Tools Assisted Analog Design, from Reconfigurable Design to Analog Design Automation

HONG, Yang

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

in

Electronic Engineering

The Chinese University of Hong Kong

March 2011

Abstract

Unlike the highly automatic flow for digital circuits design, analog design automation (ADA) is still far from mature. For mixed-signal applications, analog circuit occupies only a small part on the layout, but the design requires a considerable amount of time and effort, making ADA extremely attractive. However, there are a lot more considerations to cover in analog design flow than its digital counterparts. In addition, the ever downscaling IC means analog circuits have to face more and more small-size effects, insufficient modelings, and the inaccuracy of classic formulas, which are quite difficult to handle. To solve the problem, various tools and methods have been proposed, but all in a digital-like flow, which are trying to develop general algorithms to realize circuit and layout synthesis. Up to now there is still a lot of problems.

To solve these issues, in this thesis the consistent effort in developing a quick tools assisted IC design platform is presented. First, a reconfigurable solution is proposed for some analog/mixed-signal (AMS) system which requires flexibility to a certain extent, such as a reconfigurable RFID solution for different communicating distances. Second, for further demand of increasing the flexibility, a novel approach for ADA is presented, which provides a highly automatic design flow for analog circuits to realize the "SPEC (Specification) in, GDS out" goal. Considering all kinds of higher order effects and uncertainties under deep submicron or even more advanced technologies, reliable design and fastness in processing are the two major concerns instead of the traditional pure optimization for best performance. To get a good balance among performance, reliability and turnaround time, an Application-Specific design flow with in-built knowledge-based algorithms is applied to deal with ADA issues under advanced technologies, which can

quickly provide a reliable design with performance good enough to meet the SPECs for common use.

摘 要

相對於已經高度自動化的數字電路設計流程，如今的模擬電路自動化設計（ADA）程度仍然遠未達到成熟的地步。在很多常見的混合信號設計中，模擬電路在版圖中往往只占有很小的一部份，但相關的設計卻需要花費相當多的時間和精力。這使得模擬電路的自動化設計因其潛在前景而在當今顯得更有吸引力。但是，對於模擬電路的設計流程而言，其需要考慮的因素遠遠多於已有的數字電路。而且，隨著集成電路（IC）尺寸的不斷縮小，其模擬電路的部份不得不面對更多的問題，諸如各種小尺寸效應，並不充分的模型和越來越不精確的經典公式，這些都是很難解決但又必須面對的問題。為此，工業界和學術界提出了各種方法，開發了相關的軟件。目前總的來說，這些軟件和方法，都基於與數字電路流程類似的設計思路，即期望開發通用計算算法和引擎來解決問題，尤其是關鍵的電路和版圖的綜合問題。但到目前為止，這類方法在實際中仍然存在很多難以解決的問題。

爲了解決目前模擬電路自動化設計中的問題，尤其是處於適應小尺寸下 ADA 的需要，本文將對我們有關集成電路自動化輔助設計平臺方面的工作加以詳細闡述。首先是對於那些僅僅需要一定靈活性的模擬或者混合信號系統，例如為不同通訊距離所設計的一個可重構的射頻識別（RFID）系統，此時一個（相對應部份）可重構的解決方案往往是最佳選擇。其次，對於那些要求更多靈活性以至用戶全定制的應用，可重構方案往往不能滿足要求時，模擬電路的自動化設計就成為一個選擇。本文將具體呈現一個完整的模擬電路自動化設計流程，以實現“指標輸入，版圖文件輸出”（SPEC in, GDS out）的最終目標。考慮到現時小尺寸下的廣泛應用，在我們設計的流程中，設計可靠性和處理速度是除傳統的電路和版圖性能優化之外的主要考慮因素。爲了達到這三者的最佳平衡，我們提出一種依設計應用對象不同而不同，量身定做的專用流程結構，輔之以內建的以模擬電路知識為導向的算法

和引擎，用以處理小尺寸下的 ADA 問題。最終在確保性能達標的前提下，向用戶提供高可靠性的快速設計平臺。

CONTENTS

List of Figures

List of Tables

Acknowledgement

I am heartily thankful to all those who have supported me in both finishing this thesis and accomplishing the research work since the fall of 2007. This dissertation would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in preparation and completion of this study.

First and foremost, I express my utmost gratitude to my advisor, Professor Choy Chiu Sing, for his patient guidance, clear advice and supervision from the very early stage of this research. His email introducing Design Reuse and Reconfigurable RFID reached me even before my registration in CUHK. Every time I talked with him, I gained something new, greatly benefited from his knowledge, unique angle of view, optimistic attitude and encouragement.

Along with Professor Choy, I would like to thank Professor LEUNG Ka Nang and Professor PUN Kong Pang for their insightful suggestions and comments on my research. In developing the auto RFID and OPAMP platform, countless discussions with Professor Leung gave me lots of ideas from the big picture to detailed issues, triggering brilliant inspirations that are critical to the research. Besides the suggestions and discussions, their encouragement helps me a lot as the research on ADA is quite hard and bothers both academia and industry for decades.

In the long road of tools assisted analog design research, I must thank AI Yanqing for his generous help in developing the GUI interface and setting the whole structure for our products. Without his help, our proposed platform wouldn't be so friendly and flexible to

users.

Since the Reconfigurable RFID project starts, GUO Jianping, Ng Yuen Sum, Chan Chi Fat, Marco Ho, Shi Weiwei, Leung Lai Kan and Leung Mak Ki gave me all kinds of suggestions and related analysis that are quite helpful for my research. Their willingness to share ideas frankly and the hot discussions are all my precious memories.

Many thanks go to our laboratory technician, Mr. YEUNG Wing Yee, who has helped me a lot for both software and hardware issues. I am also grateful to Dr. JIA Jingbin, Dr. XIN Ling, Dr. ZHANG Min, Dr. He Xiaoyong, Dr. ZHENG Yanqi, Dr. Tang Siu Kei, LI Bing, Chen Yan, FAN Wen, TANG Xian, KO Chi Tung, LI Kwan Wai, LIU Zhouyi, ZHAO Yinsheng and other members in the VLSI and ASIC Laboratory for discussions and friendship. And I'm truly indebted to Mr. WANG Qian in Support Center of Synopsys China and Mr. Jason GAO from Cadence China.

Finally, I would like to thank everybody who was important to the successful realization of thesis, as well as expressing my apology that I could not mention personally one by one. All of you, as a whole, have made my time at CUHK so enjoyable that it has already become an indivisible part of my best memories.

CHAPTER 1. Basic Concepts and Objectives

1.1 A Cost-Driven Start

Unlike the mature and highly automatic flow for digital circuit design, analog circuits today are still largely being handcrafted with only a SPICE-like simulation guided by classic-formulas-based expertise. This problem goes even worse as feature size continues to downscale, leading to more and more nonidealities, all kinds of higher order effects [1], insufficient modelings [2] and the inaccuracy of classic formulas. All these issues pose a great challenge to analog designers, making analog design error-prone with long design and test cycles. As the cycle gets lengthy, the cost rises dramatically, causing the quick tools assisted analog IC design extremely attractive for shortening the time to market.

Up to now, various tools assisted analog IC design flows or platforms [3] have been developed to meet the different demands, which cover all the followings:

- Accelerating the time to market for analog circuits

  (e.g. Automatic design flow (digital-like) for analog circuits)
- Increase the yield for cost reduction, especially for the initial taped-out chips

  (e.g. *design reuse and shared-use through process or design settings*)
- Multi-Objective design on schedule

  (e.g. *meet different SPECs with different configurations in one product*)
- Facilitating the process migration

  (e.g. *parameterized process information as input*)

As shown above, obviously, quick tools assisted analog IC design will play an important role in the future of Analog & Mixed-Signal (AMS) design. To meet this demand, many

methods and approaches have been proposed, hoping to be the solutions for quick analog circuit design. But so far, there is still a long way to go as many issues become the serious obstacles that researchers have to face.

1.2 Issues and Ways in Tools Assisted Analog Design

To make traditional handcrafted analog design flow fast and automatic, many approaches and some related Computer-Aided Design (CAD) tools platforms have been developed.

In sum, all these prior work and the huge literature on related researches available now can be divided into two groups: Reconfigurable Design and Analog Design Automation (ADA). The former has some settings in process, design, or layout to make the product adjustable before layout, in layout or even after layout and tape-out. This approach is relatively simple with few novelties in design flow or the algorithms/engines for both the circuit and layout synthesis. As a result, it has been used in some applications that only require limited and predictable flexibilities, providing some pre-defined discrete choices. Obviously, its drawback is the highly limited flexibility and little extensibility for either process migration or other different applications. In addition, to achieve the limited flexibility, some pre-defined switches or the like must occupy a certain design overhead, causing a large chip area and the unavoidable switch parasitic effects.

On the contrary, the latter approach, ADA, is much more complicated and demanding than the previous one. It aims at establishing a similar design flow as the digital one, which is highly automatic and requires few manual interventions. However, though it can provide almost enough flexibility for analog applications in principle, so far it's still very difficult to get such a flow with satisfying design quality within acceptable time period.

The generalized algorithms of the circuit & layout synthesis engine are the critical points, which so far are quite difficult to translate many implicit ideas/considerations (which analog designers rely highly on to design the circuit and draw the layout) into a clear mathematical form. In addition, many unfriendly or boring settings are used, which need manual inputs to guide the flow. These settings are often quite incompatible with analog designer's working style. All the above shows that it's still a long way to go for fast automatic analog IC design with highly-matured CAD tools.

1.3 Our Objectives

1.3.1 Reconfigurable Design Platform for RFID tag

In our project of passive UHF RFID Tag design, to make the tag adaptable for wide applications, the tag must be able to communicate with the reader in different distances. As the AMS parts take a large part of the whole chip, the traditional design for tags in the manual manner is quite time-consuming. Don't mention the tag design for different communication distances. Besides the time issue, the tag must be as compact as possible to lower the cost and get competitive.

The requirement on compactness rules out the approach that provides limited flexibility due to designed hardware-based switches on chip, and the time-to-market concern asks for a fast automatic flow. As a result, a full-custom reconfigurable design flow has been proposed for passive UHF RFID tags.

Unlike the traditional hardware-based methods, a software-based (SKILL^{TM}-based) method is proposed. The main engine programs are written in Perl or Shell language to generate SPEC-dependent layout codes in SKILL. In this way the different tag layout is

achieved for different communication distances with no switch or the related parasitic effects. For compactness, as the user-input communication distance varies, we use unit-capacitor layouts instead of traditional big capacitor arrays to realize needed capacitance, which is regular and small enough to fill the possible vacant places so as to make the total layout both compact and regular. Related routings are also pre-defined and controlled by the program in a rule-aware manner to pass the following physical verification check. The total flow starts from user-input distances and automatically generates the GDSII file with the corresponding layout and physical verification results readily provided.

1.3.2 Application-Specific Design Assistant for op amp

For our future goal of achieving ADA for some large analog systems, e.g. Switched Capacitor Filter (SCF) and Analog-to-Digital Converter (ADC), one of the key components that need quick automatic design is op amp, a typical analog circuit.

Although there are lots of prior endeavors on analog design automation, especially on op amp design, many critical issues are still left unsolved or in doubt, which includes:

1) Pure optimization for theoretical optimal performance before layout generation, seriously undermined by all kinds of higher order effects, nonidealities and all the uncertainties that may come from parasitic parameters extraction and post-simulation.

2) Simple linear combination form for cost function with weights not easy for user to set. This style of settings is not readily compatible with design expertise at all, which doesn't reflect the thread analog designers have in dealing with real problems. For example, in GUI interface, users are usually asked to enter weights for area and aspect ratio, which are usually represented by some typical numbers like 1 or 3. But, neither can

the design expertise quickly give such precise settings nor can the tools tell user what the exact difference between 1 and 3 is. Are these coefficients based on a linear scale, log scale or something else? And how does the cost function achieve normalization for each figure of merit? If these issues are not transparent to users, it is almost meaningless. And even they are, this style doesn't fit the traditional design expertise well since designers usually take a serial thinking mode rather than a parallel one as they usually have strong design inclinations (but not quite clear in relative weights' ratios).

3) No detailed further classification of each figure of merit, causing the optimizing targets may deviate greatly from what analog designers expect. For example, the total routing length is far less important than the critical routing length/quality in designers' mind. But nearly all the literatures use the former instead of the latter.

As a result, to build the automatic design flow for SCF or ADC, we must have such a design flow for op amp. And to make it fast and high-quality, it must be highly compatible with design expertise, reflecting what analog designers consciously do in a good order. So, the cost function must guide the flow in a serial mode rather a parallel one when considering all the related figures of merit. To exactly show what analog designers care, detailed classification of each figure of merit has to be done. These are quite different to the traditional methods. Finally, to fully optimize the pre-sim circuit sizing and layout PR issue with acceptable turnaround time, application-specific engines based on design know-how is the only choice.

Therefore, our proposed method uses application-specific engines with build-in knowledge-based algorithms. The arbitration and optimization mechanism goes in a serial way with further classification of each major figure of merit. Powerful module generators

must cover all the analog layout rules of thumb for guaranteed layout quality and record all the I/O port-related information so as to clarify the beginning and ending of each global routing. For the possible optimization for aspect ratio, module layouts must have certain capability to reshape, e.g. transformation by decomposing into smaller units and forming stacked structures. All these novelties distinguish our proposed method from the traditional ones, with both short turnaround time and good quality achieved.

Similar with the reconfigurable RFID tag project, programs are also written in Perl and Shell languages to output SKILL^{TM}-based layout codes, making the system ready for any future update or migration. And application-specific flow sets an open structure, allowing any addition of similar specific flow for further use.

CHAPTER 2. Introduction to Reconfigurable Design

For the applications requiring limited flexibility that usually provides discrete choices, reconfigurable design is usually the first choice for industry. It often carries (or can be turned into) different configurations of the design to meet different SPECs, which is very helpful to develop a series of products to enrich the product line. Moreover, several different configurations of a product greatly improve the yield in mass production and shorten the time of fine-tuning at the beginning. To get such a reconfigurable design with several different configurations, different methods starting from process concerns to design ones have been proposed, which are described in Section 2.1 and Section 2.2. Their pros and cons are concluded in Section 2.3 as the summary.

2.1 Realization by Process

One method of reconfigurable analog design is to make metal-mask configurable, which is mainly realized by process (together with some necessary circuit considerations). The idea is to make the upper (usually top) metal routing flexible, full of options to realize distinct circuit structures based on the same FEOL (Front-End-Of-Line in process, before local/global metal layer formation) layouts. An example [4] is shown in Fig.1 as follows.

As shown in Fig.1, the design flow consists of two parts: first choose the design scenario, and then comes the optimization with recourse, which determines the configurable metal mask and related metal layer manufacturing in BEOL (Back-End-Of-Line) process.

Obviously, this approach has several drawbacks. First, the number of design scenarios is highly limited & discrete since they have to base on the same FEOL layout and the optional routings are also limited. Second, the FEOL layout must provide enough basic components to fulfill different tasks. As a result, if more flexibility is need, FEOL layout must be large enough beforehand, which may turn into a great waste for certain cases. And third, since FEOL layout is fixed beforehand, the room left for further BEOL routings is also correspondingly fixed Under such constraints, it's quite difficult to guarantee good routing quality for each kind of applications. If only the top metal mask is configurable, the situation will go even worse.

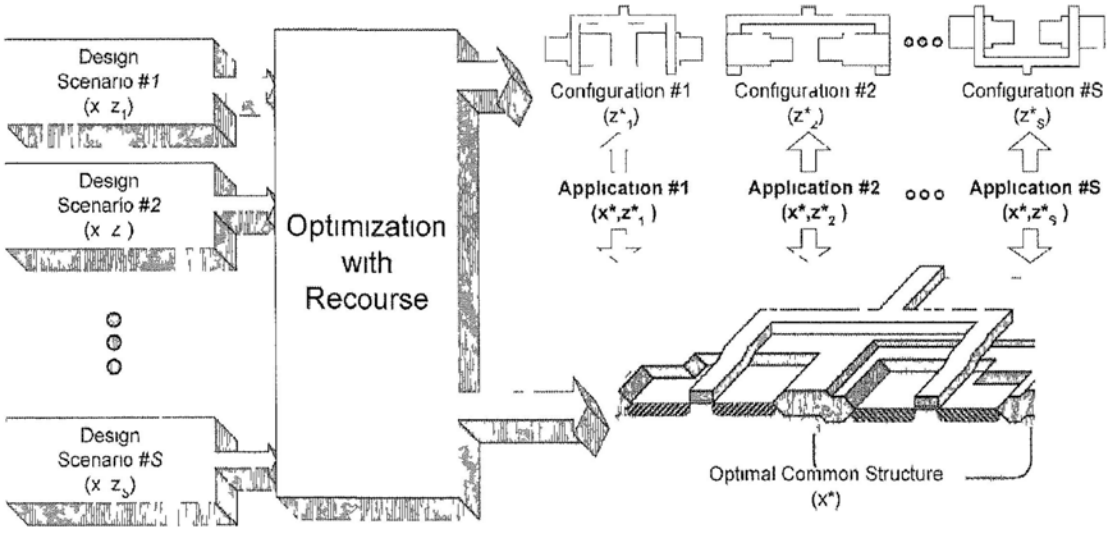


Fig 1 Reconfigurable Design configurable metal-mask

As Table 1 shows, once reconfigurable layers are only a few or just one, the optional change in basic components will be highly limited, usually only the number of unit being tunable. Thus, the common layers, esp FEOL layout, must be large enough to cover all kinds of possible needs Herein, if MOS FET or SiGe BJT is used, then large area reserve

for these transistors is compulsory, which means it will be a large waste for many cases that don't utilize all the units. Moreover, the unused ones are often set as capacitors. Since the gates are connected beforehand (usually connected by poly-silicon, which belongs to the common layer), the additional "parasitic capacitance" on gate should not be neglected.

