

Improved Performance and Variation Modelling for Hierarchical-based optimisation of Analogue Integrated Circuits

Sawal Ali, Li Ke, Reuben Wilcock and Peter Wilson

Electronics System Devices Group, School of Electronics and Computer Science

University of Southampton, UK

{shma05r, kl05r, rw3, prw}@ecs.soton.ac.uk

Abstract

A new approach in hierarchical optimisation is presented which is capable of optimising both the performance and yield of an analogue design. Performance and yield trade offs are analysed using a combination of multi-objective evolutionary algorithms and Monte Carlo simulations. A behavioural model that combines the performance and variation for a given circuit topology is developed which can be used to optimise the system level structure. The approach enables top-down system optimisation, not only for performance but also for yield. The model has been developed in Verilog-A and tested extensively with practical designs using the Spectre simulator. A performance and variation model of a 5 stage voltage controlled ring oscillator has been developed and a PLL design is used to demonstrate hierarchical optimisation at the system level. The results have been verified with transistor level simulations and suggest that an accurate performance and yield prediction can be achieved with the proposed algorithm.

1 Introduction

Advances in silicon technology over the last decade have led to increased integration of analogue and digital functional blocks onto the same chip. In such a mixed signal environment, the analogue circuits must use the same transistors as their digital neighbours. The increasing complexity and accuracy of device models has led to wide acceptance of simulation and optimisation based design techniques for the design of analogue blocks rather than hand calculations [1-3]. With reducing transistor sizes, the impact of process variations on analogue design has become very prominent and can lead to circuit performance and yield falling below specification. This issue has led to the consideration of yield in the design process, known as design for yield (DFY) [4]. In optimisation-based design techniques, the performance of the circuit must be evaluated for a large number of different circuit variables, a process known as design space

exploration. Running the entire performance evaluation at transistor level is computationally intensive especially for large and complex circuits. Due to this limitation, hierarchical design is commonly used to break down a large system into its constituent building blocks. A typical hierarchical design is shown in figure 1. Not only does this approach simplify the design task but it also speeds up the design flow by encouraging design reuse. Hierarchical based optimisation is one method used to reduce simulation time and involves the use of behavioural models prior to transistor level simulation [5-7]. Behavioural and macro modelling is a useful technique that involves developing models from simulation that relate performance to circuit parameters. Although the initial time investment is high, subsequent design flows are significantly faster [8]. Recently, macromodelling has been used to predict the parametric yield and performance of a design [9].

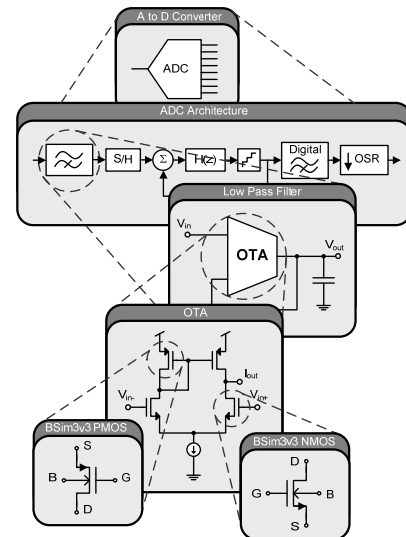


Figure 1. A typical system design hierarchy.

In this paper, a novel approach is proposed that develops a combined performance and statistical variation

behavioural model for analogue circuits. This approach is an improvement over the work presented in [10] in the way the variation model being used at the system level optimisation. On top of that the example demonstrated the impact of the methodology with higher number of variables, performance functions and complex spice simulation. The remainder of this paper is organized as follows: Section 2 provides necessary background; the proposed algorithm is detailed in section 3 and the example results are given in section 4. Concluding remarks are made in section 5.

2 Background

Simulation-based optimisation techniques are widely used for analogue circuit design and several synthesis tools have been developed that use spice-like simulators for their evaluation engine. However, these approaches are processor intensive which limits their use to smaller building blocks [11]. To overcome this problem, hierarchical based design has been proposed to divide the large system into sub blocks that can be optimised separately [11].

