2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)

Parasitic-Aware Automatic Analog CMOS Circuit Design Environment

Subhash J Patel
Sr. Engineer,
Kirtan Technologies,
Ahmedabad, India
Email: subhash.bvm@gmail.com

Rajesh A Thakker
Department of Electronics and Communication,
VGEC, Gujarat Technological University,
Ahmedabad, India
Email: rathakker2008@gmail.com

***Abstract*— In this work, the parasitic-aware design automation of the two-stage op-amp and bulk-driven OTA in $0.13\mu m$ are presented using two well-known swarm-optimization algorithms namely, ABC and PSO algorithms. To achieve the parasitic-aware design, we utilized configurable layouts. We also consider process and temperature variations in the automatic layout-level design of the op-amp. The average design time for layout-level design for op-amp using ABC algorithm is only 108 minutes while that for bulk-driven OTA is only 9 minutes. The obtained results reveal that the concept of parasitic-aware design using configurable layouts is an effective tool for designing high-performance analog CMOS circuits.**

***Index Terms*—Parasitic, Layout, CMOS, Analog Circuits, Op-Amp, OTA, Optimization**

I. Introduction

In recent years, with the increase of complexity, the design of the analog and mixed signal CMOS integrated circuits in an efficient manner is a critical task. In order to handle the complexity of integrated circuits, hierarchical blocks oriented design approach is used widely. The design analog CMOS circuit includes sizing of various MOSFETs to achieve target design specifications such as power consumption, gain, bandwidth, slew rate. Since the performance of such analog circuits is very sensitive to design parameters, their design requires lots of expertise. Further, improvement in one performance parameter may adversely affect the other performance parameters. Thus, the analog circuit design process is a knowledge intensive trade-off approach [1].

The traditional analog circuit design is carried out in three steps: (1) Topology selection, (2) Component sizing i.e. schematic-level design, and (3) Layout-design and verification i.e. layout-level design [2]. During the schematic-level design, generally, analytical calculations are followed by the circuit simulations. Finally, based on the schematic-level design, the layout of the circuit is prepared. The parasitics are extracted and the circuit is simulated to verify its performance. If desired performance is not achieved then the modifications in schematic and layout are carried out. During the schematic level design, to determine the length and width of the various MOSFETs, the behavior of the circuit is described by the set of mathematical equations. Based on this mathematical model, analytical calculations are carried out. This requires an accurate mathematical model of the MOSFET. Generally, BSIM spice models are widely used for describing the behavior of the MOSFET. The level-2 spice model of MOSFET has 13 parameters, and it is suitable to represent large-channel MOSFET (i.e. $L \geq 2\mu m$). The level-28 model has 63 parameters. Similarly, the level-54 BSIM model, which is suitable for describing the MOSFET in $0.13\,\mu m$ technology, has more than 150 parameters. This way, the BSIM models have become more complex to accommodate various short channel and other nonlinear effects of the devices. It is very difficult to consider such complex models in the analytical calculations of the circuit design process. This leads to the considerable difference in the result of analytical calculations and simulation. Thus, the dependence on the circuit simulation is increased and the design of analog circuit requires expertise. In order to overcome the difficulties associated with the analog circuit design, the research community is utilizing various optimization techniques based on classical and heuristic optimization algorithms. Due to the simplicity of the use, the evolutionary algorithms based approach is used widely at the schematic level design of the analog circuits [3]. Many schematic-level CMOS circuit design examples utilizing evolutionary optimization algorithms can be found in the literature [4]–[9]. Based on the optimized schematic, the layout of the circuit is prepared. For the analog circuit design, unlike the digital circuits, the layout of the circuit is prepared manually. Further, since the layout parasitics are not possible to consider during schematic-level design, the post-layout performance of the circuit differs from the schematic-level performance, especially, for the frequency sensitive specifications.