Table 1 Metal-Mask Configurable RF Components

RF circuit components	Common layers	Customization layers	Customization unit
Spiral Inductors	Poly Ground Shield	M6	Quarter (0.25) turns
MIM-Capacitors	Deep trench perimeter	M5, M6, V56	Unit cap and No. of units
Resistor	S/D implant or Poly-silicon	M4, V34, V23, V12	No. of unit resistor
SiGe BJT	BJT unit devices	M4, V34, V23, V12	No. of unit BJT
MOS FET	MOSFET unit devices	M4, V34, V23, V12	No. of unit MOSFET

2.2 Realization by Design

Besides the above reconfigurable realization by process, another method to realize the quick analog IC design is through the effort on design.

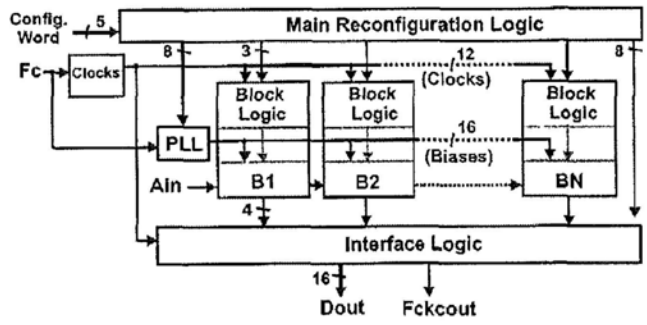


Fig. 2 A reconfigurable ADC structure

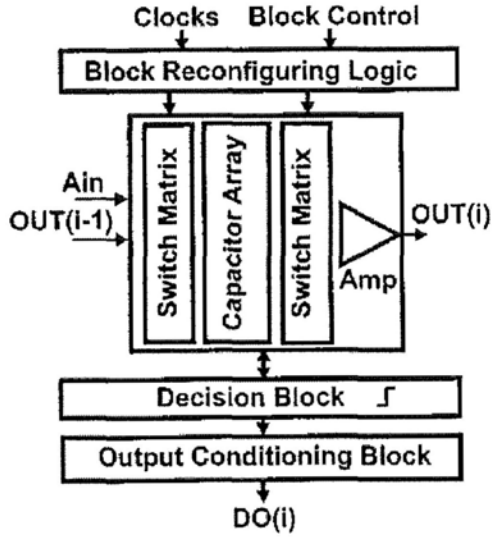


Fig. 3 Structure of a basic building block

Fig.2 and Fig.3 [8] show the structure of a reconfigurable ADC structure and a basic building block. For the critical reconfigurable parts, such the "block reconfiguring logic" shown in Fig.3, it's realized with a configurable switch matrix, which is responsible for providing the flexibility.

Obviously, this approach to get the design reconfigurable relies heavily on the design of the reconfigurable control logic parts and the characteristics of the application itself. The former will unavoidably introduce the parasitic capacitance and resistance due to the switch parasitic, which leads to performance degradation to some extent. And the penalty for this switch logic as the overhead needs a careful evaluation, which is a tradeoff between flexibility and compactness. In addition, this approach highly relies on the characteristics of the applied applications, which further limits its usage.

2.3 Pros and Cons

According to all those reconfigurable methods above, we can see that basically the basic methods mentioned above, together with the combined solution, have the following pros and cons:

Pros: relatively simple in ideas.

Cons: limited flexibility, which is also highly dependent on specific applications

Specific requirements on process (process-led configurability)

Poor in extensibility

Poor in compactness, large area reserved just for certain cases

Performance degradation due to the switch-induced parasitic effects

No effect on easing the possible process migration

For the mentioned methods (and the possible combined ones) above, poor compactness and performance degradation due to switch-induced parasitic effects are quite obvious for those carrying optional parts into tape-out. In addition, these hardware-design-based or metal-mask-dependent methods are neither readily revised for different applications nor easy to maintain.

To solve this problem, a novel software-based reconfigurable design method is proposed in Chapter 3 with a reconfigurable passive UHF RFID tag platform being shown, which carries no overhead layout into tape-out but maintains the design flexibility by software. It has no hardware-based switches and it's much easier to maintain or transplant than the hardware-design-based or metal-mask-dependent methods mentioned above. As a result, though still with finite flexibility, almost all the other drawbacks are eliminated.

CHAPTER 3. The Reconfigurable RFID Design Platform

3.1 Introduction to RFID

Ratio frequency identification (RFID) [9] is a technology of automatic wireless data collection with a long history [10]. From the point of view of small size, simplicity, and low cost, RFID tags are one of the most appropriate RF terminals for sensor networks in the various kinds of radio systems.

3.2 Why Reconfigurable Passive UHF RFID

3.2.1 Why Passive

As for RFID tag, it can be classified into three types: active tag, semi-passive tag, and passive tag as shown in Table 2 [11]. The active tag has some external components such as a battery and a crystal, and transmits ID data of the tag using the battery. The active tag supports long-distance communication and can supply power to sensors. However, the active tag has several disadvantages such as finite battery life, large volume, and high cost. The semi-passive tag has a battery and communicates to the base station (reader) using a backscatter method, in which the antenna impedance of the tag is modulated according to the data bit stream. This method is suitable for low power consumption because it does not require large transmission power. The communication distance of the semi-passive tag is over 10 meters since it operates with the battery, and some sensors can be included.

However, as it must have a battery (though a crystal is not necessary), it leads to higher cost compared with the passive one and a limited battery life that still needs to be solved.

Unlike the active and semi-passive tag, the passive tag can communicate to the reader without a battery by using transmitted power from the reader. In this mechanism, low-power operation is required for the passive tag. It communicates to the reader using a backscatter method, which leads to low power consumption since no large transmission power is required. Since it has no battery or crystal and there are no sensors included in the tags, cost can be lowered to a few cents with an acceptable communication distance of about 1–5 meters (now even 5~10m can be achieved [12]). As a result, the low cost makes passive RFID tags extremely attractive for many commercial applications.

As for its drawbacks, mainly the communication distance, to extend the distance, many studies of passive tags have recently been reported [11][12]. Schottky diodes are used in the rectifiers with a low forward voltage of 200 mV to communicate over long distances from small input RF signals, making the passive tags even more attractive.

Table 2 Comparison of three types of RFID tags

	Active tag	Semi-Passive tag	Passive tag
Communication distance	Long	Moderate	Relatively Short*
Necessity of Battery	Need	Need	No need
Cost	High	Moderate	Low

*It has been improved and still being improved by using low forward voltage transistors to generate voltage supply from small input RF signals. Now it's almost comparable to semi-passive tags in some cases.

3.2.2 Why UHF

Besides the classification of the three types based on energy transmission as above, RFID tags can also be categorized by operating frequency. For Low Frequency (LF, 125-134 KHz) and High Frequency (HF, 13.56MHz) applications, RFID systems are short range systems based on inductive coupling between the reader and the tag antennas through a magnetic field. But Ultra-High Frequency (UHF, 860-960 MHz) and microwave (2.4GHz and 5.8GHz) RFID systems are long-range systems which use electromagnetic waves propagating between reader and tag antennas. UHF RFID systems have several advantages compared to the LF/HF systems but their performance in general is more susceptible to the presence of various dielectric and conducting objects in the tag vicinity. Nowadays, near field UHF RFID receives a lot of attention as a possible solution for item level tagging in pharmaceutical and retailing industries [13].

Therefore, for massive applications of RFID tag in commercial applications such as the item level tagging in retailing industry mentioned above, passive UHF RFID tag becomes the hottest candidates. And to further lowering the cost and shorten the time to market for its design and test, a novel idea is proposed as reconfigurable design for passive UHF RFID tags, which is discussed as follows.

3.2.3 Why Reconfigurable

Fig.4 shows the diagram of a passive UHF RFID tag with no embedded NVM part (Non-Volatile Memory, such as EEPROM, ROM, Flash memory and so on). The tag gets the RF input signal from its antenna and the rectifier (voltage multiplier, usually using charge pump) generates the supply voltage. The unregulated supply voltage with ripples goes to

the voltage regulator (VR) before it is supplied to other blocks. ASK demodulator demodulates the input RF signal and send the data to baseband processor (BBP), which under the help of clock generator processes the input data and send the answer back to reader by backscattering the signal via ASK modulator and antenna. For lowering the power consumption, BBP will enter a sleep mode during the intervals, turning off all the unnecessary parts. To achieve this sleep mode, storage capacitors or NVMs will be used. However, either way will get the penalty on area since both the capacitors and memories are highly area-consuming as shown in Fig.5 [18].

At the same time, the SPEC for communication distance may vary greatly according to different applications, which will dramatically affect the size and related layout of the rectifier parts. And the rectifier parts can also be highly area-consuming if the tag is to receive a very small input RF signal in order to achieve a long communication distance. Under such cases, rectifier part has to have a lot of voltage multiplier stages (which means more layout area will be used) to generate a standard supply voltage when only receiving small input signals. In addition, the RFID tag for long communication distance may not work properly for short distance communication since the rectifier may generate higher voltages than expected, which will probably cause the following parts seriously damaged or deviated from what's expected. As a result, application-specific design for RFID tag is the best choice if possible.

Except the rectifier and storage capacitor parts, fortunately, few parts need to be reconfigurable (only backscatter part needs some flexible design or enough margins left, which is relatively easy and not mentioned in detail herein). So the VR, BBP, ASK

demodulator, clock generator and so on can be the general design for almost all the cases, greatly simplifying the task.

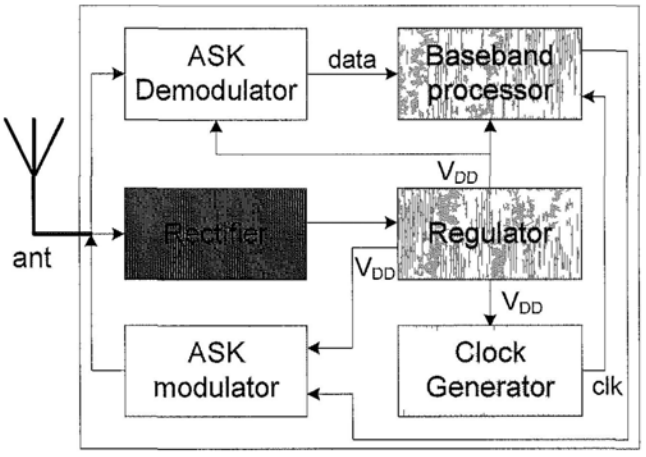


Fig. 4 diagram of a passive UHF RFID tag

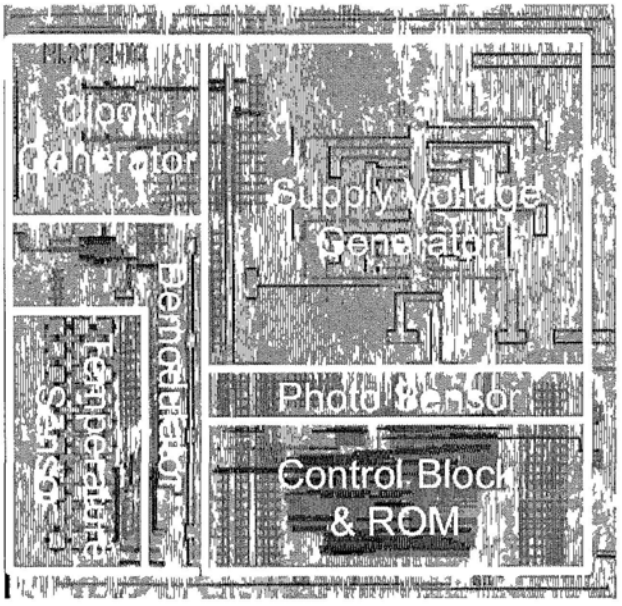


Fig. 5 Chip micro photograph of a passive RFID tag

In sum, according to different communication distances, the design and layout of the tag

may change dramatically, requiring a lot of time for design & test. And these different distances as SPECs, as well as the possible storage capacitors or NVMs, may cause the chip area to be large and volatile, totally case-dependent. But other parts are mainly the general design. Therefore, the focus is on the design of a fast & reliable platform that can provide case-dependent optimization (so as to cover the specific communication distance) for layout compactness (mainly for rectifier & storage capacitor/NVM parts).

3.3 How to Get Reconfigurable

As mentioned above, the RFID tag chip layout needs careful (even application-specific) optimization to get the layout as compact as possible and handle the high volatility of layout area due to different requirements on communication distance. The issues above actually originate from the following factors:

1) The rectifier (Voltage Multiplier, VM) part: highly area-consuming and highly dependent on the communication distance
2) The storage capacitors or NVMs part: highly area-consuming

Before solving these issues, we'd better look at what can be used and to facilitate the reconfigurable design.

1) The VM part: Though it's highly area-consuming and highly dependent on the communication distance, its possible change in structure and layout can be quite predictable and regular, which simplifies the related layout floorplan issue. The storage capacitors or NVMs part: for storage capacitors, its layout can be quite flexible to fit with other blocks as the aspect ratio for capacitor has little to do with the value. For NVMs, the memory array can also be detached into smaller ones

with fewer rows or columns. But as the rows or columns decrease, peripheral circuits as the overhead will get relatively bigger, leading to a low cell occupation ratio.

Therefore, for passive RFID tags which don't need to store too much information, storage capacitors are a better choice than NVM to increase the layout flexibility (also reduce the size since NVMs are usually larger). And what we need now is just a layout floorplan that can leave enough room for highly SPEC-dependent, area-consuming, but regular-changed VM parts and accordingly fit every possible vacancy by storage capacitors. Of course, the related routing and analog layout rule of thumb must be covered to guarantee the quality and reliability.

3.3.1 Reconfigurable VM

Fig.6 shows the circuit structure of a VM. Inside the dashed line is the first stage, having 2 capacitors and 2 diodes. Inside the dotted line and dashed-and-dotted line, there are the second and the final stages respectively, both having 2 capacitors and 2 diodes, just the same as the first stage.

As Fig.6 shows the regular structure of the VM, Fig.7 shows the corresponding regular layout structure for the VM. Inside the loop is the first stage layout. As unit layout puts I/O ports properly, the total VM layout can be achieved simply by placing all the unit layouts side by side, which further simplifies the floorplan.

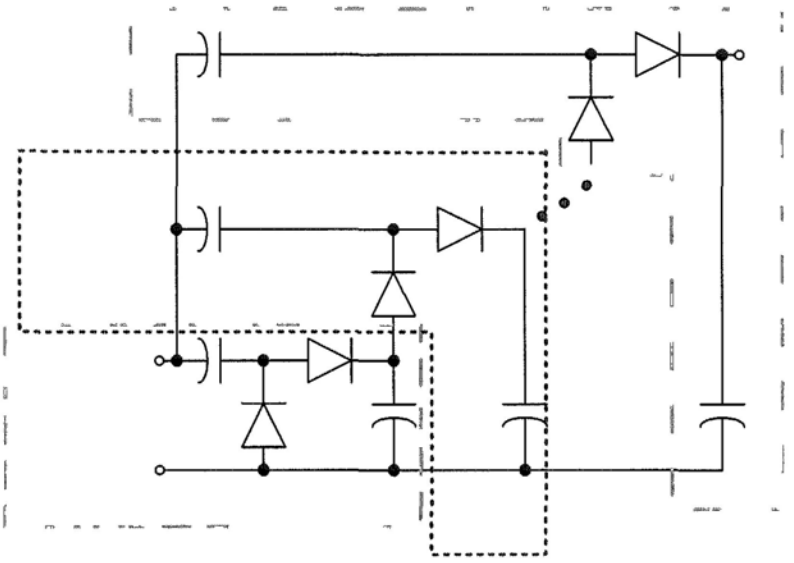
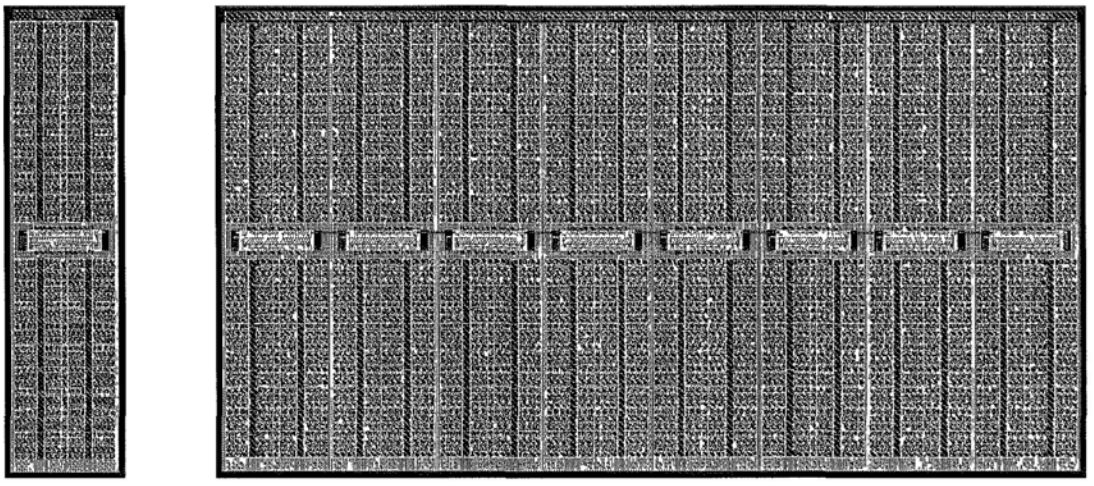


Fig. 6 Circuit structure of a VM



(a) (b)

Fig. 7(a) unit layout of the 1st stage VM (b) total layout of an 8-stage VM

3.3.2 Reconfigurable Capacitor Arrays

Since the VM part mentioned above changes dramatically but also regularly according to different input SPECs, namely communication distances, flexible storage capacitors have

to fit all the vacancies, esp. the vacant place caused by the change of the stage number of VM part.