2.1 Multi-Objective Optimisation

The optimisation formulation for more than one objective function is called multi-objective optimisation (MOO) which can be generally stated as:

$$\begin{aligned} & \text{Minimise / Maximise } f_m(x), m = 1, 2, \dots, M \\ & \text{Subject to } g_j(x) \geq 0, j = 1, 2, \dots, J \end{aligned} \quad (1)$$

Where $fn(x)$ is the set of M performance functions and $gj(x)$ is the set of J constraints and the outcome is a set of optimal solutions [12]. Figure 2 shows the relationship between the parameter space and objective space, where each point in the parameter space is a solution that corresponds to a point in the objective space. The black curve shown on the objective space is called the *Pareto front* and all solution points lying on this curve are called Pareto-optimal solutions. The algorithm used in this work for the MOO is called Non-dominated Sorting Genetic Algorithm -II (NSGA-II). This evolutionary algorithm employs an elite preserving strategy which makes sure that good design solutions found early in the optimisation will be carried to the next generation. The following shows an outline flow of an NSGA-II algorithm:

NSGA Algorithm

- Generate initial random population, size N .
 - Create offspring population.
 - Combine parent and offspring population to form R_t . ($R_t = P_t \cup Q_t$)
 - Perform non-dominated sorting and identify fronts, F_i ($i=1, 2, \dots$ etc)
 - Set new population, $P_{t+1} = \emptyset$, and fill P_{t+1} with $F_1, (P_{t+1} \cup F_i)$ as long as $|P_{t+1}| + |F_i| < N$.
 - Perform crowding sort and place most widely spread solution in P_{t+1}
 - Create offspring population Q_{t+1} from P_{t+1} and repeat until last number of generation.
-

2.2 Table Model Functions

Behavioural models employing table model functions require the generation of sampled data points from circuit simulation. Interpolation and extrapolation techniques are then used to estimate a new value from the set of known values. Verilog-A supports three types of spline interpolation: linear, quadratic and cubic. The choice of interpolation is a trade off between accuracy and complexity. Cubic spline interpolation has been employed in this work to maximise accuracy. The third degree polynomial used to create the piece-wise interpolation curve is defined by equation (3), where a_i , b_i , c_i and d_i are the coefficients for the polynomials.

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \quad (3)$$

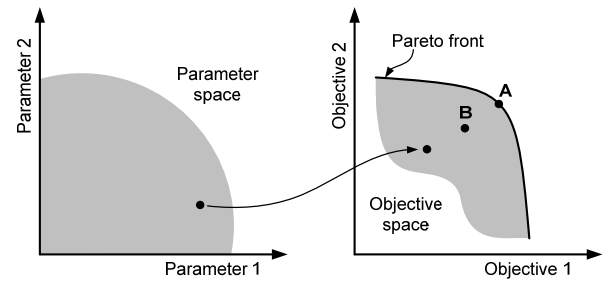


Figure 2. Parameter space and objective space.

2.3 Hierarchical-based Design

A hierarchical methodology consists of a top-down design and bottom-up verification process. Important aspects of both processes are circuit decomposition and specification propagation. Once the system architecture has been divided into sub-blocks, automatic optimisation algorithms can be applied to solve for the circuit sizing. The optimisation for hierarchical based design can be divided into two steps: In step one, the behavioural level blocks are optimised using a behavioural simulation and the design parameters that meet the system level specification are determined. In the second step, the design parameters from the previous optimisation are taken as the specifications for the circuit level optimisation which propagates the system level specification to the bottom level. The relationship between transistor level and system level optimisation in hierarchical design is shown in figure 3. The behavioural-level Pareto front determines all the solution points that meet the system level specifications. The design space of this Pareto front is then taken as the design objective for the sub-block circuit level. At the circuit level, the Pareto front is then used to determine the design parameters that best meet the design objective, resulting in the transistor dimensions.