In this work, we propose a novel concept of the parasitic-aware automatic circuit design of analog CMOS circuits that extends the design automation from the schematic-level to layout-level. This is achieved using the configurable layouts. We have utilized, widely used, ABC (Artificial Bee Colony) [10] and PSO (Particle Swarm Optimization) [11] algorithms based optimization techniques to design Two-stage operational amplifier and Bulk-driven operational transconductance amplifier (OTA) at layout level in $0.13\mu m$ CMOS technology.

We use Magic VLSI Tool (open-source tool for layout design) as circuit design framework and ng-spice simulator. The designs of the op-amp and bulk-driven OTA are carried out on the computer having following major specifications: OS

2380-6923/19/$31.00 ©2019 IEEE
DOI 10.1109/VLSID.2019.00061

- Ubuntu 12.06 (64-bits), Processor - AMD 8350 with 4GHz clock, RAM - 6GB.

The rest of the paper is organized in the following manner. In section 2, ABC and PSO algorithms are discussed briefly. The concept of the automatic circuit design and configurable layouts are discussed in section 3. The parasitic-aware design of the Two-stage opamp and Bulk-driven OTA are discussed in sections 4 and 5, respectively. Section 6 highlights the conclusions.

II. Optimization Algorithms

A. Particle Swarm Optimization (PSO) Algorithm

The particle swarm optimization algorithm was suggested by Kennedy and Eberhart in 1995 [11]. It is based on the swarm optimization technique. It simulates the social behavior of birds flocking and fish schooling. The PSO algorithm models solution candidates by particles of a swarm. Each particle of the swarm is moving in the search space with a certain velocity. Each particle has information about its current position, its current velocity, best position ever visited by itself and, best position visited among all the particles. Based on these informations the position and velocity of each particle are calculated.

Let's assume that, the swarm of N particles is considered to solve the optimization problem having D dimensions. The position of each particle represents a solution candidate. The vector of D dimensions can be used to model the position of the particle mathematically. The current position of the i^{th} particle can be represented by vector $x_i(t) = \{x_{i1}, x_{i2}, \cdots, x_{iD}\}$. Similarly, the current velocity associated with the i^{th} particle can be given by vector $v_i(t) = \{v_{i1}, v_{i2}, \cdots, v_{iD}\}$. The new velocity $v_i(t+1)$ and position $x_i(t+1)$ of the i^{th} particle can be calculated as follows,

$$\begin{aligned} v_i(t+1) = w \cdot v_i(t) + C_1 \cdot R_1 \left(P_i(t) - x_i(t)\right) \\ + C_2 \cdot R_2 \left(P_g(t) - x_i(t)\right) \end{aligned} \tag{1}$$

$$x_i(t+1) = x_i(t) + v_i(t+1)\Delta t \tag{2}$$

where, w is an inertia weight factor, R_1 and R_2 are random numbers between 0 and 1, C_1 and C_2 are acceleration constants. C_1 is known as the cognitive coefficient and C_2 is called the social coefficient. $P_i(t)$ represents the best position ever visited by the i^{th} particle. $P_g(t)$ indicates the best position ever visited among all the particles. The value of Δt is 1.

The inertia constant w controls the influence of the previous velocity on the new velocity. The value of w can be kept constant or can be varied with the algorithmic iterations. The value of w can be varied linearly with algorithmic iterations as follows,

$$w = w_{up} - (w_{up} - w_{low}) \cdot \frac{i_c}{i_{max}} \tag{3}$$

where, w_{up} and w_{low} are upper and lower limits for the w, respectively, with $w_{up} > w_{low}$; i_c is current algorithm iteration; i_{max} represents maximum allowed algorithm iterations. Further, along with bounded search-space, the concept of the velocity clamping is used widely in PSO algorithm. The velocity clamping limits velocity of the particle to a limited range of $[-v_{max}, v_{min}]$. This helps preventing swarm explosion and controlling global exploration [12]. In this work, we use $v_{max} = (x_{max} - x_{min})/2$, where x_{max} and x_{min} represent the boundaries of the search space.

