Hierarchical Synthesis System with Hybrid DLO-MOGA Optimization

ABSTRACT:

Purpose- The purpose of this paper is to present a hierarchical circuit synthesis system with a Hybrid DLO-MOGA *(Deterministic Local Optimization - Multi-Objective Genetic Algorithm)* optimization scheme for system level synthesis.

Design/methodology/approach- The use of a local optimization with a deterministic algorithm based on linear equations which is computationally efficient and improves the feasibility of designs, allows reduction in the number of MOGA generations required to achieve convergence.

Findings- This approach reduces the total number of simulation iterations required for optimization. Reduction in run-time enables use of full transistor level models for simulation of critical system level sub-blocks. Consequently for system level synthesis, simulation accuracy is maintained. The approach is demonstrated for the design of pipeline ADCs on a 0.35μm process.

Originality/value- The use of a Hybrid DLO-MOGA optimization approach is a new approach to improve hierarchical circuit synthesis time while preserving accuracy**.**

Paper Type- Research Paper

1 Introduction & Motivation

System level design automation of analogue circuits remains an important challenge for the semiconductor industry (International Technology Roadmap for Semiconductors website 2007). Traditional manual Top-Down Constrained Design (TDCD) methodologies for analogue circuits (Figure 1a) require significant designer expertise. An unknown number of design iterations may arise due to lack of knowledge of potential sub-block performance limitations. This process will achieve a finalized design that may not be optimal in terms of performance or power consumption. Advances in design automation techniques provide methodologies to ensure the design of optimal analogue circuits and systems (Gielen 2005; Rutenbar 2007). The primary considerations in the construction of a design automation system are, the choice of simulator for both circuit and system level and secondly the method of traversal of the design space at the system level. Firstly considering simulation: a trade-off exists between simulation run time and accuracy. At the sub-block level early approaches were limited by simulator run-time, therefore techniques such as simple design equations (Koh 1990) or parametric approaches to simulation (Ochotta 1994) were utilized. Modern computational power enables the use of high accuracy SPICE simulation for flat optimization of small designs or sub-blocks (Palmers 2009). For simulation at the system level particularly for large designs, behavioral or other abstract modeling of the sub-blocks may be used to improve simulation speed (Eeckelaert 2006; Yu-Tsun 2005). The use however of behavioral modeling at the system level presents a drawback that prevents widespread adoption of analogue synthesis system methodologies. Extraction or development of sub-block models (Rutenbar 2007; Jancke 2006) may be difficult or too time consuming for many designers to adopt new design methodologies. Furthermore simplistic models may fail to adequately capture sub-block characteristics leading to poor results when final verification is completed with transistor level simulation.

The second key requirement of system level design automation, i.e. traversal of the design space is now considered. Typically run-time will prevent optimization of a complete design as a flat-hierarchy. As in the sub-block case, the use of an equation-based method can be used to achieve a flat-hierarchy optimization, (Hershenson 2002) proposes the use of formulation

of specifications as posynomials and obtains solutions using geometric programming. A hierarchical approach to design automation involves breaking the complete design into sub-blocks and traversal of the design spaces in a hierarchical manner. The previously mentioned TDCD methodology (Figure 1a) may also be automated (Chang 1997). This approach is limited by the fact that produced designs are feasible but not optimal. The TDCD approach can be improved with the addition of bottom-up feasibility modeling which provides information on the design space of sub-blocks in advance of top-down optimization. Recent development of sub-block level synthesis using multi-objective optimization allows determination of the Pareto trade-off information of the sub-blocks (De Smedt 2003; Stehr 2003). SPICE level simulation is utilized at the sub-block level to ensure accurate determination of sub-block performance. This approach offers improved feasibility modeling and reduced computational cost by avoiding poor performing sub-block designs. These Pareto-optimal surfaces can be used to enable a Multi-Objective Bottom-Up Methodology (MOBU), detailed in Figure 1b, where selected Pareto-optimal designs become variables in the design space (Eeckelaert 2005). Design space exploration of the system level can be also obtained by application of MOGA optimization techniques. It is noted in Figure 1b, that the sub-block performance is determined by technology constraints leading to determination of Pareto-optimal sub-block designs and ultimately to an optimal system. This is in contrast to the TDCD approach where initial specifications for the system level or inherited specifications lead to a feasible design.

This paper presents a hierarchical synthesis system with a Hybrid DLO-MOGA at the system level. The Local optimization approach reduces the number of MOGA iterations required to optimize the system level. This reduction makes the use of full transistor level models for critical circuits at the system level affordable for practical synthesis. Selected circuits may be modeled at the transistor level where accuracy is important or extraction of a behavioral model is difficult. The approach is demonstrated for the design of pipeline ADCs on a 0.35µm process. The remainder of the paper is organized as follows: Section 2 details the proposed methodology, Section 3 outlines application of the technique to hierarchical synthesis as of a pipeline ADC, and Section 4 presents results of the synthesis process. Finally Section 5 details a summary.

2 Methodology

2.1 Hierarchical Synthesis System with Hybrid-MOGA

Hybrid MOGA

A hybrid MOGA or memetic algorithm (Moscato 1989), is formed from the combination of the genetic algorithm with a local search heuristic or algorithm (Cheng 1999). The combination of multi-objective genetic algorithm and local search was first demonstrated in (Ishibuchi 1998). Figure 2a details the generic framework of the hybrid – MOGA used in this work, with the genetic operations occurring after improvement of the initial population by local search (Ishibuchi 2002). Application of the hybrid MOGA to the system synthesis problem has two primary advantages. Firstly the computationally efficiency of the deterministic local search is much higher than the evaluation phase of the genetic algorithm, saving computation time. As mentioned in the introduction this is a key concern in system level synthesis where large, complex circuits lead to slow evaluation time. Secondly the local search employed can be utilized to ensure more feasible solutions (El-Mihoub 2006). In the case of the example shown in Section 3 for a pipeline ADC, the local search uses a deterministic algorithm to select appropriate values for the amplifier sub-blocks based on the resolution and capacitor scalings determined by the genetic search. This results in feasible solutions compared to the standard approach (Eeckelaert 2006) where the selection of amplifier sub-blocks is determined as a simple system variable in the genetic search.

Hierarchical Synthesis System

The proposed hierarchical synthesis system is detailed in Figure 3. The synthesis system utilizes a combined bottom-up/top-down methodology. The optimization procedure takes place in two steps. Initially circuit level sub-blocks (e.g. amplifier) are optimized using a MOGA implemented in MATLAB. The Cadence Spectre (Virtuoso Spectre Circuit Simulator website) simulator is used to accurately determine the objectives e.g. (Power, Open-loop Gain and Gain-Bandwidth). At the end of the sub-block synthesis process, the resulting performance data are stored and made available for the higher level system synthesis. At the completion of circuit level synthesis, the system level synthesis process then takes place.

In a standard hierarchical synthesis system the vector of design variables for the system level is comprised of variables specific to the system level circuit. Choices for the sub-blocks are also included as design variables. The i^{th} system level design variable vector will have a form

$$\overrightarrow{XS_i} = \left[x_{i,1}, x_{i,2}, .., x_{i,j}, .., x_{i,n}, s_{i,1}, s_{i,2}, .., s_{i,k}, .., s_{i,m}\right] \quad (1)$$

where $x_{i,j}$ are n system level design variables and $s_{i,k}$ are m sub-block selection variables.

This approach greatly reduces the number of possible states, as opposed to a fully flat synthesis procedure which leads to a large vector with all system level and sub-block design variables. As detailed in (Gielen 2005), since the sub-block states are Pareto-optimal the Pareto-optimal solutions of the system level will also be achieved.

The application of MOGA techniques to circuit and in particular system level design is limited by a trade-off in simulation accuracy and total run-time of the algorithm. For synthesis of a complete system level, the simulation time becomes long due to complex designs. Use of behavioral type simulators or models will reduce simulation time required to evaluate each design but also decrease simulation accuracy, possibly leading to inaccurate final results. To obtain good accuracy and synthesis results with behavioral synthesis, accurate behavioral models are required. Extraction of accurate behavioral models may be difficult or time consuming particularly for non-linear and fast settling blocks.

In this work the use of a standard MOGA for system level synthesis is replaced with a hybrid DLO-MOGA. As mentioned previously this has the result of providing improved computational efficiency and feasibility. This approach reduces the number of iterations required to reach convergence and enables the use of high quality transistor level models for critical circuits. In this work full transistor level models have been used for fast settling amplifier circuits for system level synthesis. The DLO used in this work is based on linear equations describing circuit behavior. It is noted that this equation-based approach is used to improve feasibility of solutions before the genetic search but is not used for actual evaluation of the performance of the population of designs therefore simulation accuracy is preserved. The disadvantage of the adoption of such a deterministic approach in a hierarchical synthesis system is the setup time

in developing the system of equations. The local optimization for circuit synthesis is not however limited to a strictly deterministic equation based approach and may be extended to other methods of local optimization that also are fast to compute and improve feasibility. The generalized hybrid MOGA and deterministic local optimization algorithm is detailed in the following section.