Therefore, we divide all the capacitor parts into two kinds. One is the relatively fixed portion to fit all the vacancies among other fixed parts (VR, BBP, ASK demodulator, clock generator …) in RFID design. The other is the flexible one. To fit the possible vacancies left by flexible VM part, for this kind, the unit capacitor bears the same shape and size as the unit layout of one VM stage with in-built local interconnect routings. As the number of VM stages needed changes according to the input SPEC of communication distance, storage capacitors fully occupy the vacant place, guaranteeing both compactness and regularity for the total layout, as shown through the comparison between Fig.8 and Fig.9 that will be mentioned later in Section 2.4.1 and 2.4.3.

3.3.3 Relatively-Fixed Parts

As mentioned above, for reconfigurable passive UHF RFID tags to meet different SPECs on communication distances, not all parts are needed to be highly flexible. VR, BBP, ASK demodulator, clock generator parts are all generalized design. Even the ASK modulator that is responsible for backscattering can be designed in a conservative style leaving enough room for the whole range functioning. As a result, for all these parts except VM and storage capacitors (herein NVM is not used due to its large size), general design can be enough to cover the required configurability. In addition, as these parts are relatively fixed, so do the gaps between them, providing the room for fixed kinds of capacitors to fill in. Details will be addressed in section 2.4.3.

3.3.4 Reconfigurable Routings

As the VM part changes according to input SPEC of communication distances, its I/O ports and the according flexible storage capacitor part also change to achieve the best fit for compactness and regularity. As a result, the SPEC-dependent I/O ports, together with related routings, need to be taken into account automatically by our design platform.

Herein, as we use programs to generate the layout, the design platform will automatically calculate and record the information for each I/O port, top-level placement positions together with all the I/O ports' information. Under the guide of these data, the inter-part reconfigurable routings will be placed automatically with physical verifications followed to guarantee the validity. Detailed information will be shown in section 2.4.1~ 2.4.3.

3.4 Evaluation and Conclusion

3.4.1 Ideas and Algorithms

The proposed design platform for reconfigurable passive UHF RFID tags is based on UMC 130nm technology. The 900MHz UHF RFID tag is compatible to EPC C1G2 protocol. Herein no NVM is used due to its large size compared with storage capacitors. Instead, a storage capacitor is used for storing the charge to power the tag when little or even no RF power is available to the tag during backscattering. Due to the large leakage of the MOS capacitor in 130nm CMOS technology, herein we use a special MOS capacitor with 3.3-V I/O characteristic, which occupies less area than Metal-to-Metal capacitor and has much less leakage current than the standard NMOS capacitor in 130nm technology.

As for the VM part, Low threshold-voltage NMOSFET is chosen to minimize the conduction loss. Compared with the traditional design using Schottky diodes, the utilized approach saves the fabrication cost for extra masks the Schottky diode needs. Three (8-stage, 12-stage, 16-stage) VMs are designed as examples to show the flexibility, generating an output voltage of 1.3V from the worst-case of input peak-to-peak voltage of 180mV under the condition of 50-Ω input impedance.

As mentioned in section 2.3.1~2.3.4, for different applications, the major user-defined SPEC is the communication distance. As it varies, the structure of VM changes, that is, the number of stages shown in Fig.6 varies. As a result, the VM area changes greatly but regularly. This provides a possibility to utilize the layout flexibility of the MOS capacitor to fit the changing VM area for a compact rectangle-like total layout. And for additional capacitance that needs to add, we put them in the vacancies among other relatively-fixed parts, or put them on one/two sides to guarantee both the compactness and regularity.

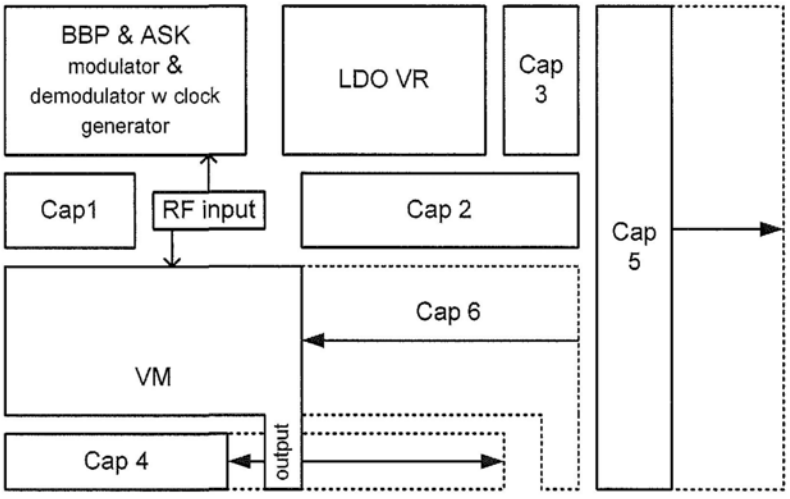


Fig. 8 Template floorplan with critical routings for reconfigurable RFID tag

Since it's optimized for 130nm technology with clear SPECs for general designs of many blocks, the rough relative size of each block is predictable. Based on these data and each block's characteristics, a template of the placement and routing (PR) can be attained as shown in Fig.8.

In getting the template floorplan shown in Fig.8, critical routings together with I/O ports distribution play a very important or even decisive role. Since the RF input signal is very small, it's quite sensitive to the possible noise. So the ASK demodulator part and the VM part have to be placed adjacent to the RF input signal as near as possible. Considering the relationship among VR, BBP & ASK demodulator/modulator and VM, VR has to be placed near both VM's output port and BBP & ASK parts. Due to the large size and possible extension (due to different input SPEC of communication distance) of VM, VR is better to lie on the same side of VM just as BBP & ASK parts do.

To make the layout compact for good regularity and low cost, the inter-block gaps are filled with fixed gaps, such as cap1, cap2 and cap3. As VM part may change according to input SPEC, cap4 and cap6 will extend or contract at the same (cap4) or opposite (cap6) pace as VM parts changes. And cap5 is the possible remaining part that is placed on one side (herein the right side) of the total layout, being adjacent to other capacitors for short connections and guaranteeing the regularity and good aspect ratio for the chip die.

In sum, the ideas of applied algorithms are:

1) Let regular VM expand or contract in a specific region which is also regular, easy to control and predict

2) As VM expands or contracts, flexible storage capacitors contract or expand, just in the opposite direction as VM. As its unit shares the same size as unit VM stage, the replacement is seamless.

3) Each unit part for both VM and capacitor carries its local routings and the inter-unit-block routings. Therefore, just side-by-side placement is enough.

4) For fixed parts and possibly-changed ports/parts, record the positions to do according routings.

5) For any additional part (if exists), lay it on one side of the chip to achieve good regularity. Of course, these additional parts should be made regular and fit with the nearby layout size before their placement.

3.4.2 Design Flow

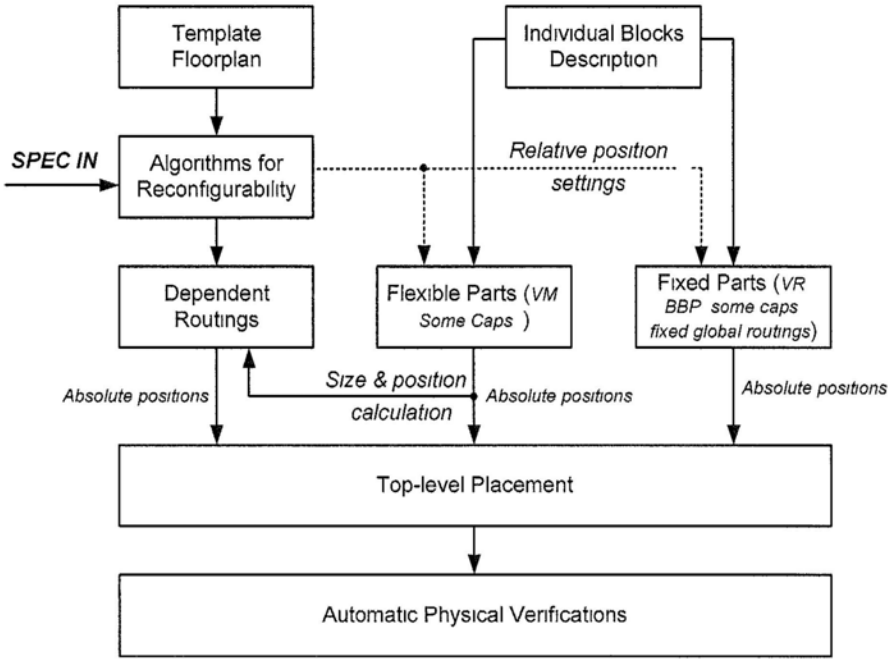


Fig. 9 Design Flow for the proposed reconfigurable RFID design

Chapter 3 The Reconfigurable RFID Design Platform

Fig.9 shows the design flow of the proposed reconfigurable RFID design platform. Input SPECs, namely communication distance or its range or the number of VM stages, tell the design platform what the flexible parts need to be (under the guide of in-built templates & design expertise). Then based on already available individual blocks, layout of fixed parts and flexible parts will be generated separately as instances with their sizes reported. At the same time, according to the template floorplan (rough placement), relative position for each block is set with some finite adjustments if necessary. Then under the guide of reported size of each instance (flexible or fixed) and the relative position settings of each block, the platform will do the top-level placement to ascertain the absolute positions for each block.

For example, if the platform program finds that input SPECs correspond to 8 stages for VM part, then cap 6 will consist of 8 unit capacitors (which has the same size as unit stage of VM). At the same time, cap 4 and VM output position can also be ascertained, providing the information for related routing. Now, according to total cap value and cap 4 & cap 6, cap 5 is known for the right side extension. Then the program will lay every instance, fixed parts (cap1~cap3, VR, BBP, ASK parts) and flexible parts (cap4~cap6 with VM and all the related routings). The detailed layout overview and the automatic physical verifications will be shown in section 2.4.3.

In essence, the proposed PR algorithm is both template-based and rule-based, guided by in-built knowledge-driven engines to cover design considerations (e.g. ASK & VM parts should be near to RF input) and analog layout rule of thumb (unit matching of caps …), thus guaranteeing the layout quality as well as the compactness and regularity.

3.4.3 Results and Evaluation

In our reconfigurable passive UHF RFID design, to lower the cost, VM uses the low threshold-voltage NMOSFET to minimize the conduction loss. Using this type of NMOS instead of the traditional Schottky diodes saves the manufacturing cost for extra masks that the Schottky diode needs. An output voltage of 1.3V is achieved from the worst-case of input peak-to-peak voltage of 180mV under the condition of 50-Ω input impedance.

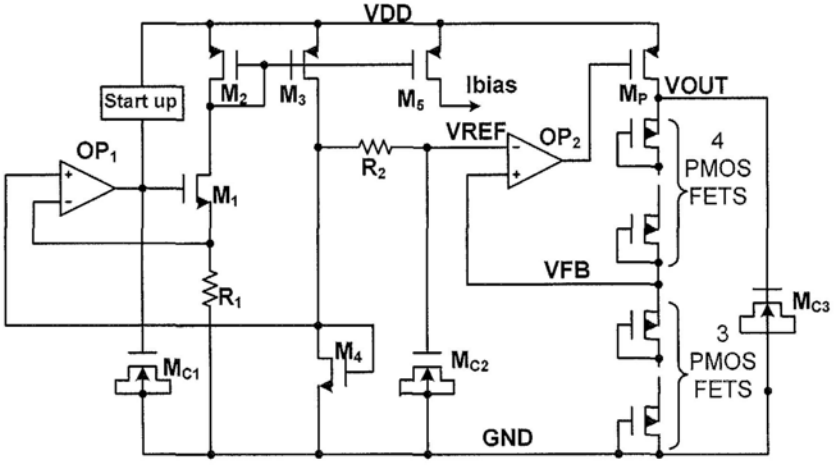


Fig. 10 Proposed LDO voltage regulator

For VR part, sub-1V operation with low temperature coefficient (tempco) and low quiescent current is achieved by using self-biased mutual thermal compensation as shown in Fig.10. The measured reference voltage is 470mV at 27°C with 25ppm/°C simulated tempco while the tempco of bias current is 550ppm/°C. A trimming to R_1 is introduced to get proper I_D with lower tempco if there is some process variation.

A 1-nF capacitor is used to store sufficient charge to power the tag when little or even no RF power is available to the tag during backscattering. Due to the large leakage of the

MOS capacitor in 130nm CMOS technology, a special MOS capacitor with 3.3-V IO characteristic is used, occupying less area than MIM capacitor with much less leakage current than the standard MOS capacitor. The measured output can be regulated to be 1.09V and has the maximum load capability more than 50μA, guaranteeing that all of the circuits can work normally without wrong operation.

As for ASK & BBP part, a five-stage current-starved ring oscillator is proposed with a frequency of 4MHz leaving enough margins to 1.92MHz, the minimum frequency required [18] to decode PIE data correctly according to the EPC C1G2 SPECs. In backscattering, modulate the input impedance of the tag when the RFID reader is sending a continuous wave to it. Since high input impedance of the tag reflects the continuous wave back to the reader while low input impedance absorbs the wave, a switch is added at the input of the tag IC. The difference in impedances is as follows:

$$Z_{diff} = (Z_{SW-OFF} // Z_{tag}) - (Z_{SW-ON} // Z_{tag})$$

Fig.11 shows the GUI interface of our proposed reconfigurable RFID design platform. User-input SPECs are the communication range (in meters), which corresponds to the number of VM stages. Herein, the number of VM stages is flexible from 8 to 16 for usual applications. After selecting communication range in the drop-down menu as shown in the figure, just click the "Generate Now" button to start the job. Once the layout is achieved, user can view the layout as shown in Fig.12~Fig.14. The optional automatic physical verification will be available as soon as layout is generated. Up to now, this design platform only supports two kinds of tools for physical verification, namely Cadence's Assura™ and Mentor's Calibre™. The platform can also be expanded to support other tools (e.g. Synopsys' Hercules) in the future if necessary.