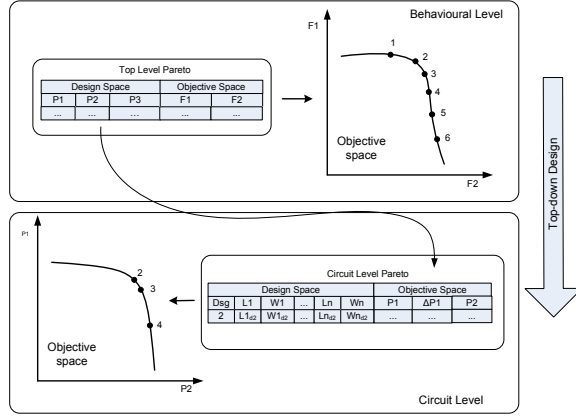


Figure 3. Hierarchical optimisation

3 Proposed Algorithm

The key steps in the proposed algorithm are shown in Figure 4. These steps are now discussed in more detail.

3.1 Netlist and Objective Function Generation

The starting point for the proposed algorithm is a circuit topology, process models and a set of performance functions. The first step involves generating a transistor level netlist for the chosen circuit topology. From this netlist a set of designable parameters are derived which will be used to change the circuit's performance. Examples of designable parameters include a transistor's length and width. Each parameter will have constraints imposed by the designer and once determined, these define the parameter space. The performance functions of the circuit are defined as the objective functions and testbench netlists are defined to simulate the performance for a certain set of design parameters.

3.2 Multi-Objective Optimisation

In this stage, the design space is explored and the objective functions are improved iteratively. The optimisation implementation is based on an evolutionary algorithm known as Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [12]. The genetic algorithm procedure involves generating a number of individuals and optimising these over a number of generations. The individuals are encapsulated in a set of parameters defined as the GA string. During the optimisation, the algorithm determines the quality of the individuals through the fitness score of each individual. The fitness score is measured from the performance evaluation.

3.3 Performance and Variation Modelling

The outcome of the multi objective optimisation is a Pareto-front (a set of optimal solutions). All the solution points on the Pareto-front and their respective design parameters can be extracted and a model can be created

that represents these data points. Interpolation is one of the techniques that can be used to model these data points.

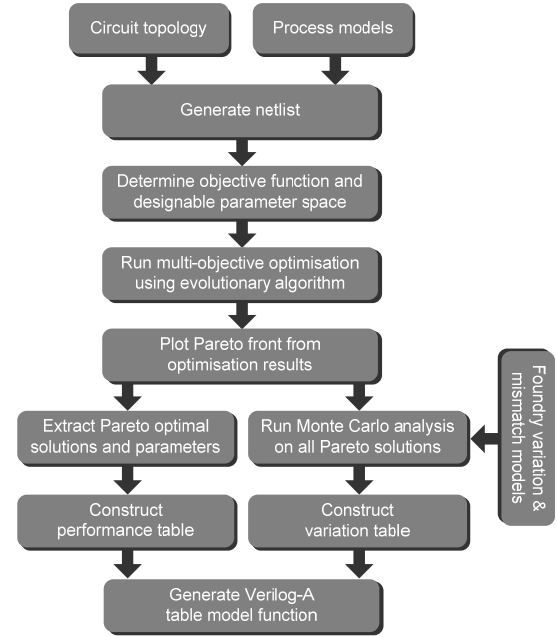


Figure 4. Novel yield targeted algorithm.

In this technique, the number of fitting parameters in the extraction process matches the number of samples in the data points meaning that all the data points are used for the model. These data points can be stored in a look-up table and interpolation can be applied to the table to find intermediate design points. The Verilog-A behavioural language is used for this process due to its support for lookup-tables and interpolation. A performance model of a circuit design is a model that relates the design performance with design parameters. Having obtained the Pareto-points, all the optimal solutions and their parameters are stored in a data file which defines the optimal performance model for the design.

