size of these populations couples with the production of cells with longer than expected lifetimes, assuring the organism a higher specific responsiveness to that antigenic attack in the future: the so called *immunological memory of the system*.

OPTIA tries to mimic the clonal selection principle for optimization: a problem is an antigen and a B-cell is a candidate solution. The affinity between an antigen and a B-cell is given by the objective function of the optimization problem [13, 14, 35].

Each B-cell is a vector of real values of dimension n, where n is the dimension of the problem; moreover each candidate solution has associated an age τ: it indicates the number of iterations since the last successful mutation [9, 13, 21]. Initially the age is set to zero.

Fig. 7.2 Pseudo-code of the Optimization Immune Algorithm (optIA)

```
1:  procedure OPTIA($d; dup; \tau_B; \rho; \beta; s_a$)
2:      $t \leftarrow 0$
3:      $BC_{arch} \leftarrow Create\_Archive(s_a)$
4:      $P^{(t)} \leftarrow Initialize(d)$
5:      $Evaluate\,(P^t)$
6:      while $\neg Termination\_Condition()$ do
7:          $P^{(clo)} \leftarrow Cloning(P^{(t)}; dup)$
8:          $P^{(hyp)} \leftarrow Hypermutation(P^{(clo)}; \rho)$
9:          $P^{(macro)} \leftarrow Macromutation(P^{(hyp)}; \beta)$
10:         $Evaluate(P^{(macro)})$
11:         $Aging(P^{(t)}; P^{(macro)}; \tau_B)$
12:         $P^{(t+1)} \leftarrow Selection(P^{(t)}; P^{(macro)}; BC_{arch})$
13:         $t \leftarrow t + 1$
14:     end while
15: end procedure
```

An initial population $P^{(0)}$ of dimension d is generated randomly, with each variable constrained into its lower and upper bounds. However, it could be useful to use an ad-hoc population to start the optimization process: optIA can take in input a *starting point* p_{st}, and it use this point to initializes one B-cell of the population and the remaining $d - 1$ candidate solutions are initialized with vectors obtained as a perturbation of p_{st}.

The algorithm is iterative: each iteration is made of a cloning, mutation and selection phase [9, 10, 30]. The algorithm stops when a given stopping criterion is verified: in particular, it ends when a maximum number of fitness function evaluations is reached.The pseudo-code of the algorithm is shown in Fig. 7.2.

The cloning phase is responsible for the production of new B-cell. Each B-cell is cloned *dup* times producing a population $P_{N_c}^{(clo)}$ of size $d \cdot dup = N_c$, where each cloned B-cell takes the same age of its parent. At the same time, the age of the parent is increased by one.

After the $P^{(clo)}$ population is created, it undergoes to the mutation phase in order to find better solutions. In the mutation phase, the hypermutation and hypermacromutation are applied to each candidate solutions [11]. These operators are the principle responsible of the exploring and exploiting ability of the algorithm.

The hypermutation operator is based on the *self-adaptive gaussian mutation* that is computed by:

$$\sigma_i' = \sigma_i * exp((\tau * N(0,1)) + (\tau' * N_i(0,1))) \tag{7.1}$$

$$x_i^{new} = x_i + \sigma * N(0,1). \tag{7.2}$$

The hypermacromutation applies a convex mutation to a given solution according to the following equation:

$$x_i^{new} = (1 - \beta) * x_i + \beta * x_k; \tag{7.3}$$

where $x_i \neq x_k, \beta \in [0.1]$ is a random number obtained with uniform distribution. Since variables x_i and x_k typically have different ranges, the value x_k is normalized within the range of x_i using the following equation:

$$x_k^{norm} = L_i + \frac{(x_k - L_k)}{(U_k - L_k)} \times (L_i - U_i) \tag{7.4}$$

where L_i, U_i are the lower and upper bounds of x_i and L_k, U_k are lower and upper bounds of x_k. The value used to mutate the variables x_i is then x_k^{norm}.