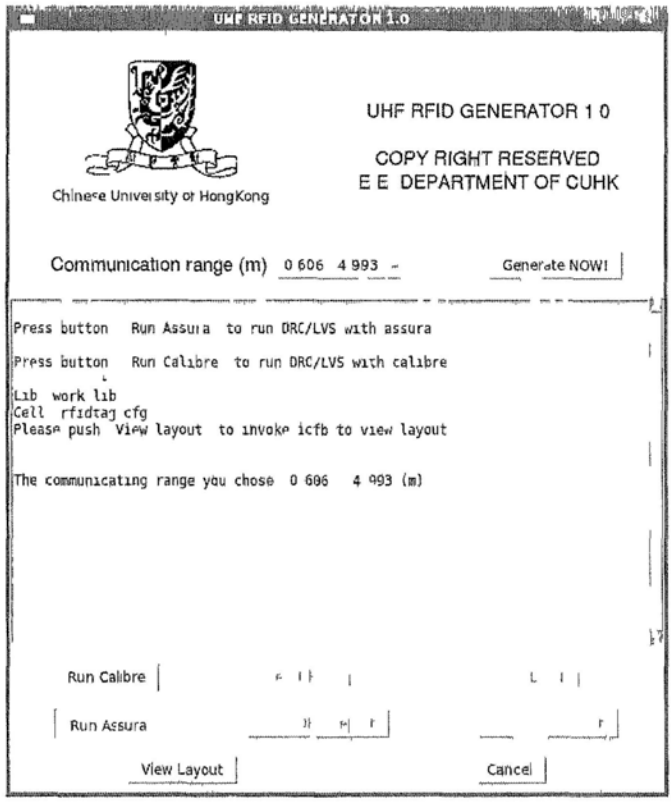


Fig. 11 UHF RFID Generator: Reconfigurable Design Platform for Passive UHF RFID

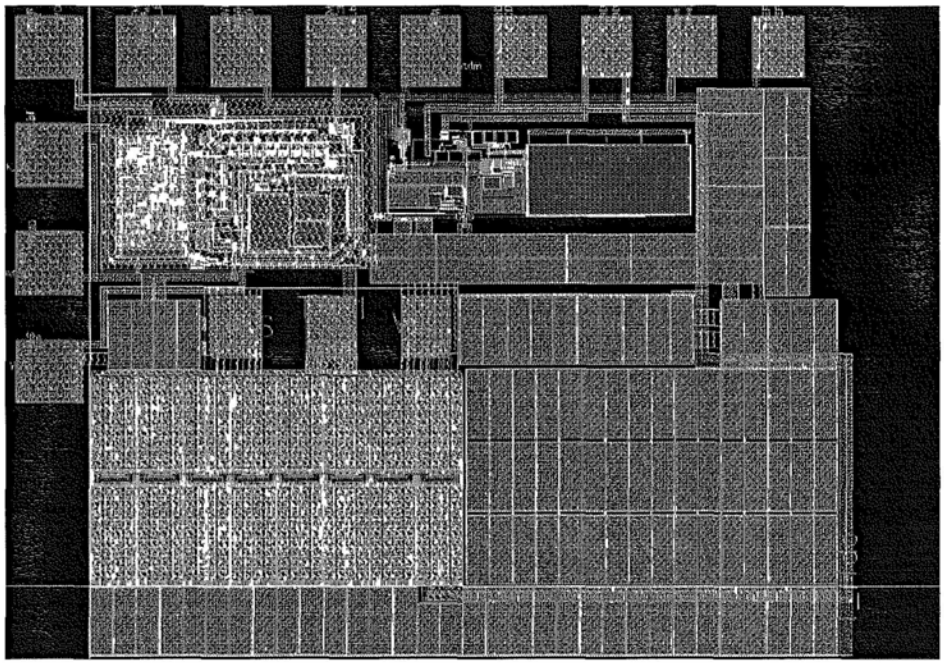


Fig. 12 Layout of RFID tag with 8 stages of VM

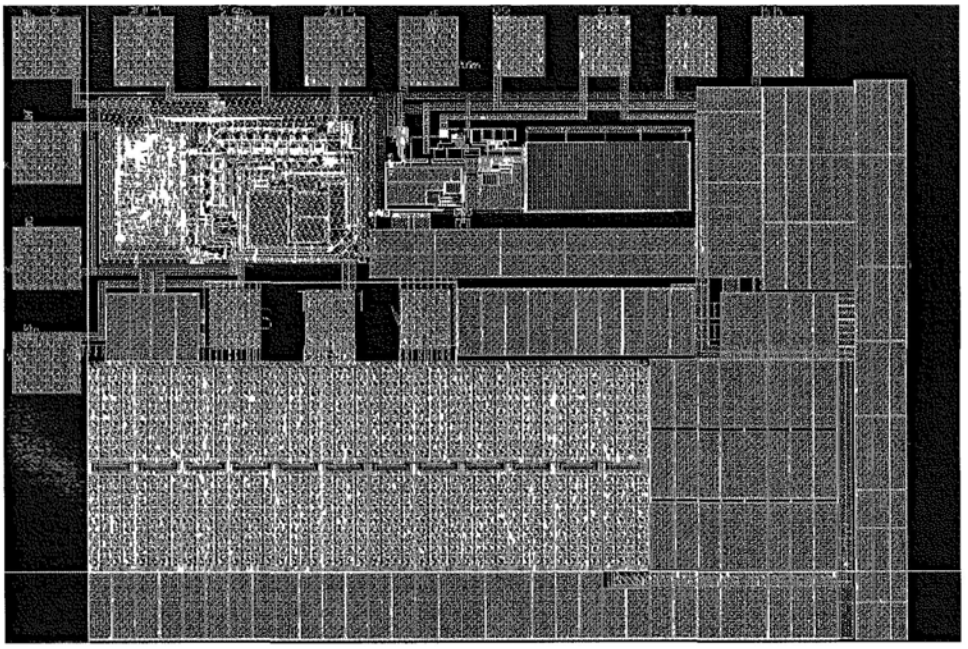


Fig. 13 layout of RFID tag with 12 stages of VM

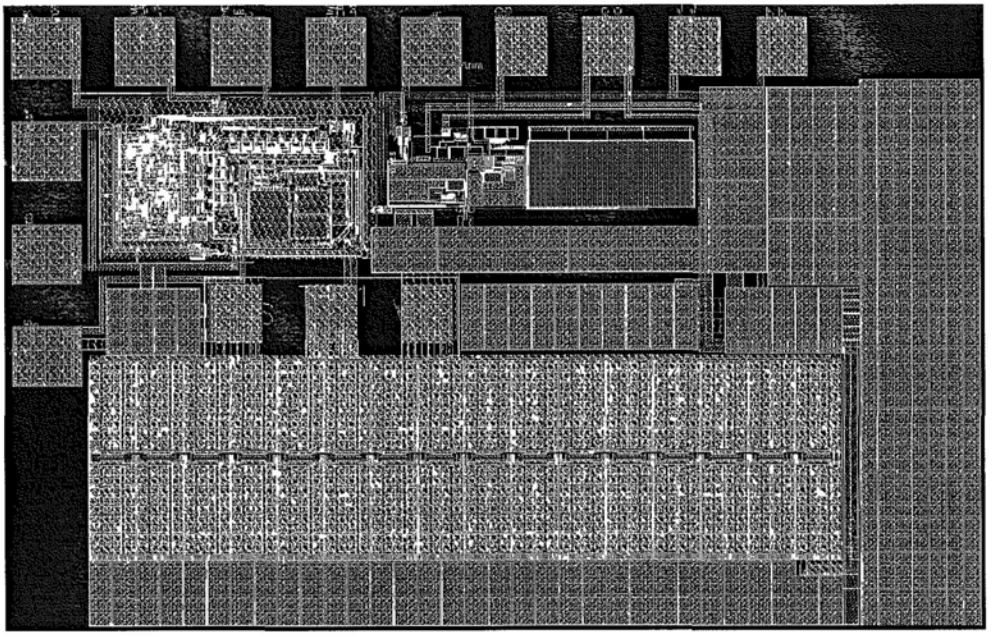
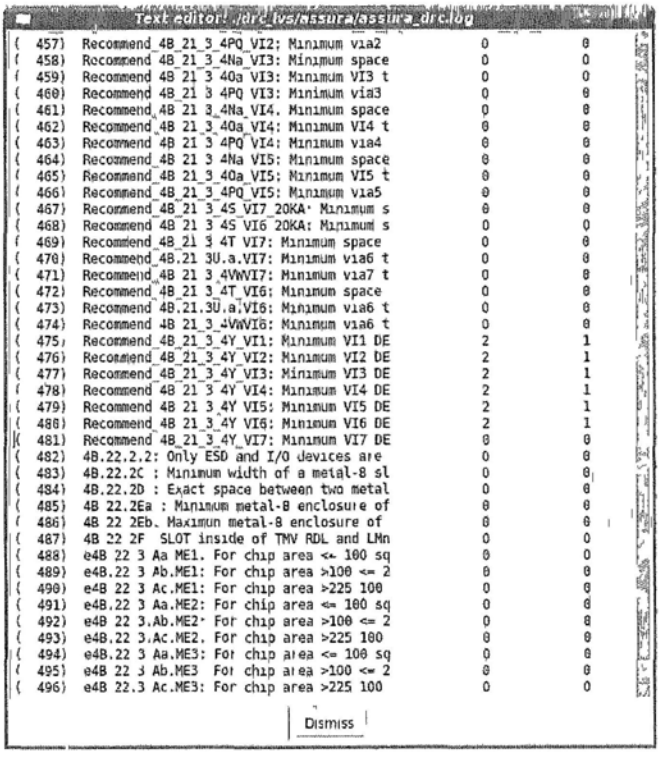


Fig. 14 layout of RFID tag with 16 stages of VM

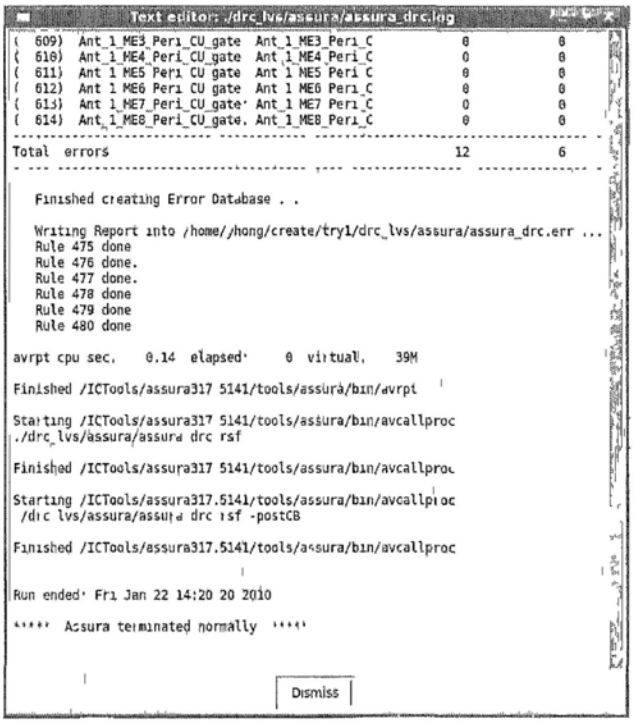
Fig.12~Fig.14 corresponds to the layout overview for 8, 12, and 16 stages of VM respectively. And the communication range of Fig.11 actually corresponds to Fig.13.

As shown in Fig.12~Fig.14, when VM part expands rightward as the number of stages increases, the originally-placed storage capacitor will be moved away, which has the same unit size (thus the same moving rate in area) as the unit VM stage. These moved capacitors will reappear on the right side, but in multiples of a new unit capacitor bearing a different aspect ratio but the same size/value. This arrangement is for optimizing the regularity of the total chip. Fixed parts, such as VR, BBP & some ASK parts, remain the same in relative positions together with their inter-block routings.

As for the flexible routings, all kinds of unit MOS capacitors have their in-built local routings, inner-block connections for global routings and relatively-fixed I/O ports, which means that the layout of any array only consists of all the unit layouts side by side with no top-level specific routings needed. This method greatly simplifies the routing issue, making the proposed platform even faster.



(a) Recommendation notes



(b) DRC summary report

Fig. 15 DRC report for generated RFID tag layout

Fig.16 LVS results for generated RFID tag layout

After layout generation, automatic physical verifications results are shown in Fig.15 (a), Fig.15 (b) and Fig.16. Obviously, with fixed parts and units of flexible parts already being silicon-proven, the total layout fluently passes the physical verification when the PR algorithms are inherently both template-based and rule-based with design expertise guided.

Fig.17 and Fig.18 show how to install our proposed design platform while Fig.19 and Fig.20 shows the situation of working directory before and after running the platform.

Fig. 17 How to install the platform (II)

Fig. 18 How to install the platform (II)

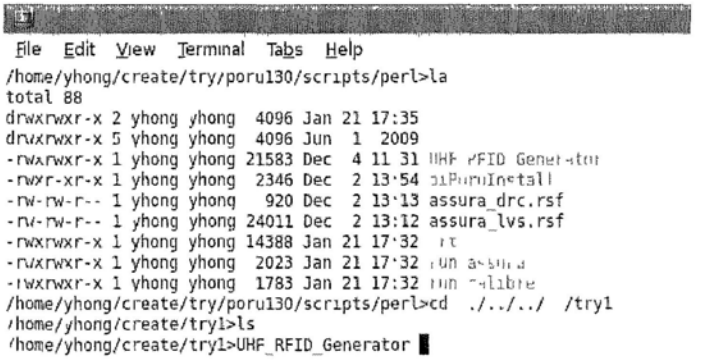


Fig. 19 Before running the platform

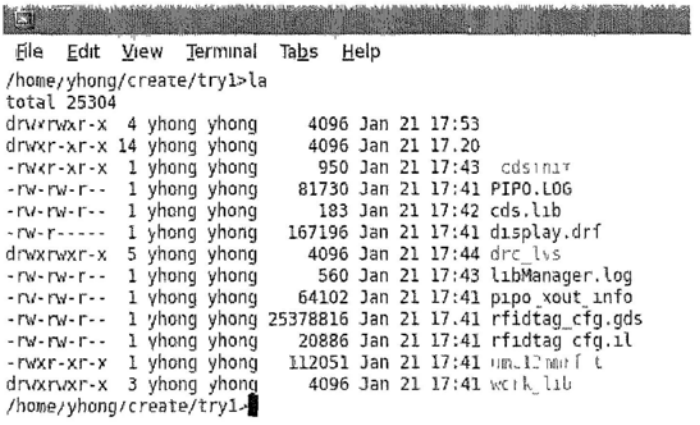


Fig. 20 After running the platform

As shown in Fig.20, after running the platform program, the output GDSII files (herein, it's file "rfidtag_cfg.gds") together with the related virtuoso layout library and physical verification checking results are all available. So far, the "SPEC in, GDS out" flow for Reconfigurable Passive UHF RFID design is accomplished.

Not including the peripheral PADs, the areas of the three layouts shown above are 1210x860um^2, 1300x860um^2, 1420x860um^2 respectively. As VM varies from 8 stages to 16 stages or vice versa, the area saved by flexible storage capacitors is about 590x320um^2, a maximum saving of 18.14% in area compared to the smallest one (with 8-stage VM). That means, if no such reconfigurable design is applied, to make the chip still flexible to

accept different communication ranges, the designed chip has to be 1420x860um^2 (not counting the area occupied by related switches) at all time. Moreover, since the reconfigurable design is achieved by software programming, the system can be quite open to make any further improvement without the flexibility-induced switch-related parasitic effects. And no special process or additional metal masks are required.

3.4.4 Testing Confirmation

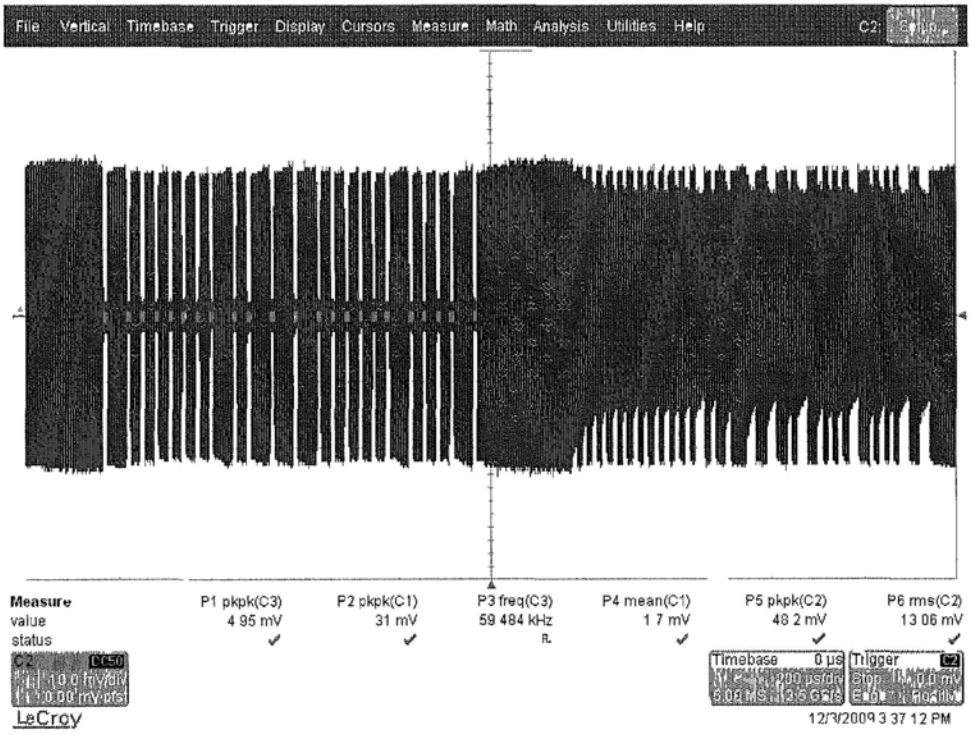


Fig. 21 backscattering of 8-stage cases: -45.2dBm as min input

Fig.21~Fig.23 shows the backscattering waves for 8-stage, 12-stage and 16-stage cases. And the minimum input signals are -45.2dBm, -45.2dBm and -32.5dBm respectively.

