Parasitic-Aware Design and Optimization of A Fully Integrated CMOS Wideband Amplifier[1]

Jinho Park, Kiyong Choi, and David J. Allstot

Dept. of Electrical Engineering
University of Washington
Seattle, WA 98195-2500

Abstract - **A custom CAD synthesis tool based on particle swarm optimization, and results from the design of an RF CMOS distributed amplifier optimized to overcome non-idealities associated with parasitic-laden passives, are presented. The particle swarm synthesis approach is shown to be more than an order of magnitude faster than the simulated annealing design and optimization algorithm.**

I. INTRODUCTION

Wideband amplifiers are used in many radio frequency (RF) and high-data rate communication systems including satellite transceivers, pulsed radar systems, optical receivers, etc. For such applications, a distributed amplifier topology is often used because it can overcome the classical amplifier gain-bandwidth tradeoff by combining the outputs from several active gain elements in an *additive* fashion [1].

The burgeoning demand for CMOS system-on-chip (SOC) solutions has spawned intense worldwide research on CMOS RF integrated circuits. Unfortunately, the mundane issue of parasitics associated with on-chip passive and active components has impeded development. In fact, without CAD optimization, many RF circuits are virtually impossible to design owing to the frequency-dependent non-linear parasitics.

To achieve optimum performance from CMOS RF and analog ICs, parasitic-aware design and optimization techniques have been developed based on the simulated annealing (SA) algorithm [2]-[4]. A key advantage of simulated annealing is that it avoids being trapping in local minima of the design space with high probability; however, it is also slow due to its use of the *Metropolis* algorithm [5]-[6]. Hence, fast and accurate synthesis of CMOS RF circuits demands a more efficient optimization methodology.

In this paper, we propose a new parasitic-aware synthesis technique based on particle swarm optimization (PSO). The PSO algorithm is fundamentally different from the SA approach, and is also considerably faster since it works concurrently with a *population* of possible solutions rather than on a single one [7]. In this paper, we describe the first application of the PSO-based CAD tool in the design and optimization of a CMOS RF distributed amplifier.

II. SYNTHESIS OF RF INTEGRATED CIRCUITS

A. Topology of the Distributed Amplifier Design Example

The four-stage distributed amplifier shown in Fig. 1 employs two artificial L-C delay lines. One—called the *gate line*—applies delayed versions of the input RF signal sequentially to the four gate terminals. The other—*the drain line*—adds the drain signal currents constructively in the load resistor. Cascode transistors (not shown) are used to reduce the Miller effect by imposing a low impedance at the drains of the amplifying devices. Cascoding also reduces capacitive coupling between the artificial transmission lines and increases gain flatness, reverse isolation, stability, and input and output impedance matching accuracy. The high output impedance of the cascode configuration also reduces loss associated with amplifier loading on the drain line.

An L-C transmission line exhibits an intrinsic mismatch at each termination point due to image impedance variations with frequency. Hence, *m*-derived half sections are inserted to improve the impedance matches to the delay lines [8].

In order to increase gain flatness and decrease gain peaking near the cutoff frequency, a staggering technique is frequently used in distributed amplifiers [9]. Staggering basically means adopting slightly different cutoff frequencies for each delay line in order to increase the overall linearity of phase response. Specifically, designing the gate line to have a slightly higher cutoff improves gain flatness.

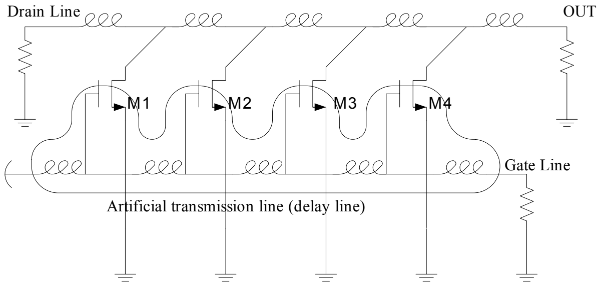

Fig. 1. Four-stage CMOS distributed amplifier with artificial L-C gate and drain delay lines.

[1] *Research supported by the National Science Foundation contracts CCR-0086032 and CCR-0120255, the Semiconductor Research Corporation contracts 2000-HJ-771 and 2001-HJ-925, and by grants from Texas Instruments, Inc., and National Semiconductor Corp.*

Assuming matched impedances at the input and output ports of the distributed amplifier, the cutoff frequency, characteristic impedance, and low frequency gain are:

$$f_c = \frac{1}{\pi\sqrt{LC}} \tag{1}$$

$$Z_0 = \sqrt{\frac{L}{C}} \tag{2}$$

$$S_{21} \cong \frac{n}{2} g_m Z_0 \tag{3}$$

where g_m is the small-signal transconductance of the identical active devices [1]. The *n* term in (3) indicates the unique property of gain addition in the distributed amplifier.