The important controlling parameters of the PSO algorithm are w, C_1 and C_2. The values of these parameters can be tuned according to the nature of the problem. The PSO algorithm starts with the particle initialization process. When the nature of the problem is unknown, random initialization can be used. Similarly, the initial velocity of the particle is also selected randomly. In each algorithmic iteration, the new position of each particle is calculated and each particle is evaluated for its fitness.

B. Artificial Bee Colony (ABC) Algorithm

The artificial bee colony algorithm simulates the social behavior of bees searching food to find the solution of the optimization problem. The algorithm is suggested by D Karaboga in 2005. This algorithm is based on the swarm optimization technique. The details about the algorithm and, comparison with other algorithms can be found in [13], [14].

In the ABC algorithm, the swarm consists of three types of bees; (1) employee bees, (2) onlooker bees and, (3) scout bee. The employee bees and onlooker bees are same in numbers. The employee bee tries to find a new solution by carrying out a local search. The onlooker bees select the solutions having higher chances of improvement and only try to improve these selected solution candidates.

Let's consider an optimization problem with D dimensions. We assume that there are N number of employee bees and the same number of the onlooker bees in the swarm. In the ABC algorithm, the solution candidate is represented by food-source. Let's assume that there are N number of food-souces. The food-source can be modeled mathematically modeled using a vector of D dimension i.e. i^{th} food-source can be represented by, $x_i = \{x_{i1}, x_{i1}, \ldots, x_{iD}\}$.

The ABC algorithm has four major phases; (1) initialization (2) employee bee phase (3) onlooker bee phase and, (4) scout bee phase. In the initialization phase, random initialization of the food-sources is carried out. In the employee bee phase, each employee bee tries to find a better food-source around existing one. To do this, each employee bee shares information about the food-source to neighbor bee. For example, a new food-source v_i around the i^{th} food-source x_i can be found by getting influence about k^{th} food-source in the following manner,

$$v_{ij} = \begin{cases} x_{ij}, & \text{if } j \neq p \\ x_{ij} + \phi \cdot (x_{ij} - x_{kj}) & \text{if } j = p \end{cases} \tag{4}$$

where, $i = 1, 2, \ldots, N$; $j = 1, 2, \ldots, D$; ϕ is an random number in range $[-1, 1]$; k represents the food-source of neighbor bee, it is randomly selected integer in range $[1, N]$, with $i \neq k$; p is randomly selected number from range $[1, D]$.

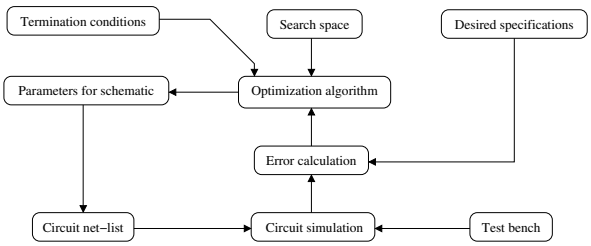

Fig. 1: Conceptual block diagram of optimizer for automatic CMOS circuit design at schematic-level.

The fitness fit_{v_i} of newly found food-source V_i is calculated as follows,

$$fit_{x_i} = \frac{1}{1 + f(x_i)} \tag{5}$$

where, $f(x_i)$ represents the cost function representing optimization problem. This is followed by the greedy selection process between v_i and x_i i.e. if fit_{v_i} is greater than fit_{x_i}, new food-source v_i is accepted otherwise, it is rejected and x_i is retained.

In the onlooker bee phase, selected food-sources having the better probability of improvement are selected. The probability of improvement P_i associated with food-source x_i can be calculated in the following manner,

$$P_i = \frac{fitx_i}{\sum_{i=1}^{i=N} fitx_i} \tag{6}$$

The selected food-sources are updated according to Eq. 4 similarly like in employee bee phase. In scout bee phase, the food-sources which are not improved after certain pre-determined trials (T_{limit}) are discarded and for each discarded food-source, new food-source is randomly initialized from the search-space. Thus, in the ABC algorithm, the employee bees try to improve each food-source equally, the onlooker bees only focus on the food-sources where the probabilities of improvements are higher.