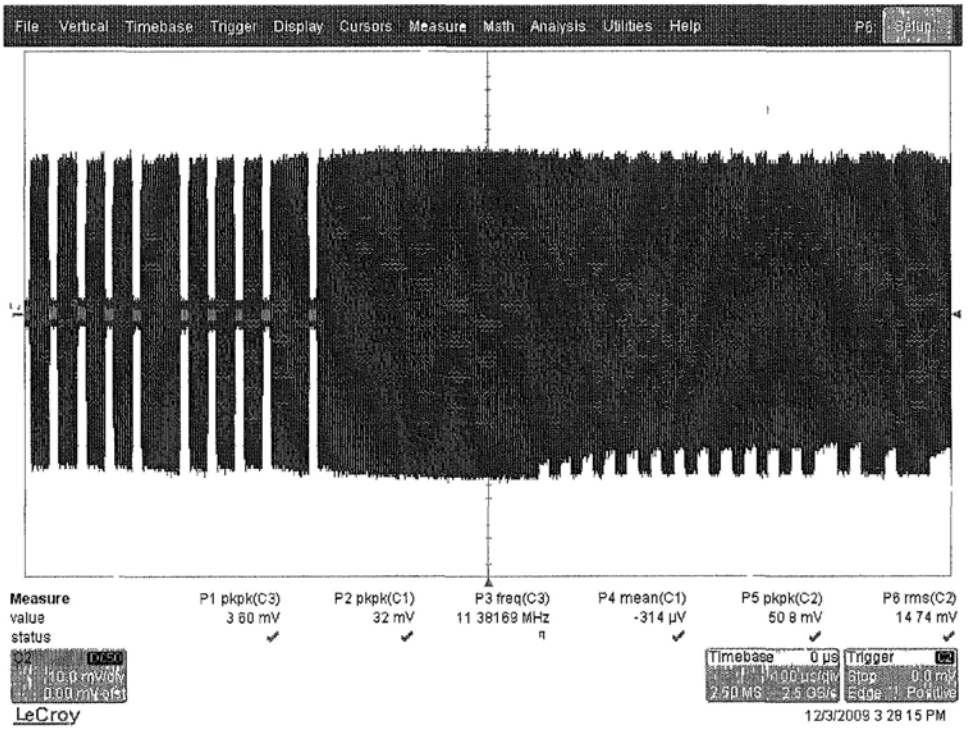


Fig. 22 backscattering of 12-stage cases: -45.2dBm as min input

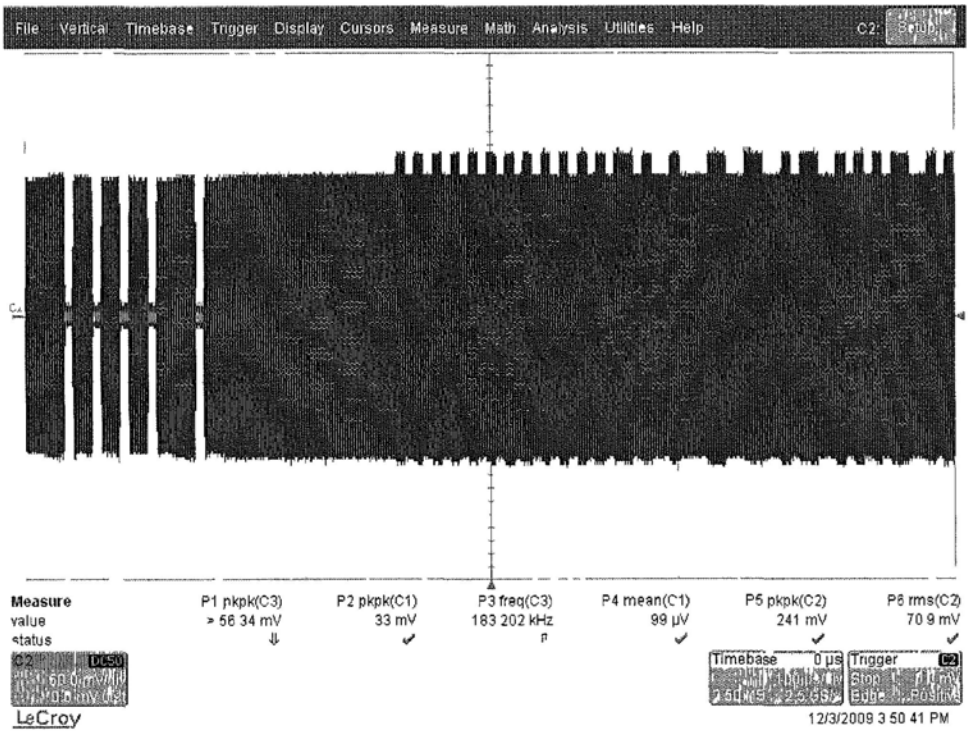


Fig. 23 backscattering of 16-stage cases: -45.2dBm as min input

Obviously, 8-stage case tops in backscattering modulation depth while 12-stage case being the second and 16-stage one being the third. All these clearly shows that due to the leakage and other disturbance issue, as the number of VM stages goes too large, the efficiency goes down, degrading the VM's sensitivity for maximum communication distances. As for min communication distances, corresponding input signals needed for 8, 12, & 16-stages are -25dBm, -23dBm, and -20dBm. The trend shows that for nearer field applications, due to the efficiency issue, large VM numbers are preferred. All these data, after formula-based conversion, are turned into ranges of application distances ready for user to choose in the drop-down menu shown in our GUI interface of Fig.11.

3.4.5 Essence and Conclusion

As mentioned above, our software-based design platform achieves a reconfigurable design flow from "SPEC in" to "GDS out" for passive UHF RFID tag applications.

Compared with the traditional reconfigurable design method, it doesn't need any special process or additional metal masks (compared with the process-related reconfigurable method). In addition, it needs no switches to achieve the reconfigurability, avoiding the flexibility-induced switch-related parasitic effects (compared with the design-related reconfigurable method). And since its reconfigurability is achieved through software programming and based on design expertise, the system can be very open and flexible to get enriched or improved to meet all kinds of reconfigurable demands with low maintenance cost.

In essence, our proposed design platform achieves the flexibility via software programs, keeping its flexible state (because software is usually much easier to change or maintain

than detailed design, layout or process settings with fewer constraints in extensibility) while integrating the design expertise with process information. Since it's for passive UHF RFID tag, a certain application, the design expertise we use can be highly specific and well targeted, which (with the process information) generates the templates of floorplan covering the considerations of both placement and routing. As a result, fastness and high quality coexist in the automation.

CHAPTER 4. Introduction to ADA

As reconfigurable design can only provide limited flexibility (usually pre-defined discrete ones), both industry and academia has already started the development of the automatic design flow for analog circuits. That is Analog Design Automation (ADA), the main idea of which is to achieve a similar automatic design flow as the digital one that is already quite mature. However, building the similar flow for analog circuits is quite difficult as analog circuits design differ much from its digital counterparts due to its characteristics. Following sections will discuss the prior work on ADA, its major obstacles and the drawbacks.

4.1 Traditional Structure

To solve the problems mentioned above, many CAD tools and methods have been developed and proposed to automate both the analog circuit design optimization and the corresponding layout generation. Although the methods and tools may vary greatly, the core idea is quite similar, that is, establishing an automatic analog design flow just similar to the mature digital one, esp. for PR engines. As a result, the critical parts go to mainly circuit synthesis and layout synthesis [1], focusing on the development of some general algorithms to solve the problem.

4.2 Circuit Synthesis

Circuit synthesis consists of two tasks: topology selection and SPEC translation. In essence, different analog circuit synthesis systems that have been explored so far can be classified based on how they perform topology selection and how they eliminate the degrees of freedom during SPEC translation or circuit sizing. It's a critical step since most analog designs require a custom design and the number of (often conflicting) performance requirements to be taken into account is large.

4.2.1 Topology Selection

In dealing with topology selection, several methods have been proposed. One is direct topology selection that user either directly select or modify an existing topology from any given library. Some tools, e.g. OASYS [21], BLADES [22], OPASYN [23], were rather heuristic in nature as they used rules in one format or another to select a proper topology (usual with hierarchy) out of a predefined set of alternatives stored in the topology library that the tools support. Another approach works in a more quantitative way that tools calculate the feasible performance space of each topology that can fit the structural requirements and then compare that feasible space to the actual user input SPECs to set the ordering of each topology. This method is tried using interval analysis techniques [24] or interpolation techniques in combination with adaptive sampling [25]. In all the above methods, topology selection is a separate step.

Differently, there are a number of some optimization-based approaches that integrate topology selection with circuit sizing as part of one overall optimization loop. This was done using a mixed integer-nonlinear programming formulation with Boolean variables

representing topological choices [26], or by using a nested simulated annealing loop where the evolution algorithm looks for the best topology and the annealing algorithm for the corresponding optimum device sizes [27]. Another approach that uses a genetic algorithm to find the best topology choice was presented in DARWIN [28].

Of all these methods, deterministic fashion is obviously attractive compared to the rather ad-hoc heuristic methods. However, as more and more small size effects show up (traditional classic formulas are no longer accurate), the feasible performance space for each topology is quite difficult to get with clear boundaries. At the same time, the possible overlap of feasible space of different topologies appears more frequently than ever as topology library gets richer and richer. As a result, the comparison in feasible space becomes increasingly difficult and time-consuming.

More important, as feature size downscales, process plays a more important role than ever. For experienced designer, the topology selection is quite clear and fast in his mind when getting a list of SPECs, making it unworthy for him to pay the lengthy time for running programs on topology selection. For beginner, their targets are usually common design. Heuristic approach asks too many detailed issues, which designers may not have the exact ideas right at the beginning. Meanwhile, the quantitative and optimization-based approaches are too time-consuming to be attractive.

Finally, as small feature size effects become more and more obvious, the design target is no longer purely optimal-performance-oriented, but caring both the performance and the design reliability. This makes the quantitative approach even more complicated especially when there are few knowledge-driven programs precluding poor candidates for simplicity and fastness.

4.2.2 Circuit Sizing

Once an appropriate topology is selected, the next step is SPEC translation, where the performance parameters of the sub-blocks in the selected topology are determined based on the overall SPEC (if there is any hierarchy). At the lowest level in the design hierarchy, this reduces to circuit sizing where sizes and biasing of all devices have to be determined to let the design meet user-input SPECs. This mapping from user-input SPECs into (preferably optimal) device sizes and biasing for a selected analog circuit topology in general involves solving the set of physical equations that govern the functioning of the circuits.

However, solving these equations explicitly is in general not possible. And analog circuit sizing typically results in an underconstrained problem with some degrees of freedom left. The two basic ways to solve for these degrees of freedom in the analog sizing process are either knowledge-based approaches exploiting analog design expertise and heuristics, or optimization-based approaches using optimization techniques [29][30].

1) Knowledge-Based sizing approaches: the first generation of analog synthesis system is knowledge-based, presented in mid to late 1980s. Specific heuristic design knowledge about circuit topology under design (including the design equations and the design strategy) was acquired and encoded explicitly in some computer-executable form, which was then executed in the synthesis for a given set of input specifications to directly obtain the design solution as shown in Fig. 21(a).

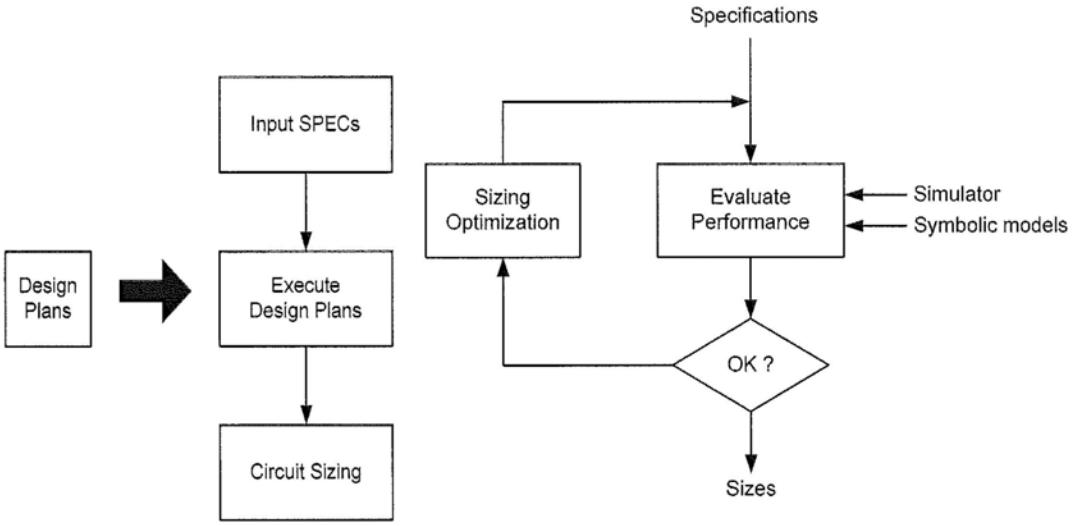
The IDAC tool [31], as a good representation of knowledge-based approaches in 1980s, uses manually derived and prearranges design plans or scripts to carry out the circuit sizing. The design equations specific for a particular circuit topology had to be derived

and the degrees of freedom had to be solved explicitly during the development of the design plan using simplifications and design heuristics. Once the topology was chosen by the designer, the design plan was loaded from the library and executed to produce a first-cut design that could further be fine-tuned through local optimization. OASYS [32] adopts a similar design-plan-based sizing approach where every block or sub-block in the library has its own handcrafted design plan, but the tool explicitly introduces hierarchy by representing topologies as an interconnection of sub-blocks. It also added a heuristic approach toward topology selection as well as a backtracking mechanism to recover from design failures. The approach was later adopted in the commercial MIDAS system [33], which was used successfully in-house for some types of data converters. Also, AZTECA [34] and CATALYST [35] use the design-plan approach for the high-level design of successive-approximation and high-speed CMOS data converters, respectively. Inspired by artificial intelligence research, also other ways to encode the knowledge have been explored, such as in BLADES [22], which is a rule-based system to do analog circuit sizing, in ISAID [36], [37] or in [38].

In all these methods, the heuristic design knowledge of an analog designer turned out to be difficult to acquire and to formalize explicitly, and the manual acquisition process was very time consuming. In addition, it has been reported [33] that the creation of a design script or plan typically took four times more effort than is needed to actually design the circuit once. This overhead to generate all this is quite large compared to a direct design, which is only worthy when that design plan will be frequently used. And finally, the coverage range is still not large enough. But it can be perfected in further development if the design effort is worthy enough.

Therefore, for knowledge-based approaches, the general design platform for analog circuits is obviously welcome if heuristic style can be avoided or rarely used. For design platform covering the analog circuit design across the board, the overhead of design effort may not be always low enough as certain cases may have few applications. And to get wide commercial use, speed is very important since up to now both knowledge-based methods and optimization-based ones are all to be used as a design assistant before ADA fully convinces the industry of its feasibility and quality.

2) Optimization-Based sizing approaches: In order to make analog synthesis system more flexible and extensible for various circuits, an alternative approach was followed since late 1980s. That's the optimization-based approach, using numerical optimization techniques to implicitly solve for the degrees of freedom in analog design for the optimal performance under the given SPEC constraints.



(a) The knowledge-based approach (b) the optimization-based approach

Fig. 24 The two basic approaches for analog circuit synthesis

Chapter 4 Introduction to ADA

There are several methods in numerical optimization. Categorized by the carrying-out forms, there are symbolic analysis techniques to automatically derive many of the design equations and the sizing plans, and a more equation-free simulation-oriented approach (as shown in Fig.24 (b)) to minimize the explicitly required design knowledge. Categorized by algorithms used for the optimization engines, which is more important, various methodologies have been proposed for optimization-based approaches as follows:

- Classical Optimization Methods: General-purpose classical optimization methods, such as steepest descent, sequential quadratic programming, and Lagrange multiplier methods, have been widely used in analog-circuit CAD. These methods can be traced back to the survey paper [39]. The widely used general-purpose optimization codes NPSOL [40] and MINOS [41], are used in [42]. LANCELOT [45], another general purpose optimizer, is used in [46]. Other CAD approaches based on classical optimization methods, and extensions such as a minimax formulation, include the one described in [47], [48] and [49], OAC [50], OPASYN [23], CADICS [51], WATOPT [52] and STAIC [53]. The classical methods can be used with more complicated circuit models, including even full SPICE simulations in each iteration, as in DELIGHT.SPICE [54] (which uses the general-purpose optimizer DELIGHT [55]) and ECSTASY [56].

  The main advantage of these methods is the wide variety of problems they can handle; the only requirement is that the performance measures, along with one or more derivatives, can be computed. The main disadvantage is that they only find *locally optimal* designs. This means that the design is at least as good as neighboring designs, i.e., small variations of any of the design parameters results

in a worse (or infeasible) design. Unfortunately this does not mean the design is the best that can be achieved, i.e., globally optimal; it is possible (and often happens) that some other set of design parameters, far away from the one found, is better. The same problem arises in determining feasibility: a classical (local) optimization method can fail to find a feasible design, even though one exists.

The problem of non-global solutions from classical optimization methods can be treated in several ways. The usual approach is to start the minimization method from many different initial designs, and to take the best final design found. Of course, there are no guarantees that the globally optimal design has been found; this method merely increases the likelihood of finding the globally optimal design, but also destroys one of the advantages of classical methods, i.e., speed, since the computation effort is multiplied by the number of different initials designs that are tried. It also requires human intervention (to give "good" initial designs), which makes the method less automated. The classical methods become slow if complex models are used, as in DELIGHT.SPICE, which requires more than a complete SPICE run at each iteration ("more than" since, at the least, gradients must also be computed).