B. Design and Implementation

The technology used is a 0.25µm CMOS process. Key amplifier specifications are a constant gain greater than 19dB over a bandwidth greater than 12GHz with linear phase over the full bandwidth. The input and output ports are matched to 50Ω, and the distributed amplifier operates from a single 2.5V power supply.

The continued scaling of CMOS technology has led to the design of RF active devices along with analog and digital circuitry on the same chip. Ironically, the poor characteristics of the *passives* render CMOS RF circuit design a daunting challenge. Fig. 3 illustrates an intolerable degradation in distributed amplifier performance when the ideal inductors are replaced by their parasitic-laden counterparts: the bandwidth decreases to about 10GHz and the gain drops to 15dB. In this example, parasitics adversely affect the final design because they were not considered early in the design phase. Conversely, the parasitic-aware design methodology considers parasitics as an integral part of the synthesis process. Efficient CAD synthesis is the key enabling technology for the parasitic-aware methodology.

III. PARASITIC-AWARE CAD SYNTHESIS

A. Particle Swarm Optimization for RF Circuit Design

In essence, we seek to *re-tune* the RF circuit to regain phase synchronization and compensate for parasitic losses, but this is a complex task that involves optimizing continuous nonlinear multidimensional functions. In order to expedite the process, we now describe a new synthesis tool based on the particle swarm optimization algorithm. This evolutionary computational technique was originally inspired from observations of group social behavior; e.g., a flock of birds flying in search of food maintains an optimum distance between themselves [7]. In PSO, each particle represents a potential solution in the design space; it changes its position and velocity according to its own experience, and by communicating with others in the group to learn their experiences. The velocity and position for the i^{th} particle are updated according to:

$$\begin{aligned} v_i = w \cdot v_i + c_1 \cdot rand \cdot (p_{id} - x_i) \\ + c_2 \cdot Rand \cdot (p_g - x_i) \end{aligned} \tag{4}$$

$$x_i = x_i + v_i \tag{5}$$

where c_1 and c_2 are positive constants, *w* is the inertia weight, p_{id} and p_g are individual and group best positions, respectively, and *rand* and *Rand* are two random functions [10].

Particle swarm optimization avoids being trapped in local minima by adjusting a particle's position and velocity according to the best individual and group experience.

In order to optimize the RF CMOS distributed amplifier implemented with parasitic-laden spiral inductors, particles representing circuit variables are initially randomly placed over the design space, and *HSPICE* simulations are executed to evaluate a cost function for each particle. Then, individual and group best positions are used to adjust each particle position for the next iteration. For this synthesis example, an acceptable solution is one for which the cost function has a value of 100, or less.

Fig. 2 shows a flow chart for the parasitic-aware RF synthesis methodology using particle swarm optimization.

B. The Vmax Parameter Value

In particle swarm optimization, *Vmax*, the maximum velocity limit for the particles, is the only critical parameter that must be specified by the user [10]. Conceptually, *Vmax* is similar to the *temp* parameter used in simulated annealing. In our example, the search space is bounded within a normalized maximum distance of 1.0. Based on past experience, a setting of *Vmax* = 1 is good for a global search, while *Vmax* < 1 is preferred for a local search.

For comparison, 40 particle swarm optimization runs were performed on the CMOS distributed amplifier with *Vmax* = 1.0, 0.1, and 0.01, respectively. The results summarized in Table I show that *Vmax* = 0.1 provides the best performance, and *Vmax* = 1 the worst. The latter observation makes sense intuitively since the fast moving particles often reach the boundaries of the design space in a few iterations, bypassing the optimal points. At the other extreme, simulation results with *Vmax* = 0.01 show that the very slow moving particles struggle to find a minimum within the fixed total number of iterations. Note that with *Vmax* = 0.01 and 1.0, the particle swarm optimizer could not meet the cost function objectives of 97, and 97 and 98, respectively, within the limit of 15,000 iterations.