III. Automatic Circuit Design and configurable Layouts

In Fig. 1, the conceptual block diagram of the optimizer for the automatic analog circuit design at schematic-level is illustrated. The search-space generally contains the information about the upper and lower bounds for the various design parameters. The proper search-space helps algorithm to avoid generating non-practical solutions. From the search space, optimization algorithm generates set of solutions containing design parameters for the circuit. From the search space, optimization algorithm generates set of solutions containing design parameters for the circuit. For the each provided solution, the circuit is simulated against pre-determined test-bench. The error is calculated from the obtained simulation results and the calculated error is used by the algorithm to generate a new set of solutions. In order to calculate the error,

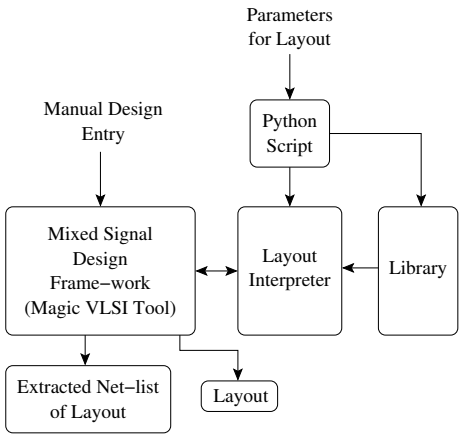

Fig. 2: User interface for configurable layout

the formula of the root-mean-square (RMS) is utilized in this work. Let's assume that there are N specifications considered for the circuit design. Then, the RMS error is calculated as follows,

$$f_e\left(RMS\ Error(\%)\right) = \sqrt{\frac{1}{N}\sum_{i=1}^{i=N} E_i} \times 100 \tag{7}$$

$$E_i = \begin{cases} 0, & \text{if } Spect_i\ is\ satisfied \\ \left(\dfrac{Spect_i^{sim} - Spect_i}{Spect_i}\right)^2, & \text{Otherwise} \end{cases}$$

where, $i = 1, 2, \ldots N$; $Spect_i$ is i^{th} desired specification; $Spect_i^{sim}$ is i^{th} specification obtained after circuit simulation. The RMS error gives equal importance to each specification.

The automatic circuit design using evolutionary algorithm is an iterative process. The algorithm continues to generate new solutions until the termination criteria are satisfied or all the specifications are obtained. The termination criteria are generally described in term of the design time or allowed number of circuit evaluations.

In order to consider the layout parasitic in the design, it is necessary to utilize extracted net-list containing all parasitics. However, in the analog circuit design, the layouts are prepared manually. Due to this, such automation in design is difficult. In order to overcome this difficulty, the concept of the configurable layout is proposed. The configurable layouts provide flexibility through which it is possible to change the dimensions of the dimensions of various layout components and distance between them by changing the parameters of the layout. Further, the configurable layout can be ported to the other technologies very easily. Further, the design environment for the configurable layout is added to the design framework as an add-on and does not block any feature of the framework. The design environment for the configurable layout using MAGIC VLSI Tool [15] is illustrated in Fig. 2. Unlike a traditional design approach for preparing a layout, the

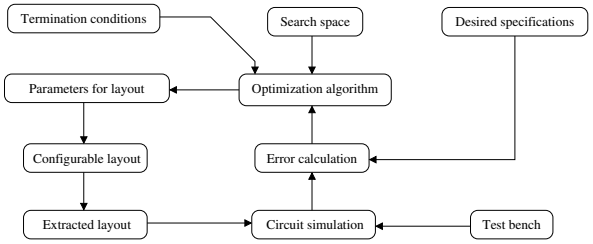

Fig. 3: Conceptual block diagram of optimizer for the parasitic-aware automatic CMOS circuit design.