- Global Optimization Methods: The most widely known global optimization methods are branch and bound [57] and simulated annealing [58][59]. A branch and bound method is used, e.g., in [60]. Branch and bound methods unambiguously determine the globally optimal design: each iteration they maintain a suboptimal feasible design and also a lower bound on the achievable performance.

This enables the algorithm to terminate non-heuristically, i.e., with complete confidence that the global design has been found within a given tolerance. The disadvantage of branch and bound methods is that they are extremely slow, with computation growing exponentially with problem size. Even problems with 10 variables can be extremely challenging.

Simulated annealing (SA) is another very popular method that can avoid becoming trapped in a locally optimal design. In *principle* it can compute the globally optimal solution, but in implementations there is no guarantee at all, since, for example, the cooling schedules called for in the theoretical treatments are not used in practice. Moreover, no real-time lower bound is available, so termination is heuristic. Like classical and knowledge-based methods, SA allows a very wide variety of performance measures and objectives to be handled. Indeed, SA is extremely effective for problems involving continuous variables and discrete variables, as in, e.g., simultaneous amplifier topology and sizing problems. SA has been used in several tools such as ASTR/OBLX [61], OPTIMAN [62], FRIDGE [63], SAMM [64], and [65].

The main advantages of SA are that it handles discrete variables well, and greatly reduces the chances of finding a non-globally optimal design. (Practical implementations do not reduce the chance to zero, however.) The main disadvantage is that it can be very slow, and cannot (in practice) guarantee a globally optimal solution.

- Convex Optimization and Geometric Programming Methods: The convex optimization methods excel in dealing with special optimization problems, in

which the objective and constraint functions are all convex. With the development of extremely powerful interior-point methods for general convex optimization problems in the last five years [66][67], these methods can solve large problems, with thousands of variables and tens of thousands of constraints. The extreme efficiency of these methods is one of their great advantages (in minutes on a small workstation).

The other main advantage is that the methods are truly global, i.e., the global solution is *always* found, regardless of the starting point (which, indeed, need not be feasible). And Infeasibility is unambiguously detected, i.e., if the methods do not produce a feasible point they produce a certificate that proves the problem is infeasible.

As already mentioned on analog-circuit CAD [68], these methods have several critical drawbacks as follows:

First, the types of problems, performance SPECs, and objectives that can be handled are far more restricted than any of the methods described above. As a result, general flexibility for wide applications is in doubt. Second, even for certain cases that these methods are already been used, SPECs and constraints meet the strict requirements (*posynomial or monomial form functions only for cost function & constraints...*) only when classical formulas are considered. That means, for small feature size, as all kinds of nonidealities and higher order effects become obvious, classical formulas with all kinds of corrections may no longer meet the requirements, making these methods almost useless. Third, throughout the related papers, few layout-aware constraints are available. And some ideas that designers

consciously do in analog layout are too implicit to translate into a simple & clear mathematical form. Don't mention meeting the requirement of posynomial form and etc.

In sum, both the traditional knowledge-based algorithms and optimization-based methods are far from enough, especially for achieving a generalized platform with good extensibility for all kinds of analog applications. As the final goal is to shorten the time to market and lower the cost, a design assistant is probably a better choice now as general algorithms & engines are hard to get with sufficient quality, reliability and extensibility. In addition, in doing the circuit synthesis, esp. the pre-layout synthesis, layout-aware design is critical and must be covered, which will play a more and more important role as feature sizes scale down with more high-order effects.

4.3 Layout Synthesis

Unlike the well-developed & commercially-available digital layout design flow, due to the special and necessary constraints imposed on analog layouts (e.g., large variation of MOS transistor sizes, sensitivity to parasitic capacitance, crosstalk, device matching, symmetry requirements, voltage drops, temperature gradients, piezoelectric effects, electromigration, etc.), analog layout design is intrinsically very difficult.

Due to the continuous downscaling in feature size, device electrical properties get worse in analog circuits due to more obvious non-idealities and all kinds of high order effects. As a result, the circuit synthesis needs to be more layout-aware than ever, and the layout synthesis has to face more special constraints. All these make the analog layout synthesis

even more challenging, especially when a digital-like highly-generalized engine is set as the target.

Up to now, several CAD tools [71] have been developed to automate the generation of analog layouts, which usually consist of three parts: module generation (MG), placement and routing (PR). Detailed analysis and comparison of these tools on MG and PR are described as follows.

4.3.1 Module Generator

Module generators are used to quickly realize the automatic layout drawing for circuit components (usually the classical ones that are often called as instances), which greatly accelerate the total layout drawing and simplify the total layout generation flow. It can cover the layout generation from single basic geometric shape (e.g. rectangular diffusion), basic component (transistor, resistor, capacitor ...), classical circuit structure (current mirror, differential pair ...), usual connection structure (layer to layer connection ...) to even an array of those components mentioned above (e.g. capacitor array, several current mirrors in parallel structure ...). With module generators & clear application of hierarchy, the higher-level PR algorithms & engines can be greatly simplified.

Up to now, many module layout generators have been proposed with considerations of analog layout rule of thumb, which is vital to guarantee modules' layout quality. Fig.25 [73] shows the generation of transistors as modules. Fig.25 (a) shows folded transistors: this structure accepts a single layout level parameter: folding amount. Node capacitances of the folded transistor are the sum of the perimeter and area capacitances of the contacts. Fig.25 (b) shows merged transistors, which are generated by abutting two transistors. Due

to geometric sharing of transistors, this structure has a more compact layout and reduces the parasitic capacitance.

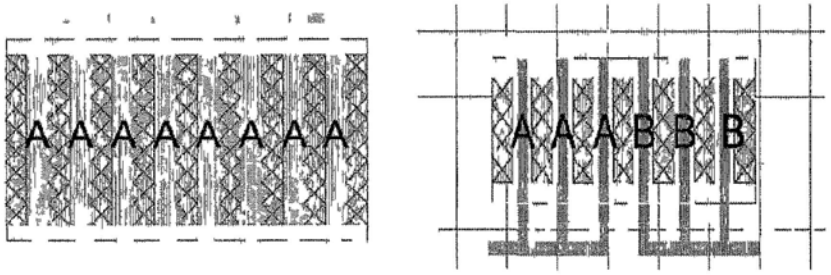


Fig. 25 (a) Folded transistor layout (b) Merged transistor layout

Fig.26 [73] shows the module layout for interdigitized transistors. In this structure, transistor folds are shuffled into each other, reducing average separation between transistors Fig.26 (a) shows two different distribution approaches. The first example has a recurring AB pattern, which means the centers of the gravity of A and B can never locate at the same place, not being a common centroid layout to be immune to linearly distributed disturbance. On the contrary, the second one is symmetrically distributed with the centers of the gravity of A and B being the same. Therefore, common centroid layout is achieved as the corresponding layout is shown in Fig.26 (b), quite helpful to minimize the mismatch

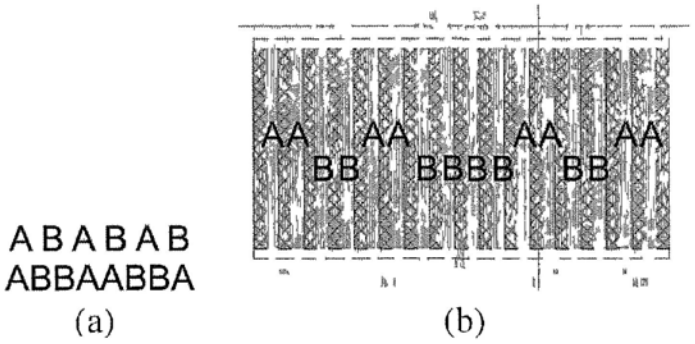
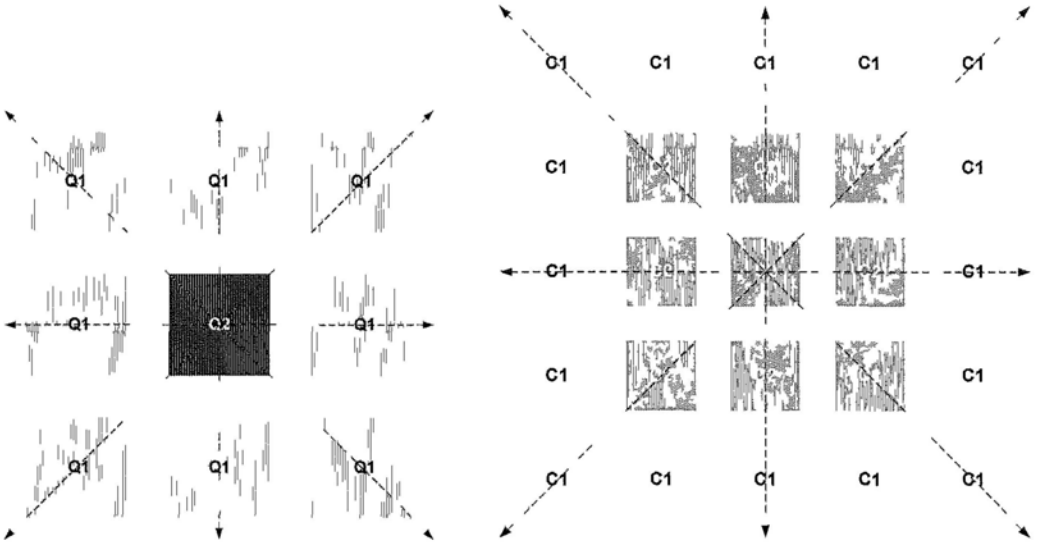


Fig. 26 Interdigitized transistor layout



(a) (b)

Fig. 27 (a) Common-centroid layout (bipolar) (b) Common-centroid layout (capacitor)

Similarly, for better matching, typical common-centroid layout for bipolar and capacitor are shown in Fig.27 (a) and Fig.27 (b) respectively. In general, the elements that need matching are categorized into several groups, each of which forms one or more "loops" consisting of specific amount of units (often squares, maintaining good symmetry & regularity for each direction).

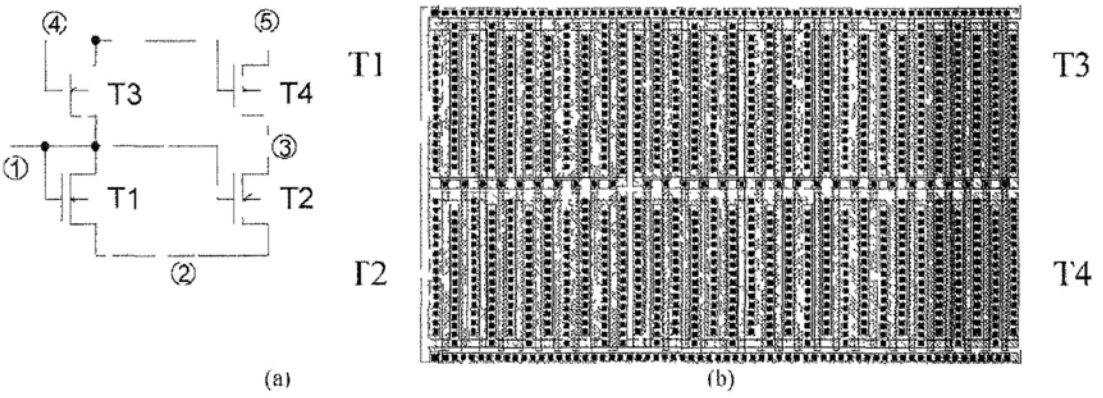
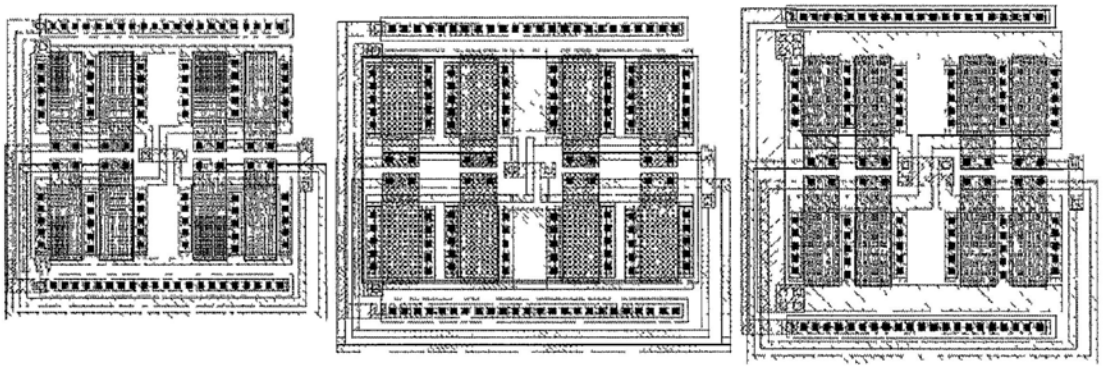


Fig. 28 (a) Schematic (b) the corresponding module layout of a cascode current mirror



(a) (b) (c)

Fig. 29 Different layout implementation of a differential pair

Fig.28 [71] (a) shows the schematic of a cascode current mirror and the corresponding module layout is depicted in Fig.28 (b). And Fig.29 [71] shows different available layout implementation of a differential pair.

As we can see from above, module generators can fulfill almost all the classical analog structures or basic components that are called frequently in usual applications. Obviously, with its help and the hierarchical calling, the PR issue of each hierarchy can be simplified, and total layout generation is greatly accelerated.

However, up to now, there are still some problems unsolved for these available module generators. First, some layout rules of thumb for analog design are still not covered in designing module generators. For example, when current mirror requires good matching, uniformity of currents' direction, as a layout rule of thumb, is not considered in any of the current mirror module generators mentioned above (interdigitized transistors are doomed to fail in achieving this due to the diffusion sharing and so on).

Second, the available module generators pay too much attention on shrinking the area compared with the care for layout quality. Analog circuits are usually sensitive & vulnerable to noises or other disturbance. So they will try to create a sound environment

usually at the cost of area since area for analog parts is often relatively small compared with the digital ones. As a result, many analog parts will have dummy parts to create symmetry and isolation, using conservative sizes or settings rather than the extremes that rule files can handle. But for the prior work, to our best knowledge, due to the great role that area plays in PR's cost function, the priority of layout quality over area is not obvious in the available module generators.

Third, so far, module generators pay little attention to the issues related to higher level PR. Efficient higher level PR algorithms and engines call for easy control and flexible arrangement of I/O ports, flexible readjustment of shapes for different requirements on aspect ratio. Also, module generators should use as few metal layers as possible to give global routing more options. In addition, besides covering the layout rules of thumb for good layout quality, module generators shall take local power/ground routings as more as possible to facilitate the coming global connection and isolation of possible disturbance and other uncertainties.

Finally, module generators are better to be capable of predicting its sizes & aspect ratios before the layout is generated. Otherwise, every change will re-generate the layout, which still doesn't know whether or not it can fit well with other parts until it's called. This will lead to more possible iterations, greatly slowing down the process and making ADA unattractive. And when user input is not complete (usually just focused on certain SPECs), module generators shall be able to detect user's intention and use acceptable values (usually build-in default values) as other unmentioned input SPECs to go through the flow quickly while still guaranteeing the quality.

In sum, module generators shall first care the layout quality, and then try to be both flexible and full featured (esp. for higher level PR needs). Finally it should be as human as possible to accelerate the process without too many heuristic inputs. All these are highly knowledge-based and application-specific. A rich library is needed before wide applications can be supported.

4.3.2 Cost Function for Placement and Routing

Cost function guides the following placement and routing algorithms and engines, which is vital for layout optimization. Traditional cost functions are usually a linear combination of all the major figures of merits with the corresponding coefficients being set either by user through a heuristic approach or by programs with default values and build-in algorithms as a guide.

In [74] (published in 1991), cost function is a linear combination of the following figures of merits: overlap, area, aspect ratio, net length, proximity and merge as shown in Equation (1).

$$Cost = w_0 \cdot \boldsymbol{Overlap} + w_1 \cdot \boldsymbol{Area} + w_2 \cdot \boldsymbol{AspectRatio} + w_3 \cdot \boldsymbol{Netlength} + w_4 \cdot \boldsymbol{Proximity} + w_5 \cdot \boldsymbol{Merge} \quad (1)$$

Where the $\boldsymbol{w_i}$ series are experimentally chosen weights. The ***Overlap*** term penalizes illegal overlaps (measured as overlap areas) among devices. As simulated annealing is used in placement strategy, it's not deterministic. As devices are moved during annealing, an illegal overlap occurs if protection frames overlap, i.e., two pieces of electrically distinct geometry overlap, or they are closer than design rules or wire space estimates allow. This term of the cost function must be driven to zero to ultimately produce a feasible layout. The ***Area*** term penalizes the total bounding box area of the placement

while the ***AspectRatio*** term penalizes deviation from the desired aspect ratio. The ***NetLength*** term is the sum of estimated lengths for each net. The Proximity term allows designers to improve matching by encouraging arbitrary (possibly unconnected) devices to cluster. Devices in such proximity groups are modeled as having a dummy net connecting their respective centers. The ***Proximity*** term is the weighted sum of these dummy net lengths. The ***Merge*** term rewards overlaps that are electrically beneficial. The engine supports a flexible model of what geometry can be merged, which integrates a lot of design-expertise-based rules or constraints.