C. Comparison of Particle Swarm Optimization and Simulated Annealing

The simulated annealing technique is widely used for

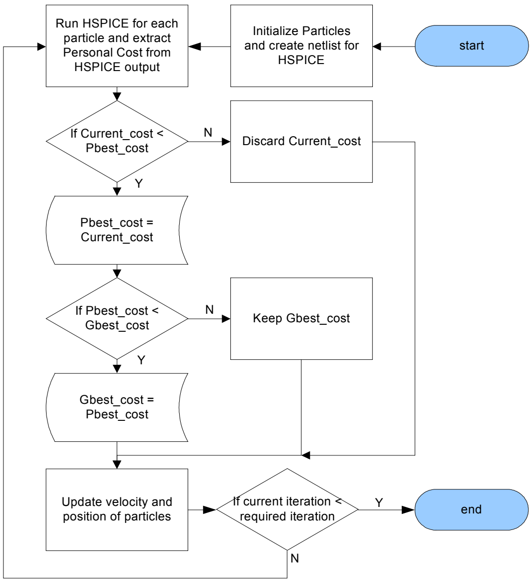
Run HSPICE for each particle and extract Personal Cost from HSPICE output
Initialize Particles and create netlist for HSPICE
start
If Current_cost < Pbest_cost
N
Discard Current_cost
Y
Pbest_cost = Current_cost
If Pbest_cost < Gbest_cost
N
Keep Gbest_cost
Y
Gbest_cost = Pbest_cost
Update velocity and position of particles
If current iteration < required iteration
Y
end
N

Fig. 2. Parasitic-aware particle swarm synthesis.

TABLE I
Number of Iterations v/s *Vmax* Value

	Vmax		
Cost	0.01	0.1	1.0
105	469.3	75.7	757.2
104	598.2	112.6	968.3
103	844.1	135.7	1023.2
102	1265	245.5	1061
101	1873	334.2	1594
100	2538	429.2	2068
99	3242	727.4	3472
98	4406	2226	X
97	X	4725	X
Min Cost	97.3	96.5	97.1

CMOS RF and analog circuit synthesis [2]-[4]. It avoids local minima by using a hill-climbing process; that is, it conditionally accepts poorer solutions in order to escape local minima while searching for the global minimum. Consequently, the hill-climbing process is inherently slow. In contrast, particle swarm optimization is a *population-based* algorithm; i.e., it uses multiple concurrent potential solutions as search agents. Thus, it achieves high efficiency in global search applications [7][10].

For purposes of comparison, all of the synthesis runs use the same CMOS RF distributed amplifier topology, cost function, and best parameter settings. The maximum number of iterations allowed is arbitrarily set to 15,000. If an optimizer cannot find an acceptable solution within 15,000 iterations, it is assumed that it failed to find the global optimum during that run. For each optimizer and cost function, 40 runs were performed and the total numbers of iterations needed to find the optimum were recorded.

Results comparing the average number of iterations to reach a specified value of cost function for both the particle swarm and simulated annealing optimizers are summarized in Table II. Clearly, the particle swarm approach reaches the optimum circuit design much more quickly and accurately than the simulated annealing algorithm. In fact, the computational advantage of particle swarm over simulated annealing exceeds an order of magnitude as the constraints become tighter and as the total iterations increase.

TABLE II
Particle Swarm v/s Simulated Annealing Interations

Cost	PSO	SA
105	75.7	168.2
104	112.6	259.1
103	135.7	401.9
102	245.5	821.4
101	334.2	3211
100	429.2	10418
99	727.4	14753
98	2226	X
97	4725	X
Min Cost	96.5	98.9

D. RF Circuit Optimization Results

Fig. 3 and 4 illustrate some key final results for the parasitic-aware synthesis and optimization of the RF CMOS distributed amplifier. Substantial performance improvements are achieved using the parasitic-aware particle swarm design and optimization approach. The optimized circuit has a much better frequency response than the un-optimized one, and the optimizer has cleverly regained phase linearity by eliminating internal parasitic mismatch effects (Fig. 4). Loss compensation and improved phase linearity also drastically improve the gain flatness as shown in Fig. 3. The gain roughness in the passband before optimization was greater than ±2.4dB, but after optimization it was reduced to less than ±0.5dB. Optimized results are summarized in Table III.

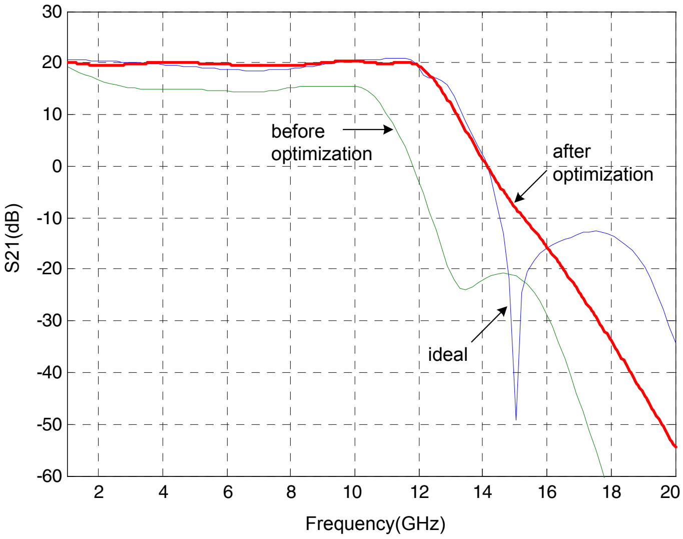

Fig. 3. Forward gain (S_{21}(dB)) magnitude results.