The mutation operators are controlled by a mutation rate α that is differently defined according to the type of operator: for the hypermutation operator, it is defined as

$$\alpha = e^{(-\rho \times f)} \tag{7.5}$$

instead for the macromutation operator is defined as

$$\alpha = (\frac{1}{\beta}) \times e^{(-f)} \tag{7.6}$$

where f is the fitness function value normalized in [0, 1].

These operators are applied sequentially: the hypermutation operator acts on the $P^{(clo)}$ and it produces a new population $P^{(hyp)}$ and the hypermacromutation mutate the $P^{(hyp)}$ generating the $P^{(macro)}$ population.

The population $P^{(macro)}$ is evaluated: if a B-cell achieves a better objective function value, its age is set to zero otherwise it is increased by one.

The *Aging* operator is executed on $P^{(t)}$ and $P^{(macro)}$: it drops all the B-cells with age greater than $\tau_b + 1$, where τ_b is a parameter of the algorithm.

The B-cells deleted by the AGING operator are not discarded but they are saved into an archive BC_{arch} of size s_a: if there is enough space into the archive, the B-cell is saved into the first available location, otherwise a random location of the archive is selected and it is substituted by this new B-cell.

The selection is then performed and the new $P^{(t+1)}$ is created by picking the best individuals from the parents and the mutated B-cells: if $|P^{(t+1)}| < d, d - |P^{(t+1)}|$ B-cells are randomly taken from the archive and added to the new population.

Many real world problems associate to each variable of an optimization problem a fineness parameter in addiction to lower and upper bounds.

OPTIA is able to handle this kind of variable using a *grid based model*: a grid can be defined as an n-dimensional space which fineness is specified by a parameter δ.

When OPTIA evaluate the fitness of a B-cell, it projects the solution on the grid and successively it evaluates the fitness of the projected point. It is possible to define different strategies to project the point on the grid. For each variable of the problem,

Fig. 7.3 Extraction of the description file

```
...
gpad_width = 7
gpad_length = 7
gpad_x = 14
gpad_y = 8
finger_1_x = 6
finger_1_o = 9
finger_1_oy = 4
finger_2_x = 12
...
```

OPTIA uses the following projection equation:

$$\Pi(x, l, u, \delta) = min(|x - x_\delta^1|, |x_\delta^2 - x|) \tag{7.7}$$

$$x_\delta^1 = \frac{x - l}{\delta} \tag{7.8}$$

$$x_\delta^2 = \frac{u - x}{\delta} \tag{7.9}$$

where x is a variable of the problem, l, u are the relative lower and upper bounds and δ is the mesh fineness parameter.

7.4.1 The Optimization Flow

The methodology proposed is based on the possibility of modifying the device layout structure by simply modifying the contents of the numerical model stored in an ASCII file. Before using the numerical model in the optimization flow, it must be translated into a text description format using a custom language named DES [1] which is based on a series of parameters that exhaustively describes the content of the numerical model.

The DES file automatically created by custom software starting from the numerical model allows for a better definition of constraints that the optimizer must take into consideration. The DES parameters are geometric locations of elements within the power Mos layout, for example, the location of the finger 1 is given by the x,y coordinates and the width parameters such as finger_1_x, finger_1_oy and finger_1_o respectively. These parameters are setup as variables which are inputs to the optimizer that can be modified by the optimizer algorithm to find the optimal location. The same is true for each of the other parameters in Fig. 7.3.

The optimization flow works accordingly as shown in Fig. 7.4. Firstly, a reference starting geometric layout must be provided. This geometric layout is simplified into

[1]DES name used to describe the description of the numerical model matrix.