Seemingly, this cost function covers a lot. However, there are many problems as follows. First, each item has different dimensions. When they are integrated into the same function, the setting of each coefficient will be a tough task. Different applications may favor different weights for each figure of merit, meaning the coefficients must be case dependent. This is quite difficult for a generalized algorithm or engine to achieve. More important, this style with all the (normalized) coefficients doesn't fit the traditional analog design style at all. Even experienced analog designer can't quickly give the exact number for each item before he or she thoroughly understand the detailed situation for each item's normalization. Second, the wiring length estimation is far from precise, mainly because called modules can't provide enough information for their I/O ports. With half-perimeter wire length method, the difference between estimated one and the real may be quite large sometimes. Third, for Area and Aspect Ratio, these two items are often inter-twisted. If the area for analog part is quite small, the aspect ratio is not quite important in most cases. Aspect ratio often really matters when the area is quite large compared with other blocks.

Therefore, their coefficients can't be independent ones set once for all by user or build-in programs.

Such traditional cost function in linear combination style can also be found in [71] (published in 2006), where cost function is similarly a linear function of area, nets and sizes (mainly caring aspect ratio). In [73] and [79], the cost function is also a simple linear combination of several figures of merits: parasitic capacitance, mismatch, aspect ratio and area. In [75], the cost function is somewhat different, as shown in Equation (2).

$$Cost = \alpha(1 + TWD) \cdot TWL + AR \cdot A \qquad (2)$$

Where TWD, TWL, AR and A stand for Total Wiring Difference, Total Wiring Length, Aspect Ratio and Area respectively. And α is a constant used to balance the weight between wiring length and area.

Compared with the former one, this cost function covers the inter-twisted characteristic between AR and A. However, for different applications, there won't be a constant α, optimal for all the cases. And the wiring difference with total wiring length is far from guaranteeing the layout quality. Finally, it's still a simple linear-like combination asking user input as a constant coefficient to deal with unpredictable applications, not fitting analog design style at all.

In sum, traditional cost function demands a clear and quantitative evaluation for each figure of merit. It even needs thorough understanding of the marginal cost/gain between each two figures of merits when user tries to set the optimal weights as coefficients. This is far from what analog designers desire and what their design expertise can tell easily.

4.3.3 Placement and Routing

Fig. 30 [75] shows a typical placement with certain simplified global routings, which is achieved under the guide of the cost function described in equation (2). As shown below, the placement pays much attention to the symmetry for both placement and routing. And actually it introduces further division, listing some primary symmetry nets that must be guaranteed.

However, the problem is: first, the wiring symmetry is not enough to cover all the basic layout rules of thumb so as to guarantee the layout quality; second, the regularity is quite poor; third, though the importance of aspect ratio highly depends on the area, these two figures of merits can't be always quantified simply in a product form to balance with others. Generally speaking, for analog parts, layout quality is always the primary concern.

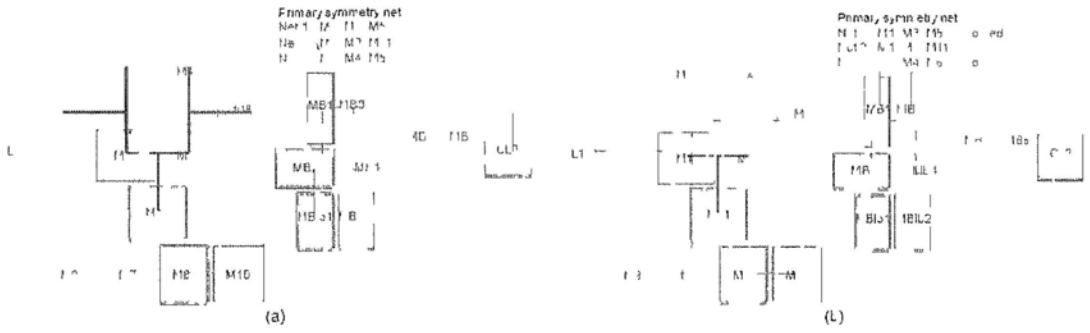


Fig. 30 (a) wiring symmetry for primary nets and unconstrained nets (b) Two primary nets (Net 1 and Net 3) are missing. (Solid lines: the wires among symmetry modules and their connected modules; Dotted lines: unconstrained wires.)

Fig.31 and Fig. 32~34 [80] show the schematic and the corresponding placement of a buffer amplifier. The placement algorithm is based on a hierarchically bounded enumeration of basic building blocks, using B*-trees and being deterministic. The result is claimed to be the Pareto front of placements with respect to different aspect ratios, as shown in Fig.32 to Fig.34.

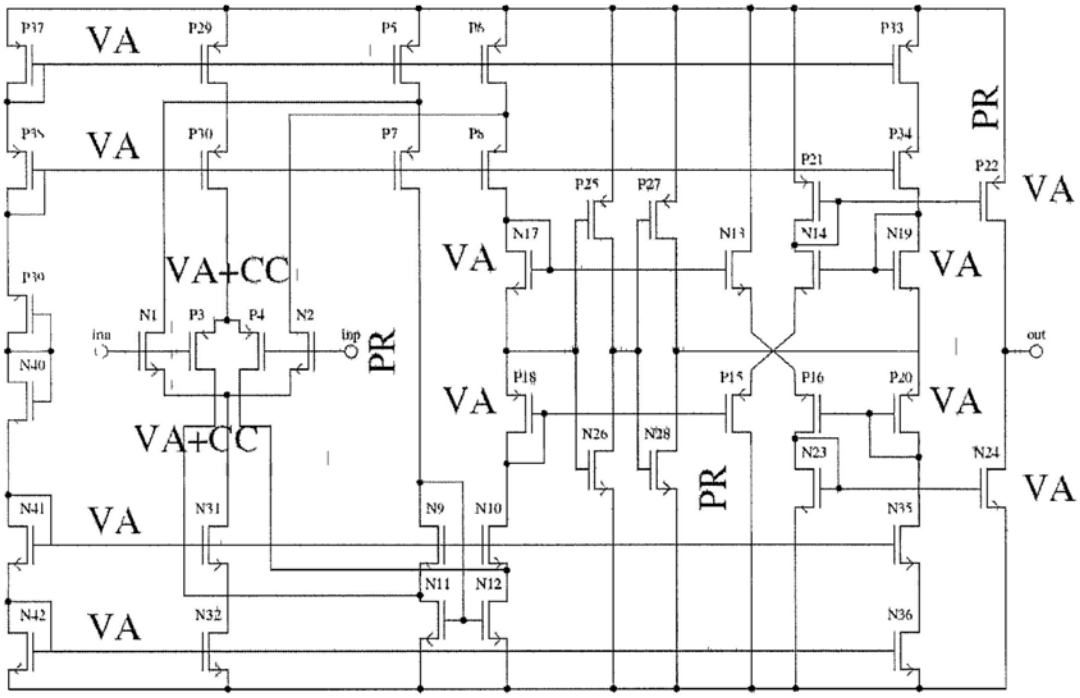


Fig. 31 Schematic of a buffer amplifier

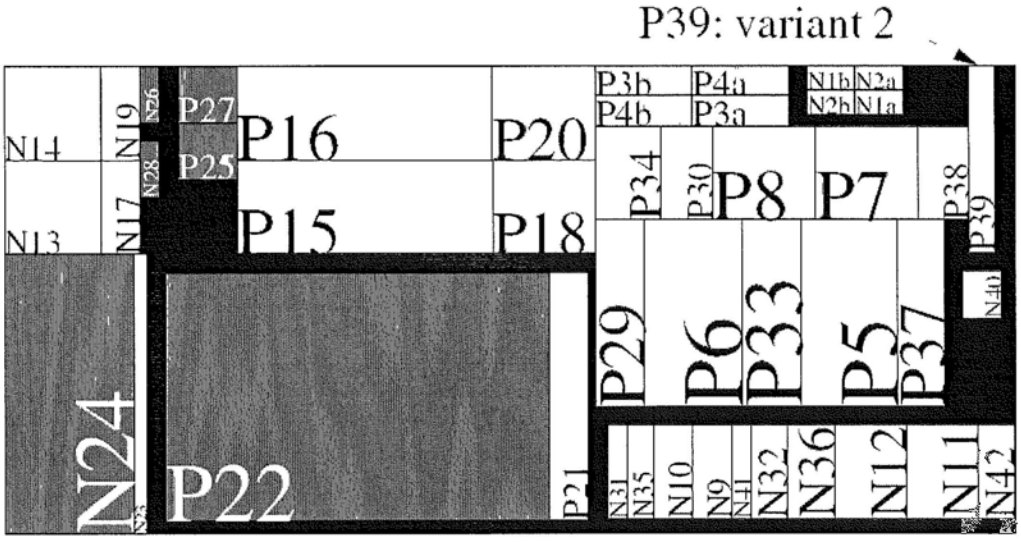


Fig. 32 one placement of an aspect ratio for the circuit of Fig.28

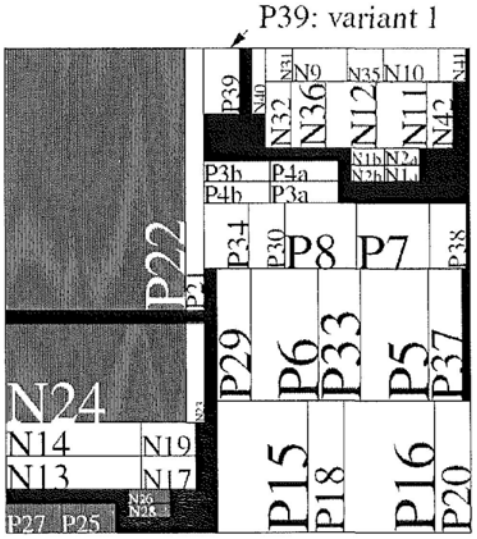


Fig. 33 one placement of another aspect ratio for the circuit of Fig.28

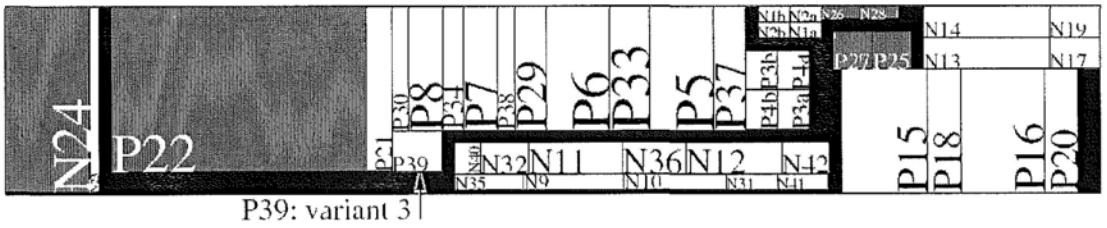


Fig. 34 one placement of another aspect ratio for the circuit of Fig.28

In this method, compactness (area usage) is the first concern while net length comes the second. Obviously layout quality is far from fully guaranteed. The current mirror parts have no common centroid layout settings, and sometimes the routings for some critical parts have to cross over other blocks or go around somewhere. As it pays much attention to compactness (area usage) & aspect ratio, they are usually not the focus for designers. For net length, the second concern, it has to rely on rough calculation (usually calculating total lengths between one shape's center to another), half perimeter of the bounding boxes estimation or something else before all the blocks provide detailed information of each I/O port.

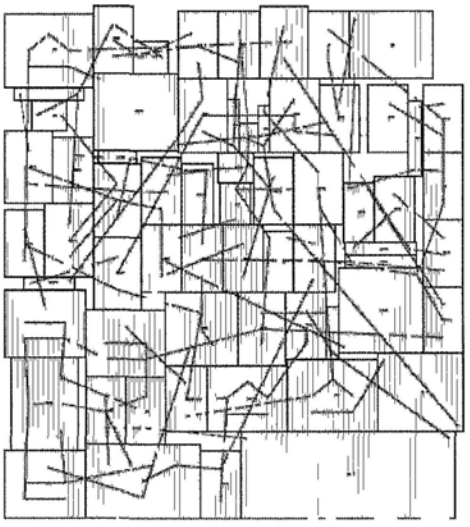


Fig 35 Resultant placements for the min product of chip area and the wire length

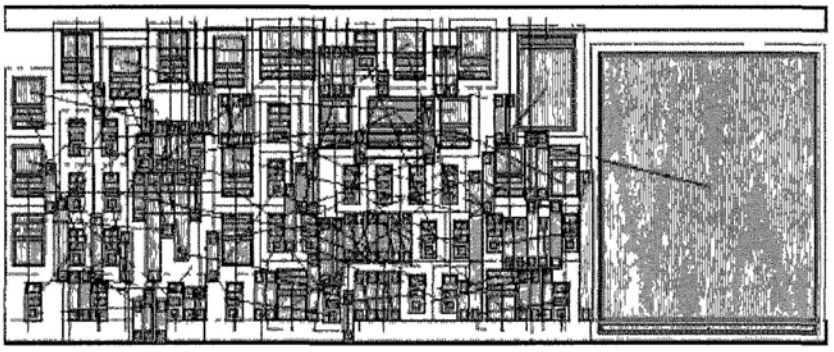


Fig. 36 Automated layout generation of a test analog circuit

As shown in Fig 35~Fig 36[81][82], their cost functions all include total wiring length as a figure of merit, just what equation (1) and the like describe. However, in real applications, neither do analog designers care the TOTAL wiring length nor do these shown methods exactly calculate the total wiring length.

Therefore, the cost function and the PR engine/layout under its guide are still far from mature, especially for generalized algorithms guided by either constraint-driven methods or optimization. Good translation of what analog designers really concern in layout generation is in urgent need

4.4 Other Issues

Besides circuit and layout synthesis, there are some other issues unsolved for ADA, such as Yield Estimation and Optimization, Analog and Mixed-Signal (AMS) Testing and Design for Testability.

For yield estimation and optimization: The manufacturing of integrated circuits suffers from statistical fluctuations inherent to the fabrication process itself [83]. These process-induced deformations (*structural failure, hard performance failure, parametric failure or soft performance failure)* result in deviations of the actual performances of the fabricated ICs from the expected performances.

The challenge is also to incorporate yield and robustness optimization as an integrated part of circuit synthesis, instead of considering it as a separate step to be performed after a first nominal-only circuit sizing. First attempts in this direction have already been presented in [84], [85], but the execution times still have to be reduced further.

For AMS design for testability: In mixed-signal designs, an additional complication is that, although analog circuits constitute only a small fraction of the die area, they are responsible for the largest fraction of the overall test cost. This is not only because they require more expensive mixed-signal test equipment, but also because they require longer test times than the digital circuitry and because there are no structured test waveform generation tools nor structured design for test methodologies. In the future, with the move toward systems on a chip and the paradigm of building systems by mixing and matching IP blocks, the role of analog BIST (Build-In Self Test) schemes will become more and more important in the future, and more research in this direction is to be expected.

So far, as circuit synthesis and layout synthesis are still far from mature in real applications, there is an even longer road to go for ADA development in design for manufacturing and testability.

4.5 Summary

Although the research on ADA has already achieved some tools that already start the commercial applications, there is still a long way to go in wide applications with several key problems shown below:

For circuit synthesis, as feature size continues the downscaling, theoretical optimization for best performance is no longer the sole target, even not the primary concern. Instead, fast reliable design with acceptable performance is what most analog designers need. Being layout-aware and fast is crucial for ADA or development of design assistant as pre-layout design can't have the final say and it will be called for many times usually.

For layout synthesis, the ideal cost function is probably not what the conventional cost functions can describe in such a linear combination form with all the coefficients being set beforehand (either by user's manual input or by system defaults) as it doesn't fit traditional analog design style at all. That means it's quite hard to abstract the implicit analog design ideas and considerations, which covers from topology selection inclination, electrical considerations, thermal issue, geometric estimation and so on. And due to its incompatibility, analog designers can't use these tools naturally, which is not good for the possible commercial use in the future.

Besides cost function, generalized PR engine and algorithm are quite hard to achieve enough optimization for each detailed case. In analog layout design, designers often take

the sequential thinking and always identify the critical parts first. In their minds, optimizing the critical parts first and keeping other parts acceptable is the big picture while giving detailed optimization for every part can easily trigger lots of iteration loops, complicating the engine for iteration convergence while costing the advantage of speed with little guarantee on quality (making ADA somewhat unattractive). As a matter of fact, constraints-driven methods with generalized algorithms are still not as good as template-driven ones (with build-in knowledge-driven engines) in layout quality [86].

For statistical AMS design to improve the yield or design for testability issue, though there are already some researches, they don't make too much sense right now as the core parts of ADA (circuit & layout synthesis) are still in doubt for their quality and efficiency.

CHAPTER 5. Proposed ADA Design

As the available analog CAD tools still have many problems making ADA still far from mature, we propose a novel approach to develop analog CAD tools used as design assistant. To make it attractive and feasible (even in the possible commercial use in the future), we propose a novel ADA tool focusing on the common applications at first.