IV. SUMMARY AND CONCLUSIONS

We have described the first use of particle swarm optimization in RF circuit synthesis. A wideband distributed amplifier has been synthesized in a 0.25µm CMOS process using the new particle swarm approach. After extending an ideal design to include parasitics associated with passive components, the performance degradation caused by phase mismatches, and capacitive and resistive losses, was intolerable. In order to circumvent this problem by effectively re-tuning the RF circuit, a highly efficient custom CAD tool based on particle swarm optimization was developed and applied to the design of the wideband amplifier. The final results were impressive and included a flat 19.8dB passband gain over a bandwidth of 12GHz. The parasitic-aware particle swarm synthesis methodology is promising for future mixed-signal and RF applications since it is an inherently parallel approach that is easily adapted for parallel execution on a multiprocessor or on a computer network.

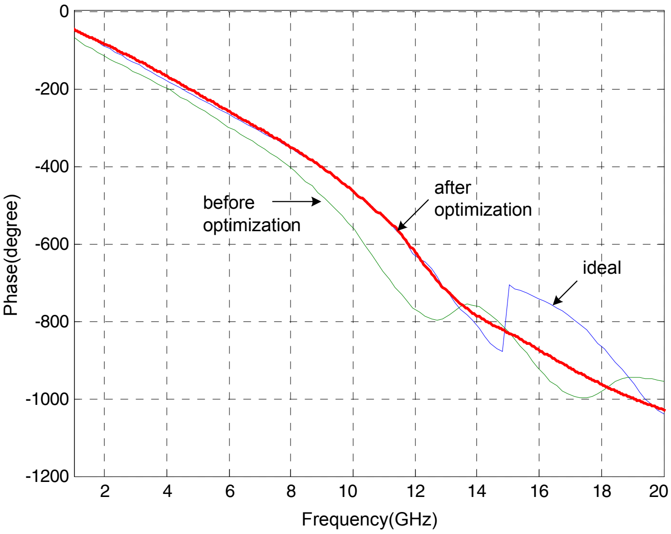

Fig. 4. Forward gain phase results.

TABLE III
Summary of CMOS RF Distributed Amplifier Performance Results using Particle Swarm Optimization

	After	Before	Ideal
Average Gain (S_{21} dB)	19.8	15.4	19.7
Gain Flatness (S_{21} dB)	±0.47	±2.49	±1.22
Std. deviation (S_{21} dB)	0.26	1.13	0.78
Average S_{11} in passband	-11.14	-21.2	-23.3
Average S_{12} in passband	-79.8	-87.4	-84.0
Average S_{22} in passband	-8.1	-20.0	-32.0
Flat gain Bandwidth (GHz)	0~12	4~10	0~12
Unity-gain Freq. (GHz)	14.2	11.8	14.2

REFERENCES

[1] E.L. Ginzton, et al., "Distributed amplification," *Proc. IRE*, vol. 36, pp. 956-969, Aug. 1948.

[2] R. Gupta, et al., "Parasitic-aware design and optimization of CMOS RF integrated circuits," *IEEE RFIC Symposium,* pp. 325-328, 1998.

[3] B.M. Ballweber, et al., "A fully integrated 0.5-5.5GHz CMOS distributed amplifier," *IEEE J. Solid-State Circuits*, vol. 35, pp. 231-239, Feb. 2000.

[4] M.J. Krasnicki, et al., "ASF: A practical simulation-based methodology for the synthesis of custom analog circuits," *IEEE/ACM Int. Conf. on Computer Aided Design,* pp. 350-357, 2001.

[5] E. Aarts, et al., "Simulated Annealing and Boltzmann Machines," New York: Wiley, 1989.

[6] R.A. Rutenbar, "Simulated annealing algorithms: An overview," *IEEE Circuits and Devices Magazine,* pp. 19-26, Jan. 1989.

[7] J. Kennedy, et al., "Particle swarm optimization," *Proc. IEEE Int. Conf. on Neural Networks*, vol. 4, pp. 1942-1948, 1995.

[8] D.M. Pozar, *Microwave Engineering*. Reading, MA: Addison-Wesley, 1990.

[9] D.G. Sarma, "On distributed amplification," *Proc. Inst. Elect. Eng.*, vol. 102B, pp. 689-697, 1954.

[10] R. Eberhart, et al., "A new optimizer using particle swarm theory," *IEEE Int. Symp. On Micro Machine and Human Science*, pp. 39-43, 1995.