2.2 Theoretical Framework: Deterministic Local Optimization Process

Consider a population of N designs represented by the matrix X, where the general optimization problem may be defined as:

Determine the vector $\overrightarrow{X_i}$ of design variables of length n, such that the constraint function $g_\alpha\left(\overrightarrow{X_i}\right)$ of length u, is satisfied and the vector function $\vec{f}_\beta\left(\overrightarrow{X_i}\right)$ of $\beta \in \{1,..,v\}$ objectives is minimized,

$$\overrightarrow{X_i} = \left[x_{i,1}, x_{i,2},.., x_{i,j},.., x_{i,n}\right], \qquad i \in \{1,.., N\} \tag{2}$$

$$g_\alpha\left(\overrightarrow{X_i}\right) \leq 0, \qquad \alpha \in \{1,.., u\} \tag{3}$$

$$F\left(\overrightarrow{X_i}\right) = \left[f_1\left(\overrightarrow{X_i}\right), f_2\left(\overrightarrow{X_i}\right),.., f_\beta\left(\overrightarrow{X_i}\right),.., f_v\left(\overrightarrow{X_i}\right)\right], \qquad \beta \in \{1,.., v\} \tag{4}$$

It is noted that it is almost never the case that a single vector will cause all the functions to be minimized (and/or maximized) simultaneously. In the design of real world circuits it is necessary to determine criterion to define an optimum solution. The most common definition of comparing trade-offs between input vectors is known as Pareto Optimality (Coello-Coello 2002):

A vector $\overrightarrow{X_p}$ is Pareto-optimal if for every $\overrightarrow{X_i}$ either

$$\forall_{\beta=\{1,..,v\}} \quad f_\beta\left(\overrightarrow{X_i}\right) \geq f_\beta\left(\overrightarrow{X_p}\right), \qquad i \in \{1,.., N\} \tag{5}$$

and there is at least one $\beta \in \{1,.., v\}$ such that

$$f_\beta\left(\overrightarrow{X_i}\right) > f_\beta\left(\overrightarrow{X_p}\right) \tag{6}$$

The use of MOGA techniques provides a multi-objective optimization method for analogue circuit design (De Smedt 2003). The basic genetic algorithm operates on a population of design variable vectors $\overrightarrow{X_i}$ after each evaluation step. The best performing design vectors are selected for recombination. In this manner (Fonseca 1993), the best performing design variables are retained within the population and the objective space is traversed towards a maxima or minima. In the full MOGA approach a multiplicity of solutions are retained within the population. Therefore no weighting of the objectives is required prior to optimization to force convergence to a particular result. The most suitable trade-off solution may be chosen after the algorithm has completed.

For the system level optimization the MOGA is hybridized with a local optimization stage (Ishibuchi 2002; Harada 2006). The local optimization is considered first for a general case. Consider a population of vectors containing the design variables of the system,

$$\overrightarrow{X}_i = \left[x_{i,1}, x_{i,2}, .., x_{i,j}, .., x_{i,n}\right], \qquad i \in \{1,..,N\} \tag{7}$$

The local optimization process (Figure 2a) applied prior to MOGA, acts to optimize the design variables of each vector in the population producing an optimized vector $\overrightarrow{X_i^O}$ where,

$$\overrightarrow{X_i^O} = LO\left(\overrightarrow{X_i}\right), \qquad i \in \{1,..N\} \tag{8}$$

and LO is the local optimization process. The effect of the process on the design vector is given in eq. (9). Note that the optimization process may operate on one or more of the design variables.

$$\overrightarrow{X_i} = \left[x_{i,1}, x_{i,2}, .., x_{i,j}, .., x_{i,n}\right] \rightarrow \overrightarrow{X_i^0} = \left[x_{i,1}, x^0{}_{i,2}, .., x^0_{i,j} ..... x_{i,n}\right], \qquad i \in \{1,..,N\} \tag{9}$$

The design variables marked by ^{0} have changed values. The effect of the local optimization process is graphically shown in Figure 4a. The initial states from the initial population or the last MOGA iteration are denoted by the solid circles. The local optimization process maps the individual vectors to new improved states and defines a new Pareto-optimal vector in the objective space F as defined in eq. (4). If the computationally efficiency of the local

optimization process is high compared to the MOGA then the overall algorithm will converge more rapidly than a standard MOGA (Ishibuchi 2002).

Now consider the case of the system design variables vector for use in analogue circuit synthesis. As detailed in Section 2.1 the vector for the system level synthesis is defined by

$$\overrightarrow{XS_i} = \left[x_{i,1}, x_{i,2}, .., x_{i,j}, .., x_{i,n}, s_{i,1}, s_{i,2}, .., s_{i,k}, .., s_{i,m}\right] \tag{10}$$

and contains system level variables $x_{i,j}$ and variables corresponding to the sub-block selections $s_{i,k}$. In the case of analogue circuit design, it is possible to optimize the choice $s_{i,k}$ for the system through a deterministic process. This is process is written as

$$\overrightarrow{XS_i^0} = D\left(\overrightarrow{XS_i}\right), \qquad i \in \{1,..N\} \tag{11}$$

A deterministic optimization process may be advantageous as prior knowledge of the likely specification of sub-blocks is often well known through the use of standard design methodologies (Goes 1998). From calculation of ideal specifications a mapping can occur between ideal specification and available optimized sub-blocks, choosing the closest match and improving the feasibility of the design. An example of how this process can be performed for the case of a pipeline ADC is shown in Section 3. Note that since specific sub-blocks are to be optimized independently, it is not necessary for the system level MOGA to contain a variable for these. Therefore a reduced chromosome vector is sufficient for the case of hierarchical analogue system level synthesis.

$$\overrightarrow{X_i^R} = \left[x_{i,1}, x_{i,2}, .., x_{i,j}, .., x_{i,n}\right], \qquad i \in \{1,.., N\} \tag{12}$$

An additional vector detailing the optimized sub-blocks is produced in the deterministic local optimization, where

$$\overrightarrow{S_i^O} = DLO\left(\overrightarrow{X_i^R}\right), \qquad i \in \{1,..N\} \tag{13}$$

This vector, which is the result of the local optimization process, is given in eq. (14) and is used in creating the netlist prior to evaluation in the MOGA step.

$$\overrightarrow{S_i^0} = \left[s_{i,1}^0, s_{i,2}^0, .., s_{i,k}^0, .., s_{i,m}^0 \right], \qquad i \in \{1, ..N\} \tag{14}$$

The deterministic local optimization process hybridized with the MOGA for circuit synthesis is shown in Figure 2b. The system level design vectors are also included in this figure. The effect of the optimization process by using a reduced chromosome is shown conceptually in Figure 4b. In contrast to Figure 4a where the exact position of each design in objective space is known prior to local optimization, the pre-optimized design before DLO occupies a region of probability defined by all possible sub-block space variables. It is then mapped to a fixed point by the optimized process. The objective space values are realized during evaluation.

3 Specific Example Pipeline ADC

3.1 Pipeline ADC

The design problem associated with the pipeline ADCs (Figure 5) is the selection of the appropriate number of stages, bits resolved in each stage and capacitor size in each stage to achieve the lowest power consumption for a given converter resolution and sample rate. In designing the optimum pipeline ADC there are 2 degrees of freedom, Resolution Scaling and Capacitor Scaling:

Resolution Scaling: The number of bits resolved in each stage is chosen. The more bits resolved in each stage the higher the closed-loop stage gain and effective load capacitance, this tends to increase power consumption within the stage. However the high stage gain reduces the required residue voltage signal (V_{RES}) accuracy at the output of the stage.

Capacitor Scaling: Minimizing capacitor sizes reduces total amplifier power consumption in a stage. The minimum capacitor size will be limited by mismatch or thermal noise considerations in the early stages where the input-referred noise contribution is greatest. The minimum capacitor size is limited by parasitic capacitance in the later stages.

In the conventional manual design of pipeline ADCs for lowest power, a strict top down hierarchical design process is followed (Goes 1998). The top level architecture is first determined by choosing the resolution and capacitor scaling to minimize power consumption. Secondly sub-block specifications are defined. Several assumptions are made in this design process: Amplifiers are well designed and do not introduce settling errors. The effect of comparator errors are eliminated by the digital error correction scheme. It is assumed that the thermal noise is the lower bound error source. The selection of the optimal combination of resolution scaling and capacitance scaling is difficult to determine in a single pass manual top-down hierarchical design methodology (Figure 1a). The use of a hierarchical synthesis approach allows the achievable performance of sub-blocks in a given technology to be considered during the system level synthesis phase.

3.2 System Level Synthesis with Hybrid DLO-MOGA

The system level synthesis procedure is detailed in Figure 6. As detailed in Section 2.1 the process is composed of a hybrid DLO-MOGA optimization algorithm. An initial population is created or updated and applied to the DLO. The optimized vectors of the population are then transferred to the MOGA for evaluation. Translation of the vectors to a form suitable for netlisting occurs before the Spectre simulator can be invoked. The remaining MOGA steps are the standard steps of: Pareto rank, Linear Selection and Crossover based Recombination.

3.3 Representation of System level Chromosome

As detailed in Section 2 deterministic local optimization may be used to determine the appropriate choice of sub-blocks in synthesis of a top level design, improving feasibility of the designs. In the case of the pipeline ADC there are three key sub-blocks: stage amplifiers, switches and comparators. In this work deterministic optimization is used to optimize amplifiers within each stage of the pipeline. The amplifiers are modeled using full transistor models in the Spectre simulator. Simple behavioral models with short development time are used for the comparators and switches to increase simulation speed.