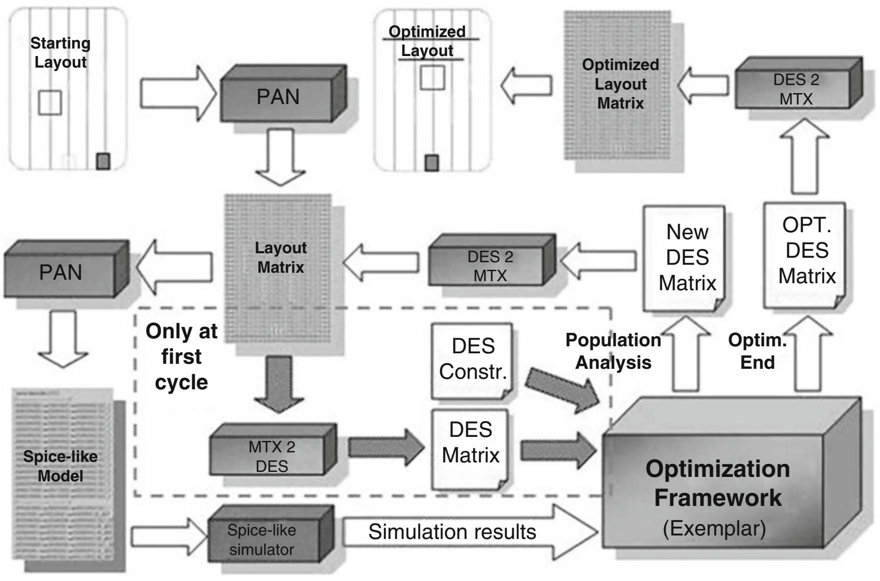

Fig. 7.4 IO Optimization Flow

a numerical mapped model of the layout which is extracted by the PAN tool so that it can be converted using DES2MTX[2] software into a detailed text description for each of the important elements. The text description contains the positional coordinates and sizes of each element which describes the make up of the discrete power MOS. These elements positions and sizes become the parameters which may be optimized in order to resolve the current density problem which has been previously described. Secondly, the detailed text description is provided as input to the optimization framework as shown in Fig. 7.3 along with constrained bounds for each elements parameters under evaluation which is provided on the first iteration cycle and used through out the entire optimization process. Also, at the same time the PAN tool also converts the numerical mapped layout into a spice like netlist which is simulated using a third party tool that is a circuit simulator such as Mentor Graphics Eldo [22] or a fast spice simulator; the results of the performances or simulation results are passed to the optimization framework as inputs. Next, the optimization algorithm goes to work by first reading these inputs and lastly providing new elements parameters values under evaluation. If the new values under optimization are not within the given constrained bounds (not yet optimal), a new text description file provided by the optimization framework is converted back to a numerical model by the DES2MTX software. Then it is converted to a new spice like netlist again,

[2]DES2MTX is software to convert description language to numerical mapped data of the PowerMos layout. At the same manner, also the Mtx2Des software has been developed.

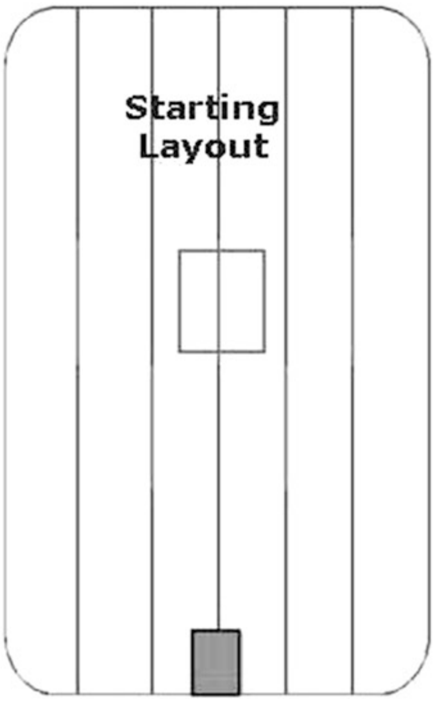

Fig. 7.5 Starting Layout

which is then simulated, and delivered to the optimization frame work. This flow is cycled until optimal solutions are found or the maximum number of iterations has been achieved by the optimizer, even if a solution cannot be obtained. However, if an optimal solution has been found, the text description file provided by the optimization framework is converted back to a numerical model by the DES2MTX. Finally, the PAN tool converts the new numerical model to its original geometric layout form as the optimized physical layout view.

7.4.2 Simulation Results

In this work, we used the optimization flow described in order to improve the robustness of power MOSFET devices in terms of maximum current density allowed in a UIS turn-off.