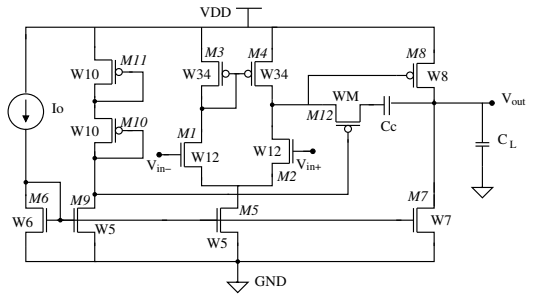

Fig. 4: Circuit of two-stage operational amplifier.

configurable layouts are described using the script containing specially designed macros. This script file is interpreted and converted into the design framework specific macros. Finally, design framework is triggered to generate the layout and netlist with all kind of the parasitics can be extracted for the simulation purpose. With the help of the configurable layouts, the layout can be modified instantly by passing appropriate values of the parameters. This provides the flexibility of the schematic to the layouts and makes them suitable for algorithm based design automation. In Fig. 3, the conceptual block diagram describing the parasitic-aware automatic circuit design is illustrated. In this approach, extracted net-list containing all kind of parasitic is utilized in optimization process instead of the simple schematic. This makes the design process parasitic-aware and robust.

IV. Two-stage Operational Amplifier

The two-stage operational amplifier (op-amp) is a very versatile circuit. It is widely used in the analog CMOS integrated circuits. It is a building block of many important circuits such as filter, mixer, modulator, ADC, DAC, amplifier, analog front-end etc. The circuit of the designed op-amp is shown in Fig. 4 [16]. n this work, the parasitic-aware automatic circuit design of the two-stage operational amplifier is carried out in $0.13\ \mu m$ and $0.35\ \mu m$ CMOS technologies. The circuit of op-amp is shown in Fig. 4. The op-amp is designed for the following specifications: Gain $\geq 80dB$, Unity gain bandwidth (UGB) $\geq 100MHz$, Phase margin $\geq 60^{\circ}$, Power consumption $\leq 30\mu W$, Rise and fall slew rate $\geq 35V/\mu S$, Power supply rejection ratio (PSRR) $\geq 75dB$, Common mode rejection ratio (CMRR) $\geq 80dB$.

TABLE I: Two stage op-amp (parasitic-aware design): Average of simulation results.

	ABC Algorithm			PSO Algorithm
Specifications	Schematic level optimization	Post-layout Simulation	Parasitic aware design	Parasitic aware design
Gain (dB)	81.6	81.6	81.7	81.5
PM ($^{\circ}$)	66.5	52.7	62.7	62.6
UGB (MHz)	169.3	147.7	136.8	110.0
PSRR (dB)	80.3	80.3	76.0	78.0
CMRR (dB)	92.6	92.6	81.8	83.4
PC (μW)	28.2	28.2	28.6	29.0
RSR ($V/\mu S$)	53.9	54.6	37.8	47.2
FSR ($V/\mu S$)	35.5	31.3	37.0	37.0
RMS Error (%)	0	6.46	0	0.23

The various design parameters for the op-amp are width and length of various transistors and value of the Miller capacitor. The value of the supply voltage VDD is fixed to 1.2V and value of a bias current I_0 is set to $3\mu A$. In the layout, the value of the capacitor depends upon the area of the capacitor. The width of the Miller capacitor in the layout of the op-amp is decided automatically in the configurable layout, while its height ($CapH$) is a design parameter. Further, the length of all transistors is kept equal. The maximum circuit evaluations during the automatic design process are set to 5000.