The structure of the pipeline ADC is encapsulated into a system level chromosome that may be processed by the MOGA. In this work the system-level pipeline ADC is represented by two independent chromosomes. The two chromosomes are comprised of a series of scaling factors eq. (15,16), where the resolution or capacitance at each stage may be determined by the cumulative product of the chromosome vector.

$$\overrightarrow{XR_i^R} = \left[xr_{i,1}, xr_{i,2}, .., xr_{i,j}, .., xr_{i,n}\right], \qquad i \in \{1,..N\} \tag{15}$$

$$\overrightarrow{XC_i^R} = \left[xc_{i,1}, xc_{i,2}, .., xc_{i,j}, .., xc_{i,n}\right], \qquad i \in \{1,..N\} \tag{16}$$

where $xr_{i,j}$ is the scaling factor for stage resolution, $xc_{i,j}$ is the scaling factor for capacitance and $i \in \{1,..N\}$, $j \in \{1,..n\}$ where N is the number of designs in the population and n is the length of the chromosome vector.

The chromosomes describe the state of each design in the pipeline with the exception of amplifiers which are determined in the deterministic local optimization process (Section 3.4). Eq. (17) defines $\overrightarrow{S_i^0}$ the vector describing the choice of sub-blocks, which is determined through local optimization.

$$\overrightarrow{S_i^0} = \left[s_{i,1}^0, s_{i,2}^0, .., s_{i,k}^0 ..... s_{i,m}^0\right], \qquad i \in \{1,..N\} \tag{17}$$

where $s_{i,k}$ is the choice of amplifier sub-block.

In order to perform evaluation of the chromosomes the MOGA must interface to the simulator. This requires that chromosomes are transformed into a format that allows a netlist to be created. This step (denoted TL1) is performed prior to local optimization as some parameters are also used in the deterministic optimization process.

TL1: Transformation of chromosome $\overrightarrow{XR_i^R}, \overrightarrow{XC_i^R}$ into high-level architectural parameters. This includes the number of bits per stage and capacitor size in each stage, gain of each stage, number of comparators and capacitors. These are summarized in Table 1. The stage resolutions are quantized to integer values. The presence of 0.5-bit error correction in each stage is also assumed.

B_{MAX} and C_{UNIT} are constants that define the maximum allowable value of bits of resolution in a stage and maximum value of unit capacitance in a stage. Note that the final stage n is always set to a 1-bit ADC and therefore has no amplifier or sample and feedback capacitors.

The proposed approach is based on scaling factors as opposed to directly specifying the resolution and capacitance of each stage, has the advantage that a monotonically decreasing set of stage resolutions and capacitances is always obtained after the recombination process.

Table 1: Determination of Architectural Level Parameters.

Parameter	Design Equation
Bits resolved in each stage in the pipeline $\overrightarrow{BS_i}$	$bs_{i,j} = B_{MAX} \prod_{h=1}^{j} [xr_{i,h}]$
Unit Capacitance of each sample/feedback capacitor in stage $\overrightarrow{CU_i}$	$cu_{i,j} = C_{UNIT} \prod_{h=1}^{j} [xc_{i,h}]$
Gain of each stage $\overrightarrow{GS_i}$	$\overrightarrow{GS_i} = 2^{\overrightarrow{BS_i}}$
Number of comparators $\overrightarrow{NC_i}$, number sample and feedback capacitors $\overrightarrow{NCS_i}$, $\overrightarrow{NCF_i}$.	$ncs_{i,1} = 1, ncf_{i,1} = 1, nc_{i,1} = 0$ $ncs_{i,n} = 0, ncsf_{i,n} = 0\ nc_{i,n} = 1$ For $i < j < n$ $nc_{i,j} = 2^{bs_{i,j}-1}$ $ncs_{i,j} = nc_{i,j}$ $ncf_{i,j} = nc_{i,j} / gs_{i,j}$
Capacitor sizes $\overrightarrow{CS_i}$, $\overrightarrow{CF_i}$.	$\overrightarrow{CS_i} = \overrightarrow{NCS_i} \cdot \overrightarrow{CU_i}$ $\overrightarrow{CF_i} = \overrightarrow{NCF_i} \cdot \overrightarrow{CS_i}$

3.4 Deterministic Local Optimization

The deterministic local optimization algorithm is based on calculation of optimal amplifier values for each system level vector of the amplifier and is detailed in Figure 7. The process consists of two steps to produce the optimized sub-block vector $\overrightarrow{S_i^0}$. These steps are now detailed.

D1: Initially ideal specifications are calculated for the stage amplifiers to achieve the slew and settling of each stage to the required accuracy. This depends on the calculated stage load capacitance and position in the pipeline. Table 2 details the calculated amplifier ideal specifications for each design vector in the population.

The result of this first phase process is a set of vectors describing ideal amplifier performance. The vectors for amplifier sub-block are of length m as the last stage of the pipeline is composed of comparators only.

$$\overrightarrow{A_i} = \left[a_{i,1}, a_{i,2}, .., a_{i,k}, .., a_{i,m}\right], \qquad i \in \{1,..N\} \tag{18}$$

$$\overrightarrow{\Omega_i} = \left[\omega_{i,1}, \omega_{i,2}, .., \omega_{i,k}, .., \omega_{i,m}\right], \qquad i \in \{1,.., N\} \tag{19}$$

$$\overrightarrow{IS_i} = \left[is_{i,1}, is_{i,2}, .., is_{i,k}, .., is_{i,m}\right], \qquad i \in \{1,.., N\} \tag{20}$$

D2: The ideal circuit specifications for each stage of each design vector in the population are then compared against the Pareto performance data obtained from the low-level synthesis process. A best case design is selected based on its fit to the ideal specification.

In order to select this best fit design, firstly a Pareto rank of the M designs in the sub-block population performance data is performed (for Power, Open-loop Gain and Gain-Bandwidth). Candidate amplifier designs are filtered before ranking to eliminate those not meeting slew and phase margin constraints. The Pareto ranked vectors of performance data from the sub-block synthesis process are given for Open Loop Gain $\overrightarrow{PA}$, Gain-Bandwidth $\overrightarrow{P\Omega}$ and power consumption $\overrightarrow{PR}$:

Table 2: Determination of Ideal Amplifier Specifications.

Parameter	Design Equation
Stage Accuracy $\overrightarrow{\mathrm{K}_i}$ in Bits, where N_{Max} is a design parameter setting the maximum resolution.	$\kappa_{i,1} = N_{Max} - bs_{i,1}$ $\kappa_{i,k} = bs_{i,k-1} - bs_{i,k}$ for $k \le m$
Settling Accuracy $\overrightarrow{\mathrm{E}_i}$	$\varepsilon_{i,k} = \frac{1}{2^{2\kappa_{i,k}}}$ for $k \le m$
Amplifier Closed Loop Gain $\overrightarrow{AV_i}$	$av_{i,k} = \frac{cs_{i,k} + cf_{i,k}}{cf_{i,k}}$ for $k \le m$
Capacitive load due to successive stage $\overrightarrow{CL_i}$	$cl_{i,k} = cs_{i,k+1} + cn_{i,k-1}$ for $k \le m-1$ $cl_{i,m} = 0$
Total Capacitive load $\overrightarrow{CT_i}$	$ct_{i,k} = cl_{i,k} + cs_{i,k} + cf_{i,k} + cl_{i,k}\left[\frac{cs_{i,k}}{cf_{i,k}}\right]$ for $k \le m$
Amplifier Transconductance $\overrightarrow{GM_i}$	$gm_{i,k} \cong \frac{ct_{i,k} av_{i,k}}{t_s} \ln\left(\frac{1}{\varepsilon_{i,k}}\right)$ for $k \le m$ where t_s is the settling time to the required accuracy.
Gain Bandwidth $\overrightarrow{\Omega_i}$ rads^{-1}	$\omega_{i,k} = \frac{gm_{i,k}}{ct_{i,k}}$ for $k \le m$
Open loop Gain $\overrightarrow{A_i}$ in dB	$a_{i,j} = 20Log_{10}\left(2^{k_{i,j}}\right) \cdot av_{i,j}$ for $k \le m$
Minimum Slew Current $\overrightarrow{IS_i}$	$is_{i,k} = ct_{i,k} \cdot t_{SLEW}$ for $k \le m$, where t_{SLEW} is the available setting time.

$$\overrightarrow{PA} = [pa_1, pa_2, .., pa_l, .., pa_M], \qquad l \in \{1,..M\} \tag{21}$$

$$\overrightarrow{P\Omega} = [p\omega_1, p\omega_2, .., p\omega_l, .., p\omega_M], \qquad l \in \{1,..M\} \tag{22}$$

$$\overrightarrow{PR} = [pr_1, pr_2, .., pr_l, .., pr_M], \qquad l \in \{1,..M\} \tag{23}$$

The ideal amplifier specifications for Gain and Gain-Bandwidth for each pipeline stage are compared against the performance data of the sub-block population.