Taken as reference the starting layout of Fig. 7.5, we obtained as result of the optimization a better new layout where that maximum currentpeak is lower than the one in the starting layout. In the built spice-like model of the device, each single cell represents a portion of area of the real device; the size of that area depends on how dense the mesh has been chosen for the discretezation necessary to extract the matrix. An evaluation of the drain current referred to the elementary

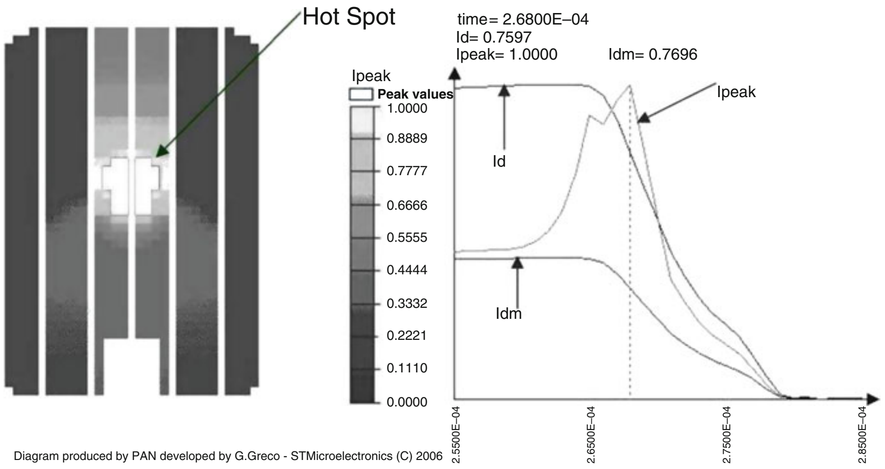

Fig. 7.6 Ipeak current 2D map and waveforms in a UIS turn-off for the starting reference layout

cell and, hence, to a given area, will give us very useful current density information. For technologist, such as parameter is known and represents a physical limit for the technology even if other causes of failure are to be searched in an excess of the silicon temperature reached due to the switching power and of some parasitic bipolars triggered by the high slopes of the drain currents flowing across the elementary cells [6]. All these aspects could be easily investigated using the PAN tool together with the optimization flow but in this work, only matters tied to peak currents are investigated and result presented. In figure 2D map and waveforms referred to the Ipeak currents across the reference starting layout is reported in Fig. 7.6 . The light area on dark background represents regions of the devices where currents are higher.

That 2D map gives information useful from the qualitatively point of view. Quantitatively information, instead, are given by the waveforms on the right part for Fig. 7.6 which reports the whole Idrain current across the drain terminal of the devices I_d, the average current I_{dm} that should have been if each cell of the devices would have switched ideally without gate/source delay and, at end, the maximum peak current found across all the cells of the devices. The difference between I_{peak} and I_{dm} represents an index of current unbalancing of the device. To study the effective of immune algorithm and two investigate the hardness of the problem, we have conducted a long series of simulation using various optimization algorithm: we have used the SIMPLEX, CRS and CRS-E methods. In our experimental protocol, when it is possible, we use as a starting point the circuit proposed by the designers or we generate one randomly. In particular, OPTIA can work completely *from scratch* or take an initial point, that is assigned to an individual and the other individuals of the population are small perturbations of this starting point.

The stopping criterion adopted is the attainment of a fixed number of simulations (in our case 10^4) or, for numerical methods, the achievement of the convergence.

Table 7.1 PowerMosfet optimization results using evolutionary and numerical algorithms

Algorithm	Initial point	Ipeak
Designer's point	–	9.948×10^{-3}
optIA	**Random point**	$\mathbf{7.281 \times 10^{-3}}$
Crs	Random point	7.296×10^{-3}
optIA	Designer's point	7.394×10^{-3}
Crs-E	Random point	7.397×10^{-3}
Simplex	Designer's point	7.894×10^{-3}