It is important to consider process variation as early as possible in the design flow. Such variations can cause a circuit's performance to vary from their nominal point, reducing overall yield. This is a very important step in hierarchical-based design for yield prediction. The specifications in hierarchical design are given for the system level, however the yield of the system is influenced by the variations in the sub-block circuits. Therefore the performance spread of the sub-block circuit needs to be predicted and the yield of the whole system optimised. Monte Carlo (MC) analysis is the best candidate for this purpose. Therefore, during this step, a MC analysis is run for each of the parameter solution sets that lies on the Pareto-front. From this simulation, a set of performance spreads is obtained. The performance spread information is stored together with the performance model in a datafile.

3.4 Lookup Table Model Development

This step in the proposed architecture involves developing a behavioural model description for the circuit block so that it can be used in system level optimisation. The performance and variation model developed in the previous stage will be defined in a behavioural language and will be added in the behavioural description of the circuit block. The performance and variation model is defined as a look-up table using the table model function in Verilog-A. This function allows the module to approximate the behaviour of a system by interpolating between the performance and variation data points extracted from the Pareto-front. The syntax of the table model function is shown below:

```
$table_model(f1,f2,"datafile.tbl", control_string);
```

Where f1 and f2 are the performance functions, 'datafile.tbl' is the text file that contains the performance functions and design parameters, and 'control string' determines the interpolation and extrapolation method. In this algorithm, a cubic spline method is used for the interpolation and no extrapolation method is used, in order to avoid approximation of the data beyond the sampled data points. The algorithm creates a table model function for both performance and variation functions.

4 Experimental Results

This section presents a complete design example using a voltage control oscillator (VCO) as a test case for the performance and variation model development. This model is used to demonstrate a behavioural modelling infrastructure to design a system level PLL using hierarchical-based design. A block diagram of a PLL is shown in figure 5. The PLL is designed to operate with an output frequency range of 500MHz to 1.2GHz and a locking time and current consumption of less than 1 μ s and 15mA respectively. Jitter will be minimised as part of the process. All the following simulations were performed using the industry standard Cadence SpectreRF™ simulator with foundry level BSim3v3 transistor models from a standard 0.12 μ m process.

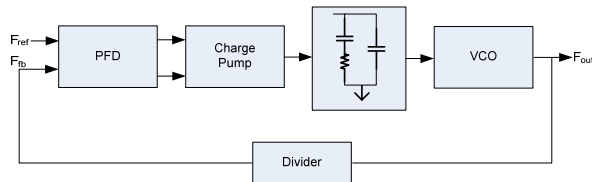


Figure 5. PLL system block diagram

4.1 Design Setup

The chosen VCO topology is a 5 stage ring oscillator as shown in figure 6. The first step is to determine the

designable parameters for the topology. In this example, these include the transistor lengths and widths making a total of 7 designable parameters. The performance functions for which the Pareto front must be generated are jitter, current consumption, gain, minimum frequency and maximum frequency. A testbench netlist was created to evaluate the performance functions.

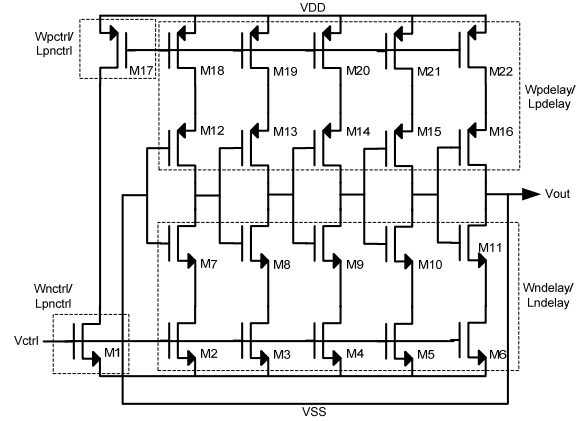


Figure 6. Five stage VCO schematic.