$$\delta a_{i,k,l} = pa_l - a_{i,k} \quad \forall \; i \in \{1,..N\}, \quad k \in \{1,..,m\}, \quad l \in \{1,..M\} \tag{24}$$

$$\delta \omega_{i,k,l} = p\omega_l - \omega_{i,k} \quad \forall \; i \in \{1,..N\}, \quad k \in \{1,..m\}, \quad l \in \{1,..M\} \tag{25}$$

where $\overrightarrow{\Delta A_l}, \overrightarrow{\Delta \Omega_l}$ are third order matrices defined by the comparison values $\delta a_{i,k,l}$ and $\delta \omega_{i,k,l}$.

$$\Delta A_l = \begin{pmatrix} \delta a_{1,1,l} & \delta a_{1,2,l} & \cdots & \delta a_{1,m,l} \\ \delta a_{2,1,l} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \delta a_{N,1,l} & \cdots & \cdots & \delta a_{N,m,l} \end{pmatrix} \tag{26}$$

$$\Delta \Omega_l = \begin{pmatrix} \delta \omega_{1,1,l} & \delta \omega_{1,2,l} & \cdots & \delta \omega_{1,m,l} \\ \delta \omega_{2,1,l} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \delta \omega_{N,1,l} & \cdots & \cdots & \delta \omega_{N,m,l} \end{pmatrix} \tag{27}$$

The amplifier design meeting the specification with the lowest Pareto rank is selected. This operation is written mathematically as:

$$s_{i,k} = \min(l): \left\{ (\delta a_{i,k,l} > 0) \middle| (\delta \omega_{i,k,l} > 0) \right\} \quad \forall \; i \in \{1,..N\}, \quad k \in \{1,..m\}, \quad l \in \{1,..M\} \tag{28}$$

where $s_{i,k}$ is an integer value given by the index of l that satisfies the above relation.

If the ideal specifications cannot be simultaneously satisfied as detailed in eq. (28), a target vector approach is applied to find the design with the closest Gain and Gain-Bandwidth product to the required specification:

$$s_{i,k} = (l): \min \left\{ w_1 \left| \delta a_{i,k,l} \right| + w_2 \left| \delta \omega_{i,k,l} \right| \right\} \quad \forall \; i \in \{1,..N\}, \quad k \in \{1,..m\}, \quad l \in \{1,..M\} \tag{29}$$

where w_1 and w_2 are weighting factors. Figure 8 details graphically the mapping of ideal specifications to synthesized designs for operational amplifiers. The application of Deterministic Local Optimization reduces the time required for convergence as the multi-objective genetic algorithm is not required to select the amplifier designs independently from the stage resolution and capacitor scaling.

4 Results

The Austria Micro Systems (AMS) 0.35μm technology is used for the hierarchical synthesis process. The sub-block and system level synthesis process for the design pipeline ADCs are now detailed.

The circuit level synthesis process (Figure 2) is used to synthesize folded cascode amplifiers used in each stage of the pipeline. Figure 9 details the parameterizable amplifier cell used for synthesis. The complete sizing for all transistors in the cell, is determined by a chromosome containing widths and lengths for bias transistors and the differential pair. Multiplier variables are used to scale the transistors of the differential and output stages. The amplifier sub-block synthesis is run for logarithmically spaced values of load capacitance. The designs are optimized for Power, Gain and Gain-Bandwidth. Phase Margin and Slew Rate may be set as constraints. The amplifier synthesis procedure required 12 generations with a population of 250 designs to arrive at a final Pareto-optimal solution. The result is a population of designs shown in Table 3, (e.g. Synthesized amplifier population size M, Pareto ranked for objectives in Power, Open-loop Gain and Gain-Bandwidth (GBW)).

As detailed in Section 3, the Pareto-optimal objective data is used for Deterministic Local Optimization during the system level synthesis. The complete sub-block synthesis requires 6 hours computation on a HP xw6400 workstation. The corresponding transistor sizes are also stored for transistor level simulation of completed designs.

The system level synthesis operates as outlined in Section 3 and Figure 6 utilizing a Hybrid DLO-MOGA, and Spectre simulation.

Table 3: Example of Synthesized Amplifier Sub-block.

Design No.	Objective $f_1\left(\overrightarrow{X_i}\right)$ Power (W)	Objective $f_2\left(\overrightarrow{X_i}\right)$ Gain (dB)	Objective $f_3\left(\overrightarrow{X_i}\right)$ GBW (rad s^{-1})	Constraint $g_1\left(\overrightarrow{X_i}\right)$ Slew Rate (V/s) >	Constraint $g_2\left(\overrightarrow{X_i}\right)$ Phase Margin > 60^0
1	1.2e-2	83	200e6	2e-3	75
2	4.3e-4	95	17e6	8.9e-6	74
⋮	⋮		⋮	⋮	⋮
M	5.8e-3	78	10e6	4.9e-4	89

The initial scaling factors for the resolution and capacitance chromosomes are generated randomly. Prior to obtaining objective values through simulation DLO occurs and optimal amplifier designs are selected. After objective values have been determined, ranking of the population can occur. In this work the Pareto ranking occurs based on simultaneously minimizing the SNR (Signal to Noise Ratio) and the total power consumption. Selection of designs for recombination utilizes a simple linear rank selection scheme, therefore the higher ranked designs have a greater probability of selection. A constraint based on goal attainment for thermal noise is included. Designs are considered feasible and Pareto ranked if measured SNR does not exceed a calculated limit due to thermal noise. Recombination of selected pairs of designs occurs with a crossover scheme.

The MOGA requires 12 generations $N_G = 12$ and contains a population of 200 designs. A simple AHDL behavioral model is used for the comparators and switches. The system level synthesis executes in 90 hours and evaluates 1400 designs (on HP xw6400 workstation). Simulation scripts and netlists for each design in the population are generated before being supplied to the Spectre simulator.

The system synthesis produces a final population of designs with optimized resolution and capacitor scalings for given SNR and power values at fixed sample rates (this work

$f_s = 2$ MS/s, 10MS/s). The results of the completed system level synthesis are detailed in Table 4. Power consumption and SNR for selected synthesized designs with the associated resolution scaling are shown.

It is observed from the Table 4 that the resolution scaling produced by the system level synthesis process for lower resolution (lower SNR) designs consists of cascaded 1.5-bits per stage. For higher resolutions, 2.5-bit or higher bit stages occur at the pipeline front end and in early stages. Due to the use of a bottom-up multi-objective methodology and the consequent availability of actual amplifier performance data in the system level synthesis process (detailed in Section 3), the system level synthesis behaves like an experienced designer by selecting resolution scalings with many bits at the front end to reduce amplifier gain and settling accuracy requirements. These selections allow for the finite Gain and Gain-Bandwidth limitations of the amplifier sub-blocks (Goes 1998). The maximum achievable resolution is limited by driving large capacitances imposed by thermal noise constraints. Power consumption is demonstrated to increase with higher resolution and faster sample rates.

Figure 10a details power consumption at each stage of the pipeline (designs 1-6), Figure 10b shows capacitor scaling at each stage in the pipeline (designs 1-6). It is noted that power consumption is dominated by early stages, this is due to the requirement for higher amplifier settling and Gain requirements in the early stages. The power consumption is often higher for the 2nd stage than the 1st sample and hold stage due to lower closed loop Gain in the sample and hold stage.

As detailed in Section 3, the system-level synthesis incorporates a Hybrid DLO-MOGA to reduce the number of iterations required to achieve convergence. To validate this approach a comparison is performed with the system-level synthesis process with a standard MOGA. Figure 11 details the results of the comparison after 7 generations and 1400 evaluations. It is noted that with the Hybrid DLO-MOGA approach that convergence is improved with far less sub-optimal designs within the population. Secondly, synthesized performance is greatly improved for lower resolution designs (50-65dB region).

Table 4: System Level Synthesis Results.

Design No.	SNR* (dB)	Power (mW)	Resolution Scaling (1^{st} 4 stages)
1	55	2.7	[1.5,1.5,1.5,1.5]
2	59	3.9	[2.5,1.5,1.5,1.5]
3	64	5.9	[2.5,2.5,2.5,1.5]
4	69	6.2	[3.5,2.5,1.5,1.5]
5	74	9.3	[3.5,2.5,2.5,1.5]
6	79	14	[3.5,3.5,3.5,1.5]
7	55	19	[1.5,1.5,1.5,1.5]
8	60	13	[2.5,1.5,1.5,1.5]
9	65	30	[2.5,2.5,1.5,1.5]
10	70	30	[3.5,2.5,1.5,1.5]
11	74	35	[2.5,2.5,2.5,2.5]

* Designs (1-6) $f_s = 2$ MS/s, Designs (7-11) $f_s = 10$ MS/s

5 Summary

This paper presents a hierarchical synthesis system with Hybrid DLO-MOGA at the system level. This approach reduces the number of iterations required to optimize the system-level. The DLO used in this work is based on linear equations describing circuit behavior and is used to improve feasibility of solutions before the genetic search. This reduction allows use of transistor level models for circuits where high simulation accuracy is required or the extraction of complex behavioral models is difficult. The disadvantage of adopting such a deterministic approach in a hierarchical synthesis system is the setup time in developing the system of equations for the DLO.

The method is demonstrated by the design of pipeline ADCs on a 0.35μm technology. The synthesis process produces resolution and capacitor scalings to minimize power at various output resolutions. The improvement produced by the Hybrid DLO-MOGA synthesis process

is quantified by demonstration of improved convergence for a given number of system level iterations compared to conventional hierarchical synthesis methodology for the same circuit.