In the first experiment, the op-amp is designed in $0.13\ \mu m$ CMOS technology using the ABC algorithm at schematic-level where the layout parasitics are not possible to consider. We carried out ten independent design trials. The average of the obtained results is shown in Table I. The obtained results reveal that the average RMS error is 0% i.e. the ABC algorithm had designed op-amp at schematic-level all ten times satisfying all the specifications. However, when the layouts are prepared from these optimized schematics and post-layout simulation is carried out for each design run, the average RMS error is jumped to 6.46%. The obtained simulation results show that the frequency sensitive specifications such as phase-margin, unity gain bandwidth, and slew rate are affected worst. The deterioration in the performance at layout-level is due to the added parasitic-components at layout-level, which is not considered at schematic-level optimization. In the second experiment, the op-amp is designed at layout-level using the concept of the parasitic-aware automatic circuit design with the help of the ABC and PSO algorithms ten times. The average of the obtained result is illustrated in Table I. The obtained results reveals the effectiveness of the parasitic-aware design concept. The ABC algorithm has designed op-amp at layout level with no error while PSO algorithm has designed op-amp with average error of 0.23%. The average design times are 107.8 and 79.4 minutes for ABC and PSO algorithms, respectively. Further, it is clear that the ABC algorithm outperforms the

TABLE II: Two stage op-amp design (parasitic-aware) using ABC algorithm considering process and temperature variations

Process	Temperature ($^\circ C$)	Gain (dB)	UGB (MHz)	PM ($^\circ$)	PSRR (dB)	CMRR (dB)	PC (μW)	RSR (V/μS)	FSR (V/μS)
TT	0	81.8	137.8	70.3	80.2	82.4	29.6	40.1	36.4
	25	81.2	130.4	67.9	79.2	81.9	29.6	40.1	36.1
	70	80.2	116.8	64.8	77.5	81.2	29.6	40.0	35.8
FF	0	81.6	135.3	72.4	80.5	83.9	29.8	40.4	36.6
	25	81.1	129.2	70.5	79.5	83.6	29.8	40.3	36.4
	70	80.2	117.5	68.0	77.9	83.2	29.8	40.2	36.1
FS	0	81.8	142.6	66.9	78.8	82.1	29.6	40.3	36.6
	25	81.2	132.3	63.9	77.6	81.5	29.6	40.2	36.4
	70	80.2	117.1	60.3	75.5	80.6	29.6	40.1	36.1
SF	0	81.9	128.9	73.4	81.2	83.3	29.5	40.0	35.8
	25	81.3	123.4	71.6	80.3	82.8	29.5	39.9	35.5
	70	80.4	112.6	69.3	78.8	82.3	29.5	39.9	35.3
SS	0	82.1	137.5	68.1	79.9	82.1	29.4	39.9	36.0
	25	81.5	127.7	65.3	78.8	81.5	29.4	39.9	35.8
	70	80.5	113.2	61.9	77.0	80.6	29.4	39.8	35.5

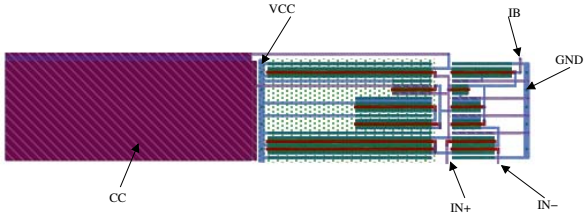

Fig. 5: Two-stage op-amp in $0.13\,\mu m$ technology: Layout of best design obtained from ABC algorithm.

PSO algorithm.

In the third experiment, we also consider the process and temperature variations in the parasitic-aware design of the op-amp. This time, we utilize only ABC algorithm. The obtained specifications at five process corners (TT, FF, FS, SF, SS) and three different temperatures (0°, 25°, 70°) are shown in Table II. The obtained layout level simulation results reveal that the op-amp is satisfying specifications at all process corners. Fig. 5 the layout of the best obtained parasitic-aware design using ABC is illustrated.

V. Bulk-driven Operational Transconductance Amplifier

The operational transconductance amplifier (OTA) is widely used analog circuit. It is a fundamental block of circuits such as the active resistor, active inductor, voltage controlled oscillator, ADC, DAC, gm-c filter, etc. In recent years, the demand for the low voltage circuits has been increased. However, due to the limitations imposed by the threshold voltage of the MOSFET, standard gate-driven circuits are not suitable for low-voltage applications. The bulk-driven technique is promising alternative to overcome limitations imposed by gate-driven technique for designing low voltage circuits [17]. In the bulk-driven technique, the input signal is applied to a substrate terminal of the MOSFET instead of the gate terminal. Due to this, the supply voltage is not limited by the threshold voltage.