4.2 Multi-Objective Optimisation

The designable parameters must be constrained within a reasonable range which defines the design space of the optimisation. In this case all transistor lengths and widths were specified to be between 0.12 μ m-1 μ m and 10 μ -100 μ m. A Genetic Algorithm then generates the design parameters according to the specified design parameters and it is these that are used in the Spice netlist for simulation. A total of 30 generations each with a population size of 100 were used in this case, giving 3,000 total samples for the optimisation.

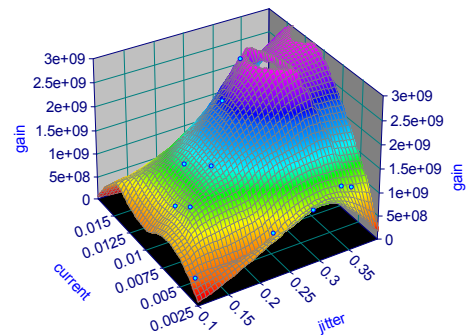


Figure 7. 3D plot of Pareto-optimal front.

The testbench netlist is used to evaluate the performance for each design parameter set (defined by GA) and the result of the simulations determines the fitness score of the individuals. Non-dominated sorting

and crowding distance sorting are applied to the solution for each generation in order to determine the final set of Pareto-fronts. The result of the optimisation is a full set of designable parameters and their performance functions.

4.3 Performance and Variation Modelling

The outcome of the MOO is a Pareto-front consisting of solution points which defines the performance model of the circuit. Figure 7 shows the 3D plot of the Pareto-front for three competing objectives: jitter, current and gain. To develop the variation model of the Pareto-front, every optimal solution undergoes a Monte Carlo simulation using foundry variation and mismatch models. 100 samples were chosen for the MC simulation and from these the variation for each performance is calculated. This completes the variation model and results are stored in a data file. At this point, a combined performance and variation model for the VCO is developed. A look-up table is defined for the table model function in the Verilog-A model given in listing 1. Table 1 shows a selection of samples points from the table and their performance and variation values.

Listing 1. Performance and Variation Model

```
analog begin
kvco_delta = $table_model(kvco, "kvco_delta.tbl", "3E");
jvco_delta = $table_model(jvco, "jvco_delta.tbl", "3E");
ivco_delta = $table_model(ivco, "ivco_delta.tbl", "3E");
fmin_delta = $table_model(fmin, "fmin_delta.tbl", "3E");
fmax_delta = $table_model(fmax, "fmax_delta.tbl", "3E");
.....
p1 = $table_model(kvco, ivco, jvco, fmin, fmax,
"p1_data.tbl", "3E,3E,3E,3E,3E");
p2 = $table_model(kvco, ivco, jvco, fmin, fmax,
"p2_data.tbl", "3E,3E,3E,3E,3E");
P7 = $table_model(kvco, ivco, jvco, fmin, fmax,
"p7_data.tbl", "3E,3E,3E,3E,3E");
fptr=$fopen("params.dat");
$fwrite(fptr, "\n Generated Design Parameters\n");
$fwrite(fptr, "%e %e %e %e %e %e", p1,p2,p3,p4,p5,p6,p7);
$fclose(fptr);
.....
end
```

Design:	Kvco (Mhz/V):	ΔKvco:	Jvco (ps):	ΔJvco:	Ivco (mA):	ΔIvco
20	997	0.50%	0.13	22%	8.62	2.9%
21	373	0.45%	0.11	22%	3.58	2.7%
22	1090	0.32%	0.29	25%	2.79	2.6%
23	1620	0.30%	0.19	23%	8.46	2.9%
24	2280	0.28%	0.36	26%	4.98	2.7%
27	1850	0.29%	0.21	23%	6.74	2.8%
28	1450	0.29%	0.12	22%	6.16	2.8%
29	1600	0.35%	0.30	25%	2.68	2.6%

Table 1. Performance and variation values.