The proposed methodology may be extended to other complex circuits and systems such as analogue-to-digital converters, oscillators and power converters. Future work will involve targeting the hierarchical synthesis at very low geometry processes <90nm where design automation is critical to achieve the required circuit performance, focusing on extending the approach to encompass other converter architectures. Other non-deterministic and computationally efficient local optimization algorithms could be investigated with the aim of ensuring robust synthesis and further reducing or eliminating the dependence on user generated behavioral equations.

References

Cadence Spectre Circuit Simulator website, http://www.cadence.com/products/rf/spectre_circuit/pages/default.aspx.

Coello-Coello, C. A. . van Veldhuizen, D. A, and Lamont, G. B. (2002), "Evolutionary Algorithms for Solving Multi-Objective Problems", Kluwer Academic Publishers, Netherlands.

Chang H. et al. (1997) "A Top–Down, Constraint–Driven Design Methodology for Analog Integrated Circuits". Kluwer Academic Publishers, 1997.

Cheng R. Gen, M. Tsujimura, Y. (1999) "A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies" *Computers & Industrial Engineering*, Volume 36, Issue 2, April 1999, Pages 343-364.

De Smedt, B. and Gielen, G.G.E. (2003) "WATSON: design space boundary exploration and model generation for analogue and RFIC design". *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. Vol. 22(2): pp. 213.

Eeckelaert, T. McConaghy, T. Gielen, G. (2005) "Efficient multiobjective synthesis of analog circuits using hierarchical Pareto-optimal performance hypersurfaces" *Design, Automation and Test in Europe, DATE 2005*. Proceedings 2005, Page(s): 1070 - 1075 Vol. 2

Eeckelaert, T. Schoofs, R. Gielen, G. Steyaert, M. Sansen, W. (2006), "Hierarchical bottom-up analogue optimization methodology validated by a delta-sigma A/D converter design for the 802.11a/b/g standard", *Proceedings 43rd ACM/IEEE Design Automation Conference 2006*, pp. 25–30

El-Mihoub, TA. Hopgood, A.A. Nolle, L. Battersby A. (2006) "Hybrid genetic algorithms: A review", *Engineering letters*, Vol 13, issue 2, 2006.

Fonseca, C.M. and Fleming, P. (1993) "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization*". Proceedings of the Fifth International Conference on Genetic Algorithms*, San Mateo, California, 1993, pp. 416–423.

Gielen, G.G.E. McConaghy, T. and Eeckelaer, T. (2005), "Performance space modeling for hierarchical synthesis of analogue integrated circuits," *Proceedings. 42nd Design Automation Conference*, 13-17 June 2005, pp. 881–886.

Goes, J. Vital, J.C., and Franca, (1998) J.E. "Systematic design for optimization of high-speed self-calibrated pipelined A/D converters". *Circuits and Systems II: Analogue and Digital Signal Processing, IEEE Transactions on*, Vol. 45, no 12, pp. 1513.

Harada, K. Sakuma, J. Kobayashi S. (2006) **"**Hybridization of genetic algorithm and local search in multiobjective function optimization: recommendation of GA then LS" *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 667 – 674**.**

Hershenson M. (2002) "Design of Pipeline Analogue-To-Digital Converters Via Geometric Programming". *International Conference on Computer Aided Design, ICCAD*, 2002, Page(s): 317 - 324

International Technology Roadmap for Semiconductors website (2007), http://www.itrs.net/Links/2007ITRS/Home2007.htm.

Ishibuchi, H. Murata, T. (1998) "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Trans. on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 28, no. 3, pp. 392-403, 1998.

Ishibuchi H., Yoshida Y. (2002), " Hybrid Evolutionary Multi-Objective Optimization Algorithms ", Abraham A. (Ed.), Design, Management and Applications, Frontiers in Artificial Intelligence and Applications, IOS Press, Netherlands, pp. 163-172

Jancke, R. and Schwarz, P. (2006) “Supporting analogue synthesis by abstracting circuit behavior using a modeling methodology”. ISCAS 2006, *Proceedings 2006 IEEE International Symposium on Circuits and Systems*. pp. –1474

Koh, H.Y., Sequin, C.H.; Gray, P.R. (1990) “OPASYN: a compiler for CMOS operational amplifiers” *Computer-Aided Design of Integrated Circuits and Systems*, IEEE Transactions on, 1990. Vol. 9(2): Page(s): 113 - 125

Moscato, P. (1989) "On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms". *Caltech Concurrent Computation Program* (report 826, 1989).

Ochotta, E.S. Rutenbar, R.A. Carley, L.R. (1994) “Analog circuit synthesis for large, realistic cells: designing a pipelined A/D converter with ASTRX/OBLX”*, Proceedings of the IEEE Custom Integrated Circuits Conference, 1994*. Vol 3, issue 15, pg. 273-294

Palmers, P. McConnaghy, T. Steyaert, M. Gielen, G.G.E. (2009), “Massively multi-topology sizing of analogue integrated circuits”. *Proceedings DATE '09, Design, Automation & Test in Europe Conference & Exhibition*, 20-24 April 2009, pp. 706 - 711

Rutenbar, R.A. Gielen, G.G.E. and Roychowdhury, J. (2007), “Hierarchical Modeling, Optimization, and Synthesis for System-Level Analogue and RF Designs”. *Proceedings of the IEEE*, 2007. Vol. 95(3), pp. 640

Stehr, G. Graeb, H. and Antreich, K. (2003), “Performance trade-off analysis of analogue circuits by normal-boundary intersection”. *Proceedings Design Automation Conference, 2-6 June 2003*, pp: 958–963.

Yu-Tsun, C. Dong, C. Jea-Hong, L. Gin-Kou, M. Rutenbar, R.A. Mukherjee, T. (2005), “Designer-driven topology optimization for pipelined analogue to digital converters”. Proceedings DATE '05 Design, Automation and Test in Europe, 2005, Vol. 1, pp. 279–280.