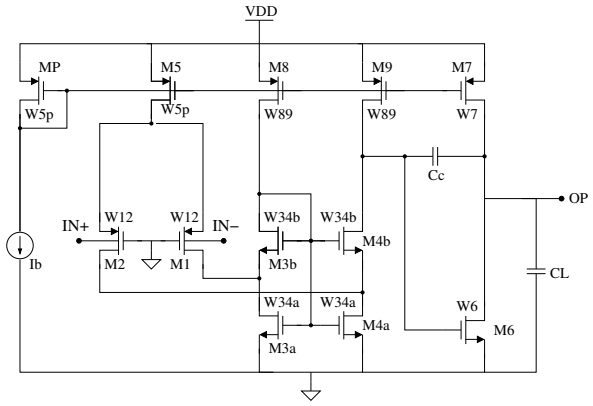

Fig. 6: Circuit of bulk-driven OTA.

Further, the use of the bulk-driven technique does not require any change in the MOSFET structure. This allows the use of the bulk-driven method in standard CMOS technology without any modifications [18]. However, the transconductance offered by the bulk-driven device is significantly low compared to the gate-driven device. This results in the lower gain of the amplifier.

In this work, the parasitic-aware design of bulk-driven OTA in $0.13\mu m$ technology is presented using ABC and PSO algorithms. In Fig. 6, the circuit of the designed OTA is illustrated. The supply voltage is set to $0.25V$. The value of the current source I_b is set to $15nA$. The bulk-driven OTA is designed to satisfy the following specifications: Gain $\geq 40dB$, Bandwidth (BW) $\geq$ $1KHz$, Phase margin $\geq$ 60°, Power consumption $\leq 20nW$, Rise and fall slew rate $\geq 0.3V/mS$.

Each algorithm has designed bulk-driven OTA at layout-level 10 times and for the comparison average of the obtained results is considered. The design parameters are width and length of the various transistor and value of the Miller capacitor. The capacitor value is decided by its area. The width of

TABLE III: Bulk-driven OTA (parasitic-aware design): Average of simulation results.

Specification	ABC	PSO
Gain (dB)	41.5	41.1
PM ($^\circ$)	63.2	63.0
BW (KHz)	2.2	2.3
PC (μW)	19.8	19.5
RSR (V/mS)	0.69	0.72
FSR (V/mS)	0.99	1.01
RMS Error (%)	0.0	0.0
Time (Minutes)	8.4	5.7

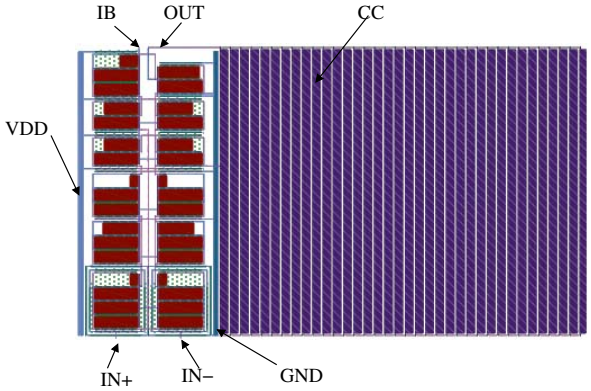

Fig. 7: Bulk-driven OTA (parasitic-aware design): Layout of best design.

the capacitor is calculated automatically in configurable layout and its height is a design parameter.

The obtained results reveal that the parasitic-aware layout-level design of the bulk-driven OTA takes the average time of 8.4 minutes for ABC algorithm and 5.7 minutes using PSO algorithms. Both algorithms have designed bulk driven OTA successfully in a time-efficient manner. The layout of OTA from the best design is illustrated in Fig.