4.4 Behavioural Description

The resulting performance and variation model become a part of the behavioural model of the VCO block. This model is given in listing 2 which includes the lookup-table of the Pareto-front. All the individual blocks in a PLL system including the PFD, CP and VCO were behaviourally modelled based on [13].

Listing 2. VCO behavioural model

```
module vco(out, outmin, outmax, in);
.....
parameter real kvco=500e6;
parameter real ivco=3e-3;
.....
analog begin
kvco_min = kvco - ((kvco_delta/100)*kvco);
kvco_max = kvco + ((kvco_delta/100)*kvco);
.....
jvco = $table_model(kvco, ivco, "data.tbl", "3E, 3E");
jvco_min = $table_model(kvco_min, ivco_min, "data.tbl", "3E, 3E");
jvco_max = $table_model(kvco_max, ivco_max, "data.tbl", "3E, 3E");
.....
delta = jvco*sqrt(2*ratio));
delta_min = jvco_min*sqrt(2*ratio));
.....
@(cross(phase-0.25,+1,tol))begin
dt = delta*$rdist_normal(seed, 0, 1);
dt_min = delta_min*$rdist_normal(seed,0,1);
.....
V(out)<+transition(vout,0,tt);
V(outmin)<+transition(vout_min,0,tt);
V(outmax)<+transition(vout_max,0,tt);
end
```

4.5 Hierarchical optimisation

The behavioural models are combined together for PLL system level optimisation using the NSGA-II algorithm. The designable parameters for the optimisation are gain (Kvco) and current (Ivco) for the VCO and C1, C2 and R1 for the loop filter.

During the optimisation, the variation model of VCO is used to interpolate the minimum and maximum VCO gain and VCO current for each of their nominal values generated by GA. Based on these values, the performance model is then used to interpolate the nominal, minimum and maximum of the VCO jitter. Therefore the output of the VCO behavioural block which will include the nominal, minimum and maximum values will be used to determine the PLL performances. At the end, the optimised system level will have their nominal as well as minimum and maximum performances. Table 2 shows some of the optimal solution samples for the PLL system level optimisation. From the optimal solutions, a solution that meets the specifications and variations will be selected as the design solution (shown by the shaded area).

Design Parameters Kv = VCO gain (MHz/V), Iv = VCO current (mA)									Performance Parameters (PLL) Ltime= lock time (us), Jit = Jitter sum (ps), Curr = current (mA)						
Kv	Kvmin	Kvmax	Iv	Ivmin	Ivmax	C1	C2	R1	Lt	Jit	Jit_min	Jit_max	Curr	Curr_min	Curr_max
1540	1536	1545	16.1	15.68	16.52	2.1p	1.8p	2k	0.90	4.30	4.23	4.38	26.1	25.6	26.6
684	680.9	687.1	3.78	3.68	3.88	5p	2.3p	3.8k	0.82	4.24	4.18	4.30	13.78	13.68	13.88
1050	1046	1053	3.96	3.95	3.97	3p	1p	1k	0.85	4.37	4.27	4.66	13.96	13.85	14.07
1480	1475.7	1484.3	10.4	10.1	10.7	4.8p	3p	2k	0.79	4.17	4.13	4.21	20.4	20.10	20.70

Table 2. PLL system level solution samples

To verify the predicted yield given by the proposed approach, a Monte Carlo analysis with 500 samples was run on the final design. This analysis confirmed a yield of 100%. Figure 8 shows the transistor level result for the PLL locking time simulation.