Figures

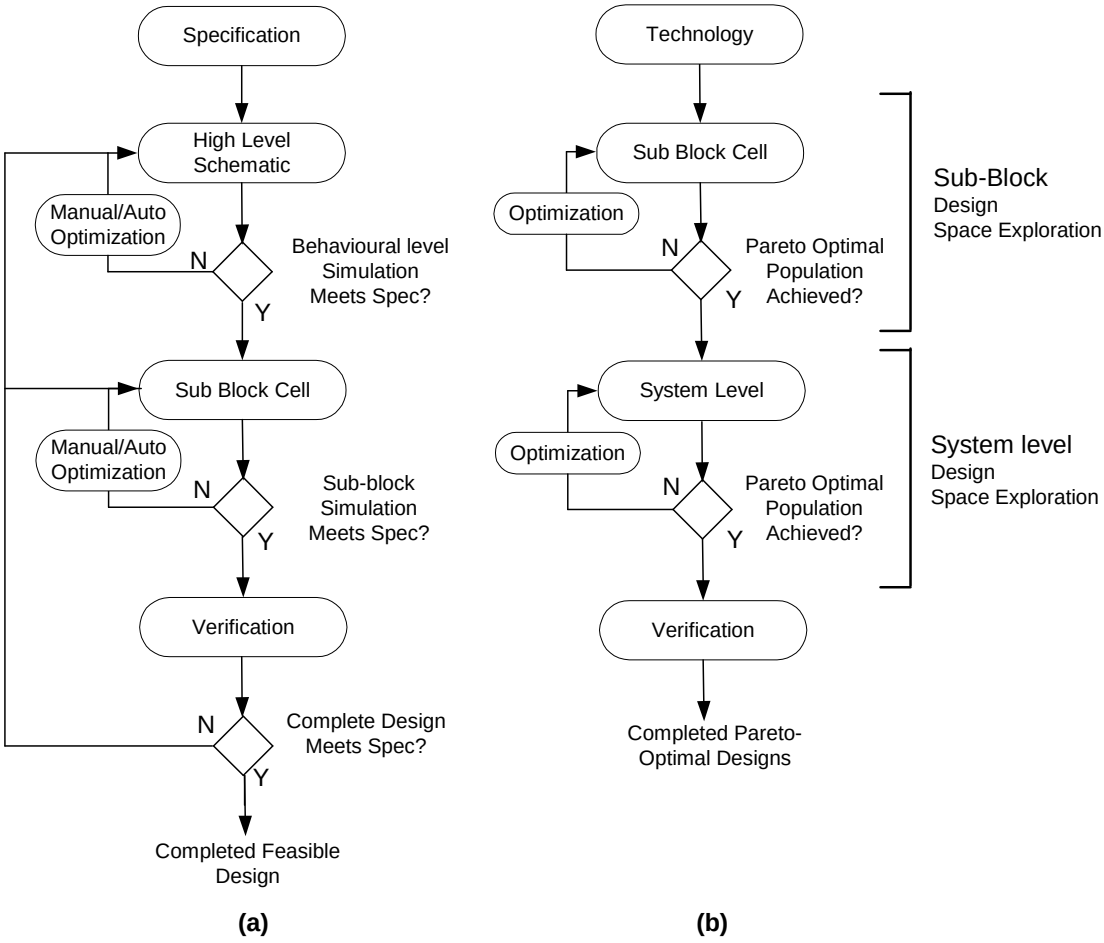
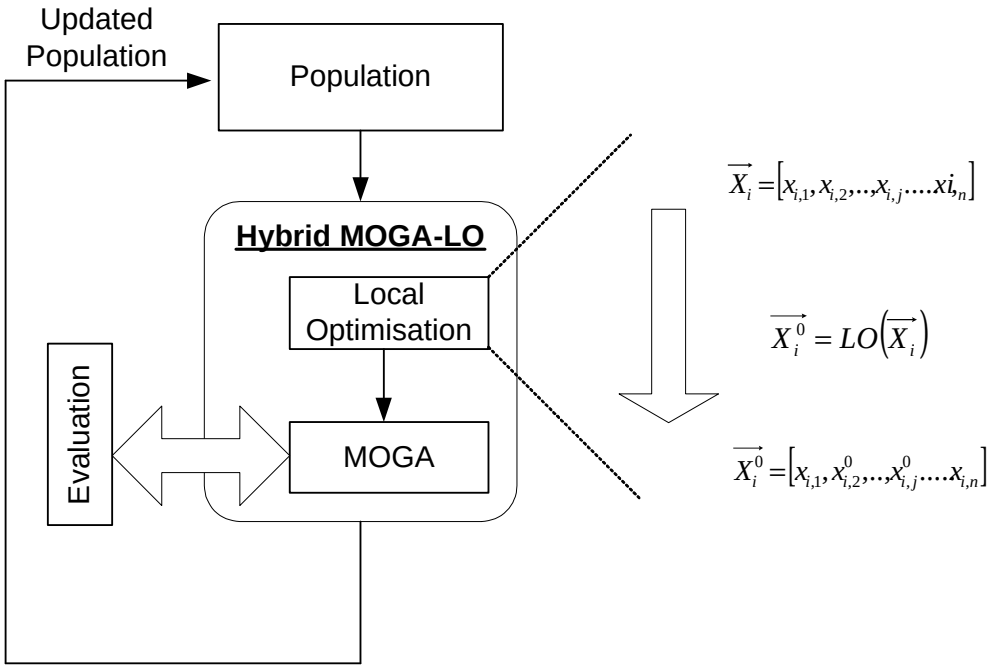
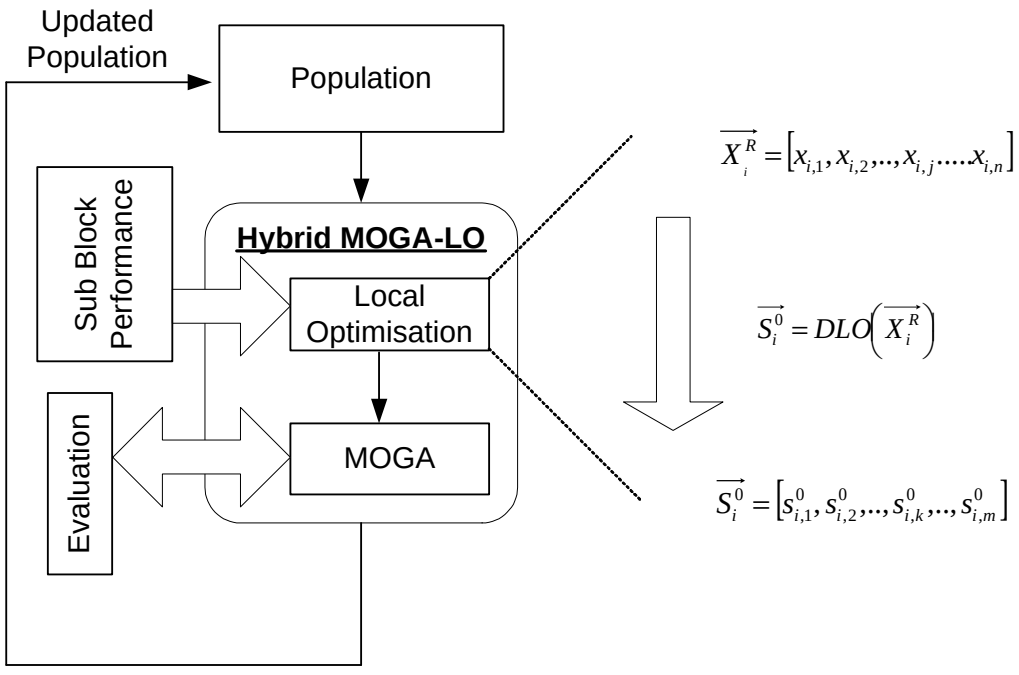


Figure 1 *(a) Top Down Constrained Design Methodology, (b) Multi-Objective Bottom-Up Methodology.*



(a)



(b)