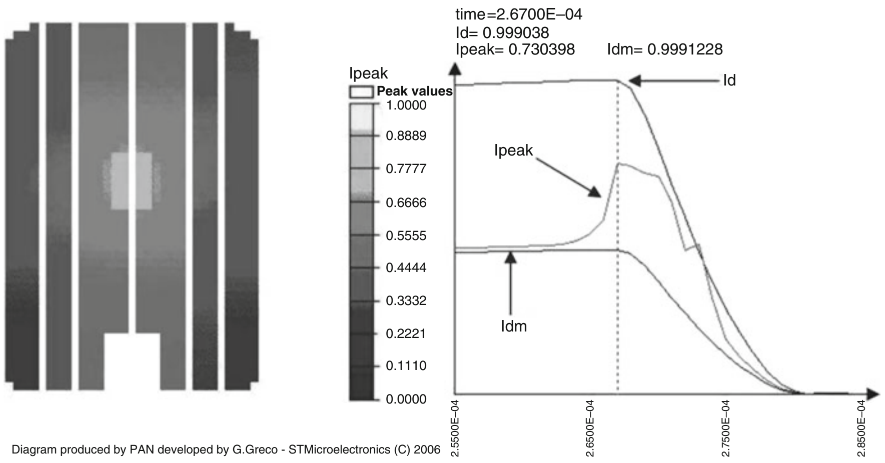

Fig. 7.7 Ipeak current 2D map and waveforms in a UIS turn-off for the optimized layout

In Table 7.1, we report for each algorithm the best solution found in terms of IPEAK by the various algorithms: by inspecting the results, OPTIA found the best solution using random points. From an optimization point of view, this is not surprising because the designer's circuit can be a local optimum that prevent the algorithm to find new good solutions. The 2D final current map and relative waveforms have been shown in Fig. 7.7. As it is possible to observe, during the analyzed range of time, where the current peak occurs, the final value for that parameter is normalized decreasing from 1.0000 to 0.73098 obtaining an improvement of approximately 27 %. In order to obtain this new result the optimizer has modified the geometry of the original layout by repositioning and modifying elements in the layout. The optimizer has decreased the size of poly silicon gate areas and repositioned them along the gate metal finger. It also repositioned the gate metal fingers in horizontal fashion. The final result is a new layout geometrically adjusted and optimized for the better (Fig. 7.8).

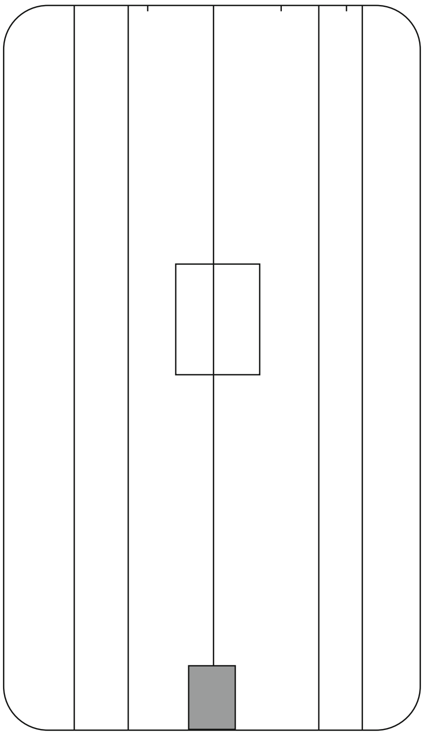

Fig. 7.8 Optimized Layout

7.5 Conclusion

In this work we have shown an innovative methodology for the power MOSFET design aimed to improve robustness and performances. This methodology opens new spaces in power device designing giving to the designers new innovative CAD tools that allows investigating problematic until now little afforded due to the lack of means. The work, yet in the preliminary phase has shown enormous potential for investigation in future work, and of course, will be treated. The designed evolutionary algorithm was shown to produce acceptable solutions in most cases, where classical techniques failed.