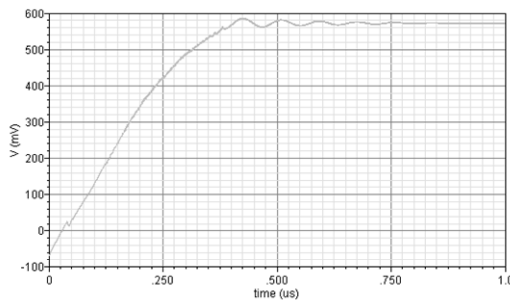


Figure 8. PLL locking time simulation plot

5 Conclusions

This paper has presented a new approach for hierarchical-based design that combines performance and yield optimisation for a mixed-signal system. Multi-objective optimisation based on an evolutionary algorithm is used to explore tradeoffs between performance and yield, leading to a set of Pareto optimal solutions for the design. Monte Carlo variation analysis is performed on all the Pareto optimal solutions, and a table is constructed for both the performance and variation analysis. A behavioural model developed in Verilog-A is used together with this table to determine the parameters required to achieve the highest yield within a given specification. The behavioural model is used for a system level simulation and the approach demonstrates a successful top-down optimisation. These benefits are enjoyed without a corresponding drop in accuracy. A 5 stage VCO and PLL system design were used to demonstrate the proposed algorithm and the behaviour has been verified with transistor level simulations.

References

- [1] B.De Smedt and G.Gielen, "Watson: design space boundary exploration and model generation for analogue and rfic

design," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transaction on*, vol.22, no.2, pp. 213-224, Feb. 2003.

- [2] Stehr G., Graeb H., and Antreich K., "Performance trade-off analysis of analog circuits by normal-boundary intersection.," in *Proc. Of Design Automation Conferenc 2003*, pp. 958-963.
- [3] M. Krasnicki, R.Phelps, J.R.Hellums, M. McClung, R.A Rutenbar and L. Richard Carley, "ASF: a practical simulation-based methodology for the synthesis of custom analogue circuits," in *Proc. ICCAD 2001*, pp 350-357.
- [4] B. D. Smedt, G. Gielen, "HOLMES: Capturing the yield-optimized design space boundaries of analogue and RF Integrated Circuits." In *Proc. Of the Design, Automation and Test in Europe Conference and Exhibition*, 2003.
- [5] Tiwary S.K., R.R.M.T., "Pareto Optimal Modeling for Efficient PLL Optimization," in *Technical Proceeding, 2004 NSTI Nanotechnology conference and Trade Show*, 2004.
- [6] J. Zou, D. Mueller, H. Graeb, U. Schlichtmann, "A cppll hierarchical optimization methodology considering jitter, power and locking time," in *Proc. 43rd ACM/IEEE Design Automation Conference*, 24-28 July 2006, pp. 19-24.
- [7] J. Zou, D. Mueller, H. Graeb, U. Schlichtmann, E. Hennig and R. Sommer, "Pareto-front computation and automatic sizing of cppll," in *Proc. 8th International Symposium on Quality Electronic Design ISQED '07*, 26-28 March 2007, pp. 481-486.
- [8] H. Liu, A. Singhee, R. Rutenbar and L. Carley, "Remembrance of circuits past: macromodeling by data mining in large analog design spaces," in *Design Automation Conference, 2002. Proceedings. 39th*, 10-14 June 2002, pp. 437-442.
- [9] Shao, J. and Harjani, R. "Macromodeling of analog circuits for hierarchical circuit design," In *Proc. Of 1994 IEEE/ACM International Conference on Computer Aided Design*, pp. 656-663.
- [10] S.Ali, R. Wilcock, and P. Wilson, "Behavioural performance and variation modelling for hierarchical-based analog integrated design," in *Proc. IEEE International Behavioural Modeling and Simulation Workshop BMAS 2008 (In Press)*.
- [11] T.Eeckelaert, T. McConaghy and G. Gielen, "Efficient multiobjective synthesis of analog circuits using hierarchical pareto-optimal performance hypersurfaces," in *Proc. Design, Automation and Test in Europe*, 2005, pp. 1070-1075.
- [12] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons Ltd, 2001.
- [13] K. Kundert, "Predicting the phase noise and jitter of PLL based frequency synthesizers," <http://www.designer-guide.com>