Figure 2 *(a) Hybrid LO-MOGA, (b) Hybrid DLO-MOGA.*

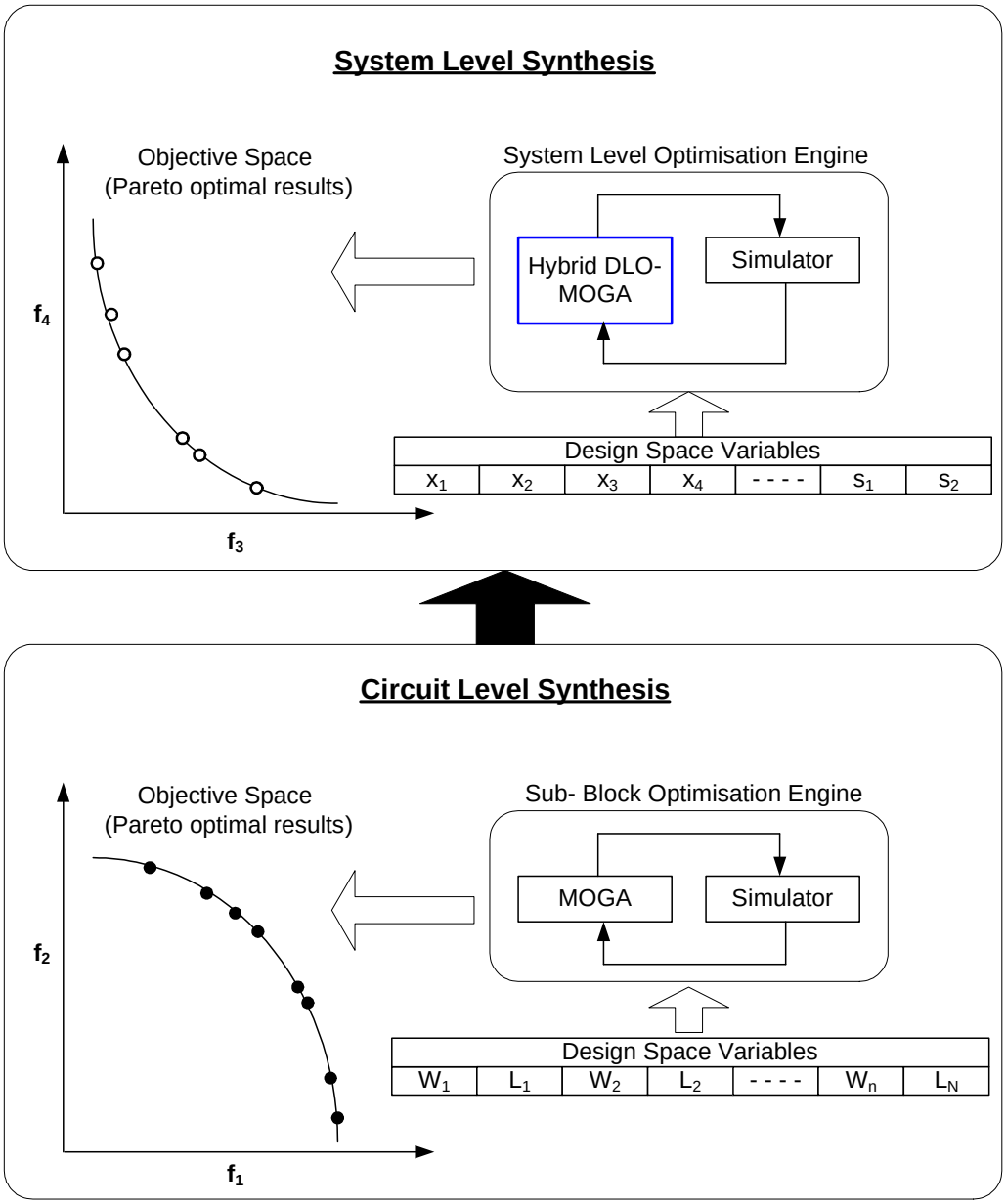


Figure 3 *Hierarchical Synthesis System.*

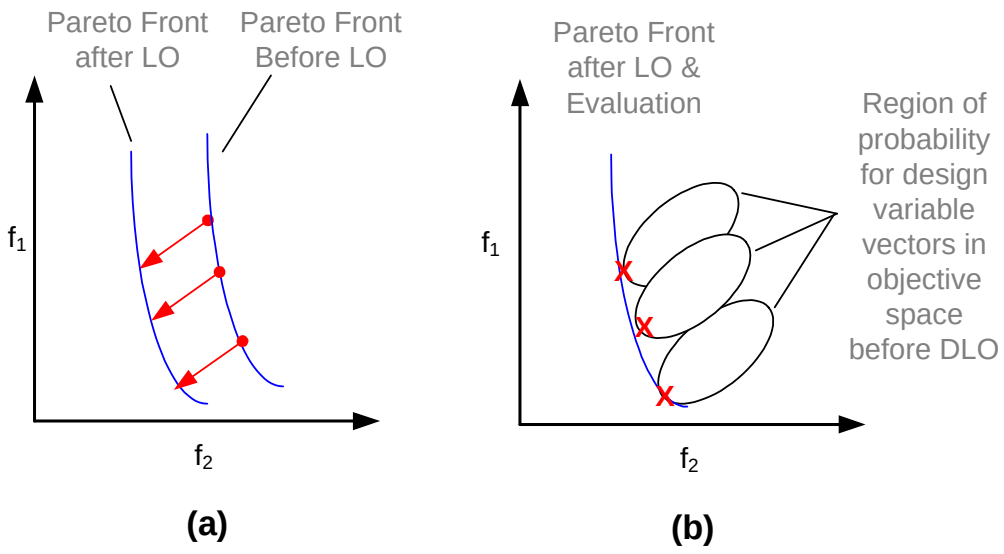


Figure 4 *Graphical Representation of Optimization: (a) LO-MOGA, (b) DLO-MOGA*

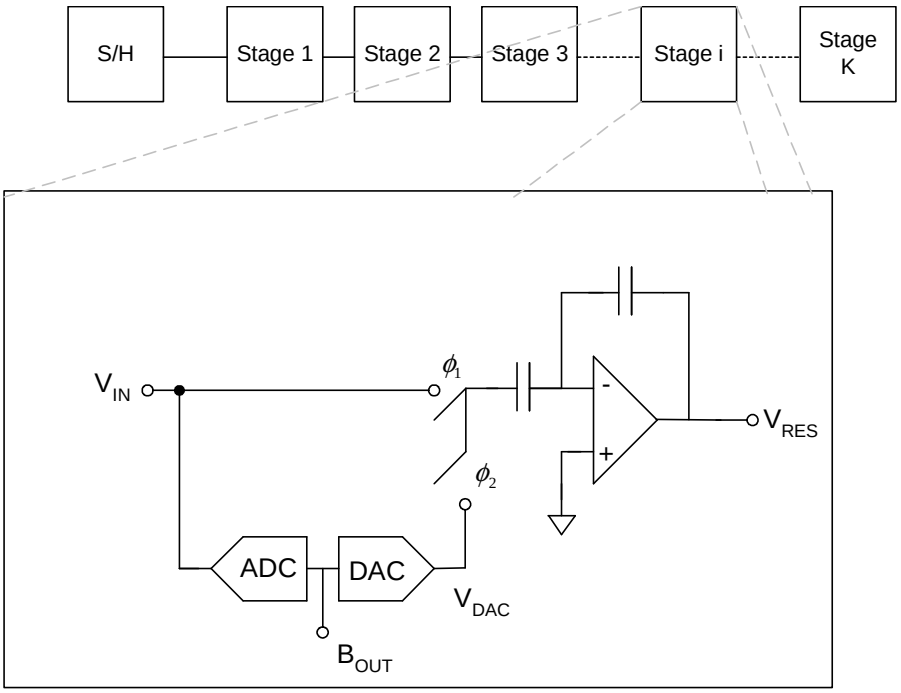


Figure 5 *Pipeline ADC.*

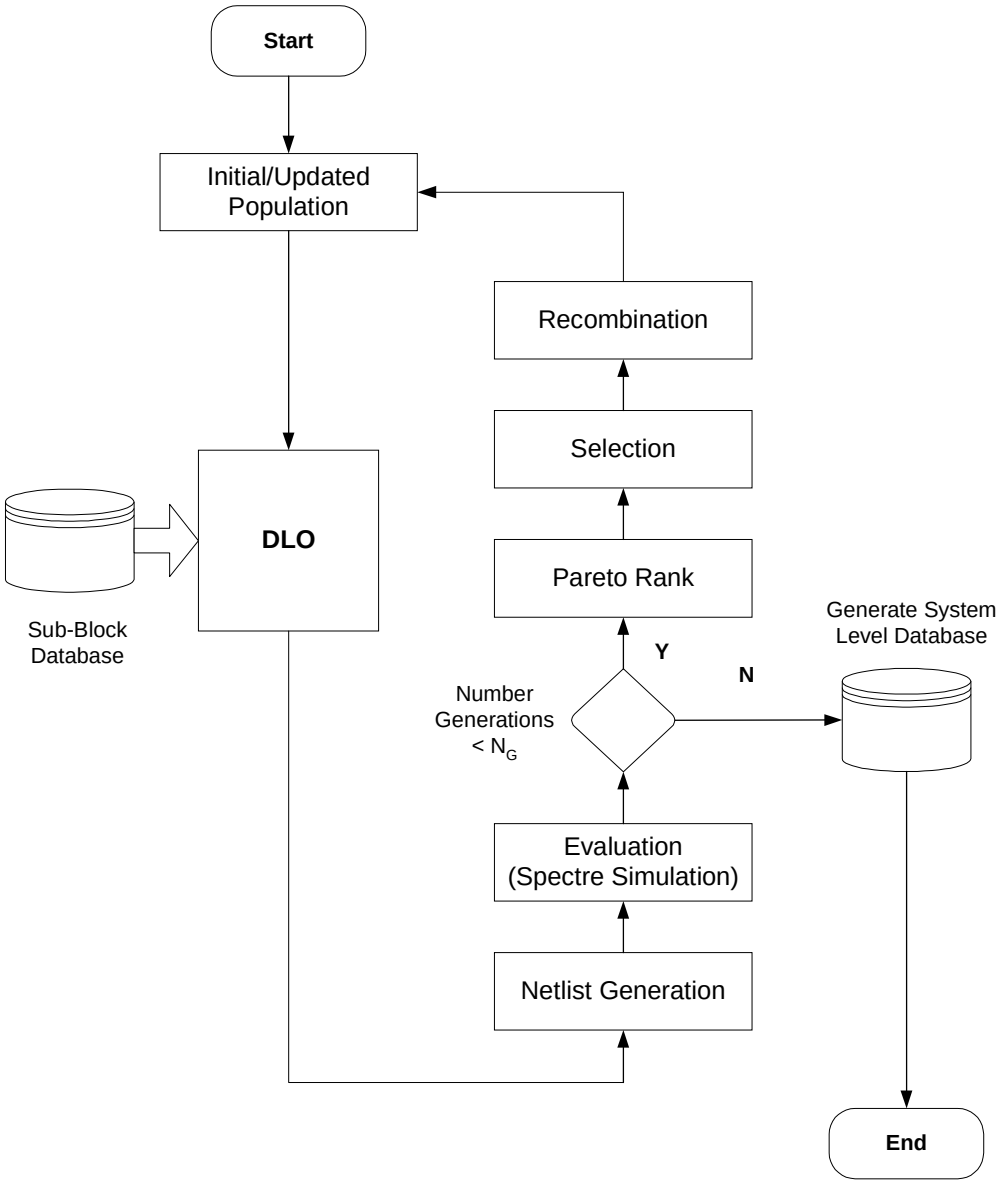


Figure 6 *Hybrid DLO-MOGA System Level Synthesis.*

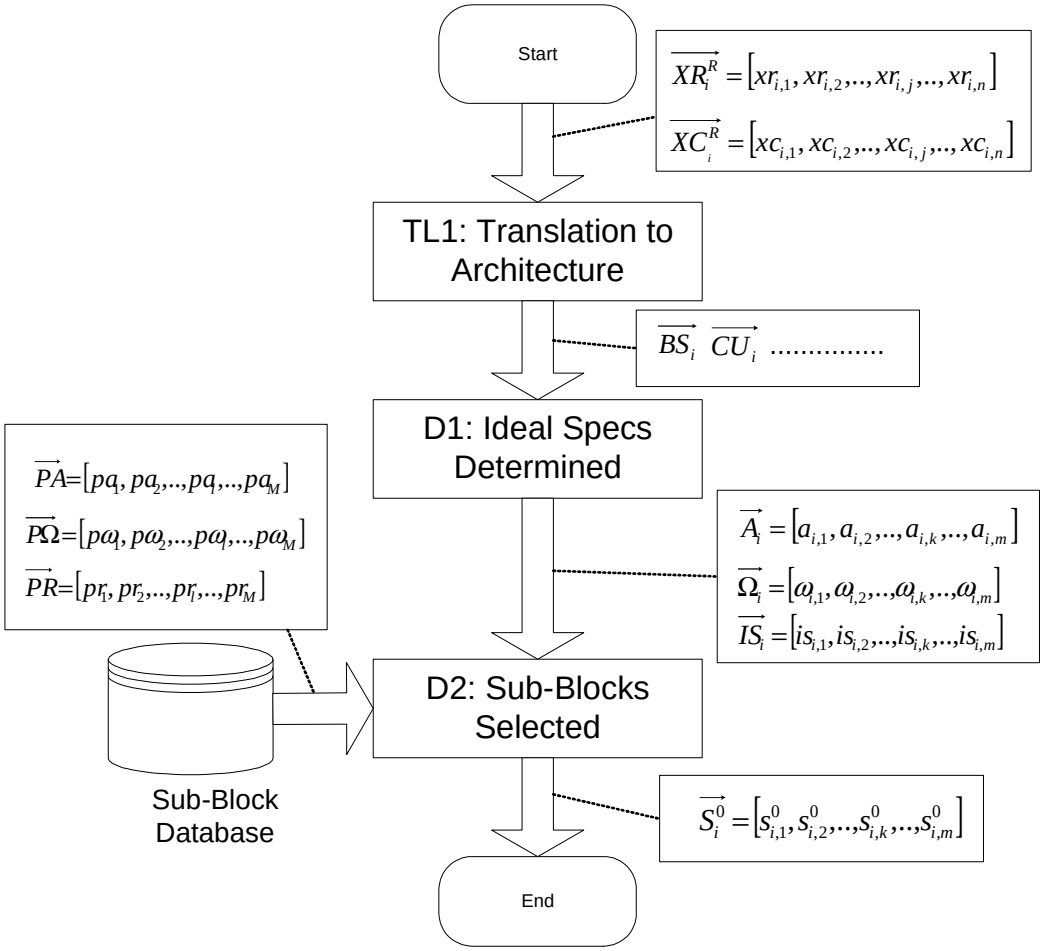


Figure 7 *Deterministic Local Optimization (DLO).*