References

1. Anile, A., Cutello, V., Nicosia, G., Rascuna, R., Spinella, S.: Comparison among evolutionary algorithms and classical optimization methods for circuit design problems. In: The 2005 IEEE Congress on Evolutionary Computation, Edinburgh, vol. 1, pp. 765–772 (2005)
2. Biondi, T., Greco, G., Bazzano, G., Rinaudo, S.: Analysis of the internal current distribution in power mosfets operated at high switching frequency. In: Proceedings of MSED,Workshop on Modeling and Simulation of Electron Devices, Pisa, vol. 15, pp. 4–5 (2005)

3. Biondi, T., Ciccazzo, A., Cutello, V., D'Antona, S., Nicosia, G., Spinella, S.: Multi-objective evolutionary algorithms and pattern search methods for circuit design problems. J. Univers. Comput. Sci. **12**(4), 432–449 (2006)
4. Biondi, T., Greco, G., Bazzano, G., Rinaudo, S.: Effect of layout parasitics on the current distribution of power mosfets operated at high switching frequency. J. Comput. Electron. **5**(2–3), 149–153 (2006)
5. Biondi, T., Greco, G., Bazzano, G., Rinaudo, S.: Method for modeling large-area transistor devices, and computer program product therefore. U.S. Patent N. 11/770,578 deposited in (2007)
6. Biondi, T., Greco, G., Bazzano, G., Rinaudo, S., Allia, M., Liotta, S.: Distributed modeling of layout parasites in large-area high-speed silicon power devices. IEEE Trans. Power Electron. **22**(5) (2007)
7. Brachetti, P., De Felice Ciccoli, M., Di Pillo, G., Lucidi, S.: A new version of the Price's algorithm for global optimization. J. Global Optim. **10**(2), 165–184 (1997)
8. Budihardjo, I., Lauritzen, P., Mantooth, H.: Performance requirements for power MOSFET models. In: 25th Annual IEEE Power Electronics Specialists Conference, PESC'94 Record, Taipei, pp. 69–76 (1994)
9. Castrogiovanni, M., Nicosia, G., Rascuná, R.: Experimental analysis of the aging operator for static and dynamic optimisation problems. In: Knowledge-Based Intelligent Information and Engineering Systems, pp. 804–811. Springer-Verlag Berlin, Heidelberg (2007)
10. Ciccazzo, A., Halfmann, T., Marotta, A., Nicosia, G., Rinaudo, S., Stracquadanio, G., Venturi, A.: New coupled EM and circuit simulation flow for integrated spiral inductor by introducing symbolic simplified expressions. In: IEEE International Symposium on Industrial Electronics, ISIE 2008, Cambridge, pp. 1203–1208 (2008)
11. Conca, P., Nicosia, G., Stracquadanio, G., Timmis, J.: Nominal-Yield-Area Tradeoff in Automatic Synthesis of Analog Circuits: A Genetic Programming Approach Using Immune-Inspired Operators. In: 2009 NASA/ESA Conference on Adaptive Hardware and Systems, San Francisco, pp. 399–406. IEEE (2009)
12. Consoli, A., Gennaro, F., Testa, A., Consentino, G., Frisina, F., Letor, R., Magri, A.: Thermal instability of low voltage power-mosfets. IEEE Trans.Power Electron. **15**, 575–581 (2000)
13. Cutello, V., Nicosia, G.: An immunological approach to combinatorial optimization problems. In: Advances in Artificial Intelligence – IBERAMIA, Seville, pp. 361–370 (2002)
14. Cutello, V., Krasnogor, N., Nicosia, G., Pavone, M.: Immune Algorithm Versus Differential Evolution: A Comparative Case Study Using High Dimensional Function Optimization. Adaptive and Natural Computing Algorithms, ICANNGA 2007, April 11-14, 2007, Warsaw, Poland. Springer, Lecture Notes in Computer Science 4431:93–101 (2007)
15. Cutello, V., Lee, D., Leone, S., Nicosia, G., Pavone, M.: Clonal Selection Algorithm with Dynamic Population Size for Bimodal Search Spaces. Advances in Natural Computation, ICNC 2006, September 24-28, 2006, Xi'an, China. Springer, Lecture Notes in Computer Science 4221:949–958 (2006)
16. Cutello, V., Narzisi, G., Nicosia, G., Pavone, M.: An immunological algorithm for global numerical optimization. In: Artificial Evolution, pp. 284–295. Springer-Verlag Berlin, Heidelberg (2006)
17. Cutello, V., Nicosia, G., Pavia, E.: A parallel immune algorithm for global optimization. In: Intelligent Information Processing and Web Mining, IIS 2006, June 19-22, 2006, Ustron, Poland. Springer, Series on Advances in Soft Computing, pp. 467–475 (2006)
18. Cutello, V., Nicosia, G., Pavone, M.: Exploring the capability of immune algorithms: A characterization of hypermutation operators. In: Artificial Immune Systems, ICARIS 2004, September 13-16, 2004, Catania, Italy. Springer, Lecture Notes in Computer Science 3239:263–276 (2004)
19. Cutello, V., Narzisi, G., Nicosia, G., Pavone, M.: Real coded clonal selection algorithm for global numerical optimization using a new inversely proportional hypermutation operator. In: SAC 2006, Dijon, vol. 2, pp. 950–954. ACM (2006)