VI. Conclusion

We have proposed the concept of parasitic-aware automatic circuit design for analog and mixed signal CMOS circuits. The concept is implemented using MAGIC VLSI tool by developing configurable layouts and interfacing optimization algorithm with it. The proposed concept enables the possibility of considering layout parasitic from the beginning of the design process. The parasitic-aware designs of two-stage op-amp and bulk-driven OTA are carried out in $0.13\mu m$ CMOS technology. The obtained results show the effectiveness of the proposed approach and reveal that the parasitic-aware optimization approach is very effective for designing high-performance circuit. Further, configurable layouts provide flexibility of the porting the existing solutions in other CMOS technologies with fewer efforts compared to traditional design approach.

References

[1] P. Mandal and V. Visvanathan, "Cmos op-amp sizing using a geometric programming formulation," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 20, no. 1, pp. 22–38, 2001.

[2] P. K. Rout, "Fast and robust design of cmos vco for optimal performance," Ph.D. dissertation, National Institute of Technology, Rourkela, India, National Institute of Technology, Rourkela, India, Aug. 2015.

[3] S. Patel and R. A. Thakker, "Automatic circuit design and optimization using modified pso algorithm," *Journal of Engineering Science and Technology Review*, vol. 9, no. 1, pp. 89–94, 2016.

[4] M. Barros, J. Guilherme, and N. Horta, "Analog circuits optimization based on evolutionary computation techniques," *INTEGRATION, the VLSI journal*, vol. 43, no. 1, pp. 136–155, 2010.

[5] S. L. Sabat, K. S. Kumar, and S. K. Udgata, "Differential evolution and swarm intelligence techniques for analog circuit synthesis," in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*. IEEE, 2009, pp. 469–474.

[6] R. A. Thakker, M. S. Baghini, and M. B. Patil, "Automatic design of low-power low-voltage analog circuits using particle swarm optimization with re-initialization," *Journal of Low Power Electronics*, vol. 5, no. 3, pp. 291–302, 2009.

[7] B. P. De, R. Kar, D. Mandal, and S. Ghoshal, "Optimal design of high speed symmetric switching cmos inverter using hybrid harmony search with differential evolution," *Soft Computing*, pp. 1–19, 2015.

[8] A. Sallem, B. Benhala, M. Kotti, M. Fakhfakh, A. Ahaitouf, and M. Loulou, "Application of swarm intelligence techniques to the design of analog circuits: evaluation and comparison," *Analog Integrated Circuits and Signal Processing*, vol. 75, no. 3, pp. 499–516, 2013.

[9] A. Somani, P. P. Chakrabarti, and A. Patra, "An evolutionary algorithm-based approach to automated design of analog and rf circuits using adaptive normalized cost functions," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 3, pp. 336–353, 2007.

[10] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," *Journal of global optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.

[12] F. Van den Bergh and A. P. Engelbrecht, "A study of particle swarm optimization particle trajectories," *Information sciences*, vol. 176, no. 8, pp. 937–971, 2006.

[13] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Applied Mathematics and Computation*, vol. 214, no. 1, pp. 108–132, 2009.

[14] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (abc) algorithm," *Applied soft computing*, vol. 8, no. 1, pp. 687–697, 2008.

[15] "Magic VLSI Layout Tool," http://opencircuitdesign.com/magic/, accessed: 2017-08-28.

[16] P. Allen and D. Holberg, *CMOS Analog Circuit Design*. OUP USA, 2012.

[17] Y. Haga, H. Zare-Hoseini, L. Berkovi, and I. Kale, "Design of a 0.8 volt fully differential cmos ota using the bulk-driven technique," in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*. IEEE, 2005, pp. 220–223.

[18] R. He and L. Zhang, "Evaluation of modern mosfet models for bulk-driven applications," in *Circuits and Systems, 2008. MWSCAS 2008. 51st Midwest Symposium on*. IEEE, 2008, pp. 105–108.