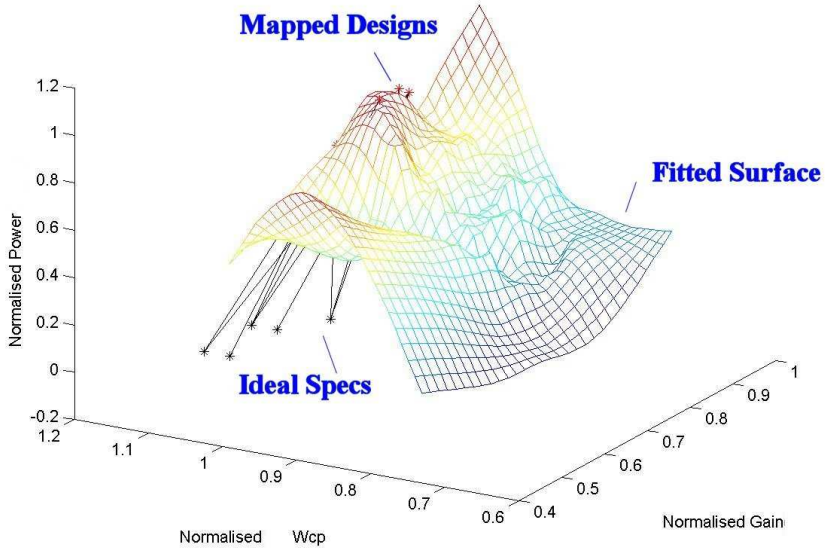


Figure 8 *Amplifier Selection.*

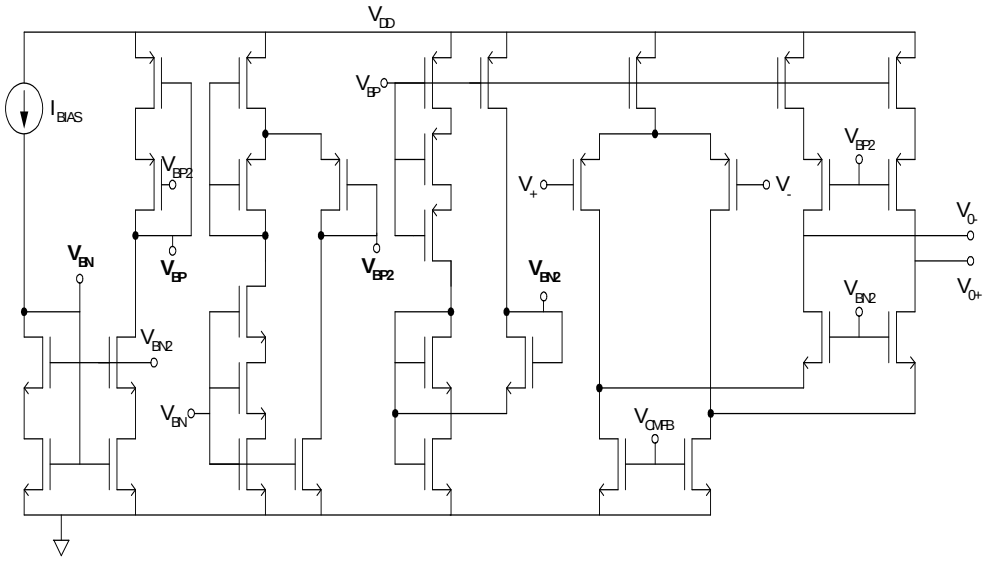
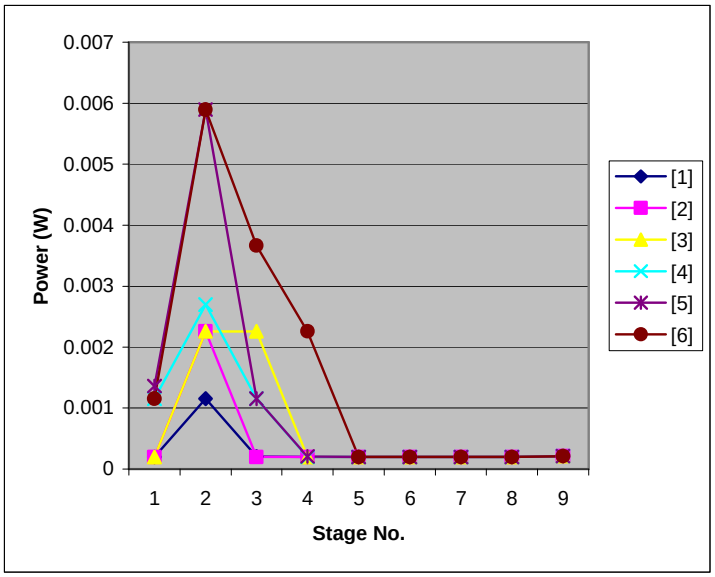
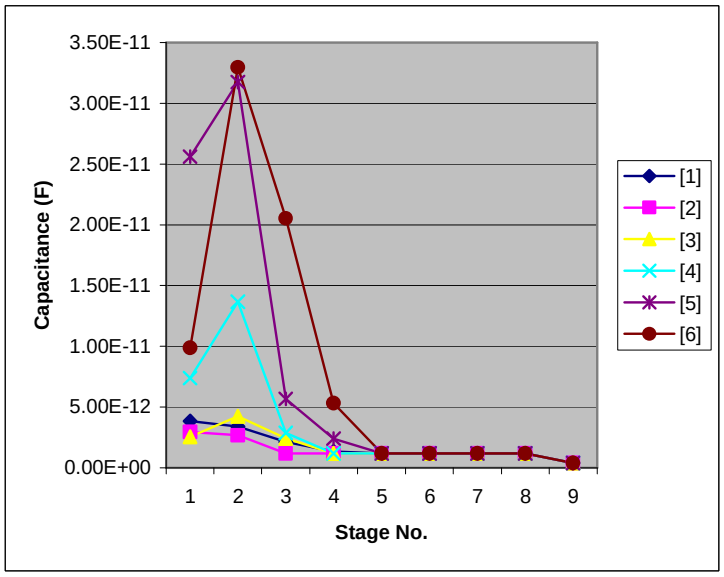


Figure 9 *Parameterizable Amplifier Cell.*

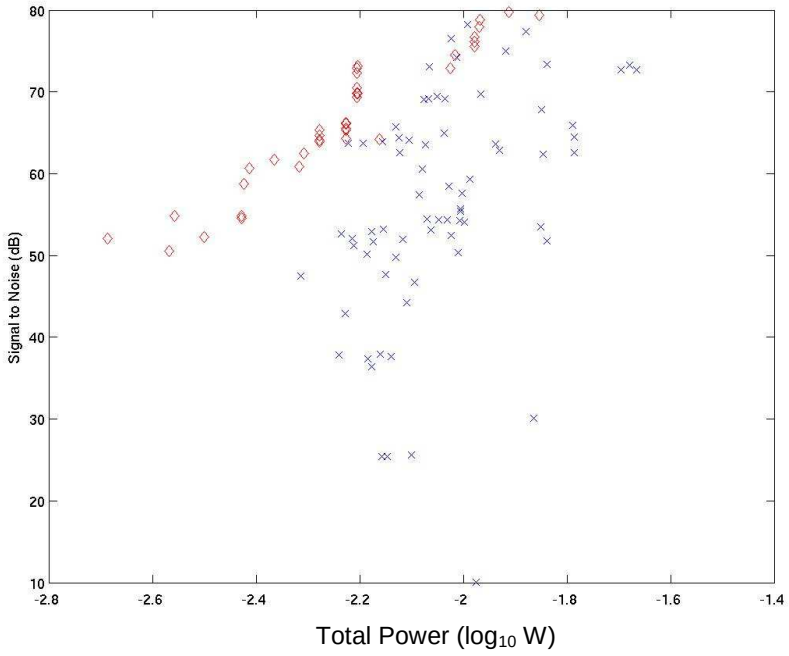


(a)



(b)

Figure 10 *(a) Power Scaling with Synthesized Pipelines, (b) Capacitor Scaling with Synthesized Pipelines.*



.

Figure 11 *Comparison of Synthesis Results: Hierarchical Synthesis with DLO-MOGA (◊), Hierarchical Synthesis with MOGA (x).*