20. Cutello, V., Nicosia, G., Rascuna, R., Spinella, S.: Optimising an inductor circuit and a two-stage operational transconductance amplifier using evolutionary and classical algorithms. International J. Comput. Sci. Eng. **2**(3), 158–169 (2006)
21. Cutello, V., Nicosia, G., Romeo, M., Oliveto, P.: On the convergence of immune algorithms. In: IEEE Symposium on Foundations of Computational Intelligence, FOCI 2007, Honolulu, pp. 409–415 (2007)
22. Eldo user's manual. Mentor Graphics Corporation (2007)
23. Graeb, H.: Analog Design Centering and Sizing. Springer Netherlands, Dordrecht (2007)
24. Hohl, J., Galloway, K.: Analytical model for the single event burnout of power MOSFETs. IEEE Trans. Nucl. Sci. **34**, 1275–1280 (1987)
25. Hu, C., Chi, M., Patel, V.: Optimum design of power MOSFET's. IEEE Trans. Electron Devices **31**(12), 1693 (1984)
26. Jones, D., Perttunen, C., Stuckman, B.: Lipschitzian optimization without the Lipschitz constant. J. Optimiz. Theory Appl. **79**(1), 157–181 (1993)
27. Kraus, R., Mattausch, H.: Status and trends of power semiconductor device models for circuit simulation. IEEE Trans. Power Electron **13**, 452–465 (1998)
28. Nelder, J., Mead, R.: A simplex method for function minimization. Comput. J. **7**(4), 308 (1965)
29. Nicosia, G., Rinaudo, S., Sciacca, E.: An evolutionary algorithm-based approach to robust analog circuit design using constrained multi-objective optimization. In: Research and Development in Intelligent Systems XXIV, 10-12 December 2007, Cambridge, England, UK. Springer, pp. 7–20 (2007)
30. Nicosia, G., Rinaudo, S., Sciacca, E.: An evolutionary algorithm-based approach to robust analog circuit design using constrained multi-objective optimization. In: Research and Development in Intelligent Systems XXIV, 10-12 December 2007, Cambridge, England, UK. Springer, pp. 7–20 (2007)
31. Powell, M.: An efficient method for finding the minimum of a function of several variables without calculating derivatives. Comput. J. **7**(2), 155 (1964)
32. Price, W.: A controlled random search procedure for global optimisation. Comput. J. **20**(4), 367 (1977)
33. Rinaudo, S., Moschella, F., Anile, A.M., O.Muscato: Controlled random search parallel algorithm for global optimization with distributed processes on multivendor cpus. In: Arkeryd, L., et al. (eds.) Progress in Industrial Mathematics – ECMI 98, Gothenburg. B.G. Teubner, Stuttgart/Leipzig (1999)
34. Rinaudo, S., Moschella, F. & Anile, A., M.: Parallel implementation in an industrial framework of statistical tolerancing analysis in microelectronics. In: EURO-PAR'99 Parallel Processing. Lecture Notes in computer Science, vol. 1685. Springer, Berlin/Heidelberg (1999)
35. Tarakanov, A., Nicosia, G.: Foundations of immunocomputing. In: IEEE Symposium on Foundations of Computational Intelligence, FOCI 2007, Honolulu, pp. 503–508 (2007)
36. Vladimirescu, A.: The SPICE book. Wiley, New York (1994)