So far, our platform CIRCLES (CIRCuit and Layout Expert System) supports two topologies for op amp: 2-stage op amp with Miller Compensation and single-stage current mirror op amp. Besides the op amp case, regular analog circuits such as voltage multiplier (charge pump) can also be supported (As it's quite easy for our platform to realize the function of reconfigurable design shown in Chapter 3, VM cases won't be repeated here as the layout generation have the similar idea). In the future, based on our op-amp platform and powerful module generators (covering R, C, MOSs, and many circuit structures…), relatively complex circuits such as SCF and ADC can also be realized (see Section 6.3).

Combining circuit synthesis and layout synthesis, our platform accomplishes the fast "SPEC in, GDS out" flow, guaranteeing the layout quality while keeping the manual option. Before introducing our proposed tool, problems and ideas on ADA are summarized as follows.

5.1 Problems and Ideas

As described in Chapter 3, there are already some analog CAD tools available as the progress achieved in Analog Design Automation (ADA). However, so far, the CAD tools

for analog circuits are still far from mature compared with their digital counterparts. Compared with the analog layout made by manual work, the available CAD tools still have the following problems in common applications:

1) Circuit Synthesis:

1.1) initial guessing: irrelevant, often causing lengthy iterations

1.2) optimizing: global optimal is guaranteed only in principle or at some cases (when conditions can be written in certain forms).

1.3) few consideration for reliability especially for small size effect

1.4) convergence issue: need manual input or heuristic steps

2) Module generators of the layout synthesis:

2.1) not covering all the basic layout rules of thumb, e.g. uniformity of current flow direction for current mirror, symmetric dummy cells for critical layout vulnerable to disturbance from outside;

2.2) not flexible enough to re-adjust the generated shapes for good regularity;

2.3) not friendly enough to global routing, e.g. lack of I/O ports relative location report, few local routings for power/ground to ease the possible global interconnection and so on.

2.4) the module library is still far from enough and it should be built in a style that has good extensibility.

3) Cost function and PR of the layout synthesis:

3.1) cost function is a simple linear combination of figures of merits with weighted coefficients being set by either user or system (default). This kind of configuration

doesn't fit the design expertise and manual design style at all, thus being unattractive as a design assistant for analog designers.

3.2) placement algorithms of available tools pay too much attention to compactness and the possible aspect ratio requirements from users with no sufficient guarantee on layout quality. As a matter of fact, for many common applications, compactness/area is not the top concern of user. As area is not the top concern or even quite small, aspect ratio plays a much more insignificant role than generally imagined.

3.3) routing algorithms usually focus on the total net length, not exactly what tops in designers' concerns when they do it manually. Moreover, due to the lack of global routing information provided by module generator, routing algorithms often have no idea of where global routings exactly begin or end. Instead, they use half-perimeter estimation or other methods (e.g. summing the total distance among all block centers), which may deviate greatly from the reality.

From all these above, we can see that:

For circuit synthesis: initial guessing needs to be knowledge-driven, minimizing the possible iteration. Facing small size effects and the real needs, the goal should be near optimal or good enough performance with guaranteed reliability and enough margins. Heuristic inputs should be avoided or replaced by input of usage description while manual control can be still reserved as an option.

For module generator: the information of I/O ports will be covered with all the basic rules of thumb in analog layout need to be realized. In essence, it will be much layout quality oriented rather than compactness oriented. Local routings for power & ground will be presented in a regular way to facilitate the coming global interconnection. For flexible

readjustment, module code is favored rather than direct layout until every parameter is finally ascertained. As module generators are inherently application-specific and knowledge-based, any available analog layout that proves to be classical can be and should be abstracted into module generator library to provide universal & silicon-proven solutions for analog designs.

For cost function or evaluation function, to our best knowledge, major figures of merits (such as area, routing length) should be divided into critical parts and non-critical ones. PR is primarily determined by critical parts, but not in a linear combination manner. (Since each figure of merits has different units, normalization is not easy and in fact highly case dependent). Essentially, coefficient of critical/non-critical part of one figure of merit is no longer a constant but a function dependent on usage description, other figure of merit and design expertise. The cost function form should reflect some serial thinking pattern that analog designers consciously do. That is, optimizing the critical parts first while keeping the rest acceptable (using design expertise & usage description to quantify the "acceptable").

For PR issue: placement for critical parts are layout quality oriented (mainly critical-routings-driven); for the rest part (non-critical area), the policy is compactness & aspect ratio (if area is large) oriented. This kind of serial processing with different policies for different parts is exactly what analog designers manually do for their layout and what they expect ADA to do.

In short, to simplify and accelerate circuit synthesis, initial point guessing needs to be knowledge-driven. Manual intervention or direct input is almost compulsory to deal with the possible non-convergent issues. Module generator program needs to cover all the basic

layout rules of thumb, optimizing for layout quality as the top priority. To maintain great flexibility, modules are better to remain in code form rather than real layout until everything is ascertained. Cost function utilizes the serial thread, being quality-driven for critical parts while doing compactness/regularity-driven for the rest. Of course, dividing algorithms and many other details are inherently knowledge-based.

5.2 Circuit Synthesis

Now we start to do the proposed ADA tool for the first step: circuit synthesis. Here it mainly consists of two parts: topology selection and circuit sizing. As the layout hasn't been generated so far, it must be a pre-layout design with pre-layout simulation to decide whether or not the performance is good enough to meet the SPECs.

As the prior work on analog circuit synthesis pursues a global optimization for optimal performance of the analog circuits, optimization methods that can guarantee the exact point with optimal performance either have harsh requirements on the type of constraints and conditions, or only achieve the goal in theory.

Now with the ever-decreasing feature size in deep submicron era, more and more higher-order effects come with all kinds of nonidealities, causing huge variation and uncertainty, making the traditional only-optimal-performance-oriented methods no longer as suitable as they were. Instead, reliable design with acceptable performance (of course, meeting the SPECs at the least) is to be the trend in our view.

To quickly get the result and still remain universal for many common uses, we choose knowledge-based methods with simulation-based inner loops for fine tuning in iteration.

Here we describe how to design an op amp based on standard UMC 130nm technology. For topology selection, so far we let user directly select the corresponding topologies from

the menu as shown in Fig.37, where 2-stage op amp with Miller compensation is supported as well as single-stage current mirror op amp. For these two topologies, the most obvious difference between them is the gain. So we use the following arbitration principle based on our knowledge and general requirements on gain: two-stage op amp is designed for DC gain ≥40dB while single-stage op amp is designed for DC gain <40dB.

In the future, according to different SPEC ranges and usage description inputs, a more sophisticated judging program will be used to deal with topology selection (for beginners) while direct selection will still be kept (for advanced users).

For general-purpose op amp, the key SPECs are: DC gain (A_V), Unity Gain Frequency (UGF), Slew Rate (SR), Loading Capacitance (C_L), Common Mode Rejection Ratio (CMRR) and Power (P). For other SPECs: Input Common Mode Range (ICMR) is relatively fixed under a given process and a given rail-to-rail setting since it mainly relates to threshold voltage (V_T, nearly fixed by given process) and overdrive voltage (V*, usually tunable in a very small range, 0~200mV at the most). Input Referred Noise (IRN) mainly depends on the transistors' sizes while Area is usually not the primary concern for general-purpose op amp design, especially for mixed-signal applications as analog parts are quite small compared with the digital ones. Output Swing (OSW) highly depends on topology, user-input tolerance or error (ERR) and feedback factor for op amp's application in a system with (negative) feedback. And for PSRR, it also highly depends on topology and DC gain.

Therefore, the key SPECs together with the topology can to a great extent determine the circuit design and the values or typical ranges for other SPECs. That's why the initial point setting is mainly based on the analysis of the key SPECs and topology as follows.

5.2.1 Initial Point Setting

Suppose we have the following input SPECs shown in Fig.37. According to the input SPEC for DC gain, we choose 2-stage op amp with Miller Compensation and PMOS-differential-input. Fig.38 shows the general schematic of a 2-stage op amp with Miller compensation. There are 26 design parameters altogether to be determined:

- The finger widths (w), lengths (l) and No. of fingers (nf) for M_0 to M_7
- The bias current I_{BIAS}
- The Miller compensation capacitor C_C

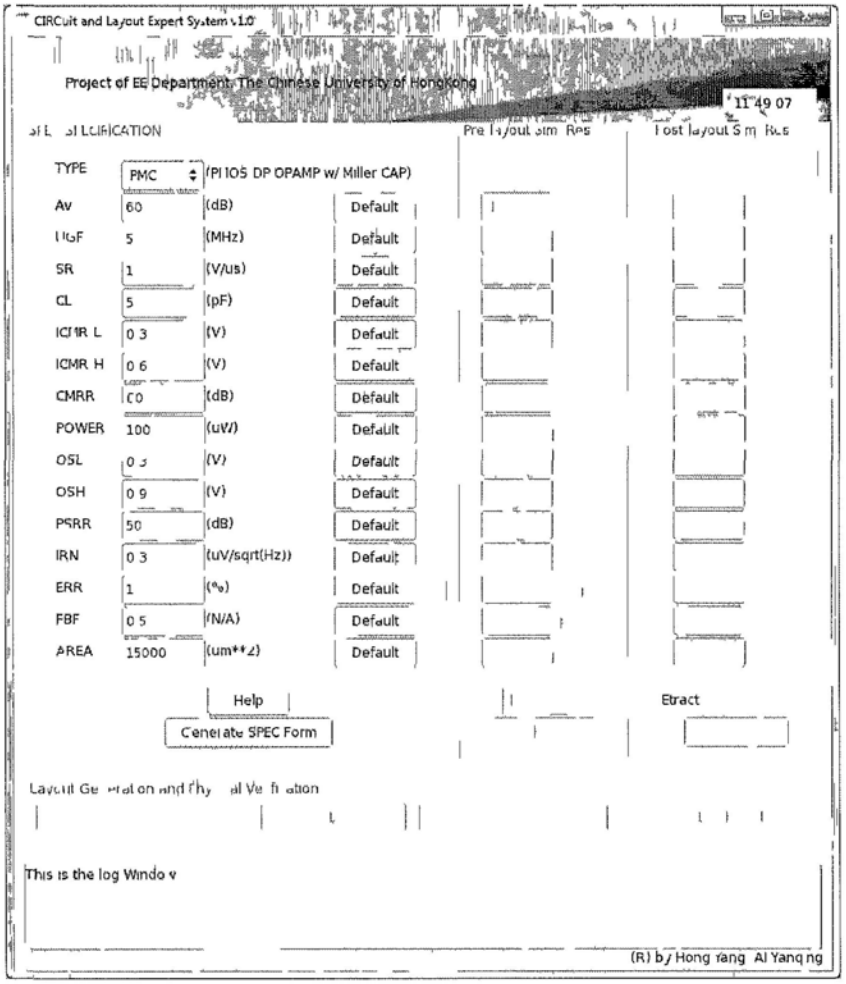


Fig. 37 User-input SPECs for a general-purpose op amp

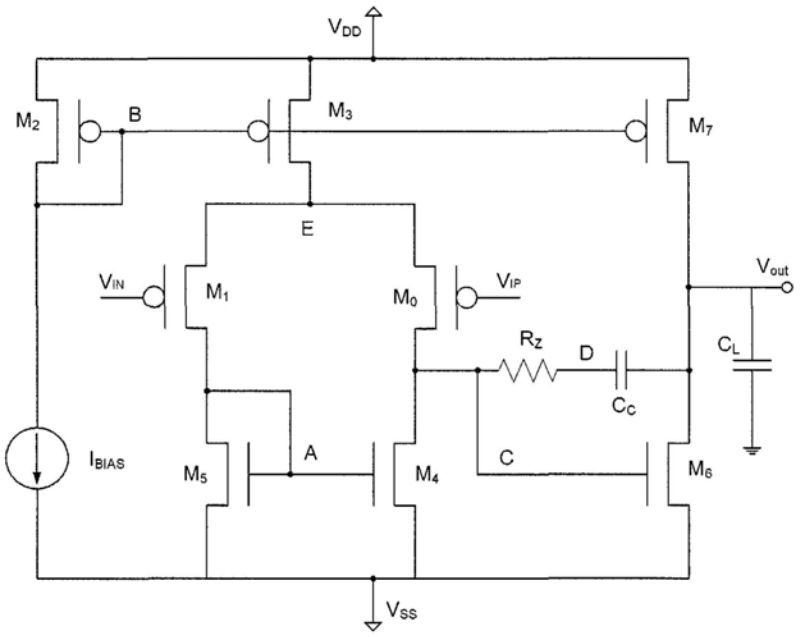


Fig. 38 Schematic of 2-stage op amp with Miller cap and PMOS input

The nulling resistor is not included on purpose. Its value, R_Z, is highly unpredictable as it is normally realized by wire resistance. As long as stability can be guaranteed even when R_Z=0, with a nonzero R_Z (usually quite small), stability can only be better. The whole idea is helpful in getting a reliable design and improving the yield.

The problem of finding the right values for the design parameters can be greatly simplified by considering analog design knowledge and experience together with layout rules of thumb. From Fig.38 or the equivalent netlist input, M_2, M_3, M_7 form one current mirror (CM) while M_4 and M_5 form another. Therefore, l_2=l_3=l_7 while l_4=l_5. As for the differential pair (DP) M_0~M_1, it suggests: w_0=w_1, l_0=l_1 and nf_0=nf_1. Considering the balance, w_4=w_5 (finger width) and nf_4=nf_5. Due to layout rules of thumb such as unit matching, for CM MOSs, w_2=w_3=w_7 while nf_2, nf_3, nf_7 may differ. And to realize common centroid layout, nf_0 and nf_1 must be even numbers as they are equal. It is the same for nf_4 and nf_5. As for nf_2, nf_3, nf_7, there can only be one odd number at the most.

Up to now, we have 10 equations as shown above. However, compared with those 26 variables (w_i, l_i, nf_i for M_0~M_7, I_{BIAS} and C_C when R_Z is zero and C_L relies on user), these equations are still far from enough. But considering the process information and reliable design style for common use, design expertise can help us to get a reliable and proper initial point to start.

The thread starts from SR and C_C, where we can get the relationship between I_{BIAS} and the ratio of W_3 to W_2 as shown in equation (3). Herein for user's convenience, input SPEC of SR refers to the average SR of output from 30% to 70% (rail to rail) when input pulse changes from 25% to 75% (rail-to- rail). So, a simulation/experience-based coefficient α is introduced between theoretical SR and input SR. For PMOS-input 2-stage op amp case, α is initially set to be 5.

$$\alpha \cdot SR = \frac{W_3}{W_2} \cdot \frac{I_{BIAS}}{C_C} \qquad (3)$$

From the stability requirement, Phase Margin (PM) must be larger than a certain amount, usually 60°. So according to design expertise, the second pole (p_2) must be at least 2.2 times higher than UGF, which gives the inequality (4).

$$C_C \geq 0.22 \cdot C_L \qquad (4)$$

As soon as C_C is set, the transconductance of the MOSs in differential-pair, g_{m1}, can be obtained as equation (5) shows:

$$UGF \approx \beta \cdot \frac{g_{m1}}{C_C} \qquad (5)$$

Where β is a coefficient to compensate the decreasing effect on the output resistance due to channel length modulation. Usually β varies from 1.05 to 1.15, which depends on the biasing and process. Initial setting is set relatively large on purpose, guaranteeing a larger transconductance for sufficient dc gain.

According to user-input C_L, once we set C_C, we know the drain current of M_3, and also the drain current of M_0~M_1 and M_4~M_5. Then W/L and V* can be ascertained for M_0~M_1. Traditionally, the next step is to calculate V* and W/L of M_4~M_5 based on ICMR requirement. However, as feature size goes down, channel length modulation gets more and more obvious. M_1 and M_5 no longer behave as classic formulas. Even adding λ by pre-testing can't help since λ changes drastically based on different biasing. As a result the V* of M_5 will often be a wide & unreliable range.

Facing the small size effects mentioned above, we focus on M_3 instead of M_5 as M_3 plays an important role in achieving enough CMRR (usually no less than 60dB). Under 130nm technology, our simulation shows that CMRR only meets the requirement when M_3 goes into deep saturation ($V_{DS3} \gg V_{M3}^*$). As the room for V* is quite limited, esp. with a low supply voltage (1.2V for standard 130nm process), we set default values for initial V* to guarantee CMRR. Herein, for 2-stage op amp case, V* is set to be ~60mV as the initial value for biasing current mirror MOSs. Accordingly, V_{DS3} must be maintained at a value no less than 100mV, even 150mV for an acceptable CMRR. Actually, this is also a requirement from precise current mirroring. As channel lengths shorten, only in deep saturation region can the match between I_{M3}/I_{M2} and nf_3/nf_2 be acceptably precise. In this way, l_3 is ascertained, which is mainly a knowledge and technology driven process. Once we know I_{M3}, V_{M3}^* and l_3, the product of $W_3 \cdot nf_3$ can be set.

For reasonable phase margin, the first zero (z_1) must be placed ten times higher than the bandwidth or UGF, which requires

$$g_{m6} > 10g_{m0} \qquad (6)$$

For our 130nm case, initial setting is $g_{m6} = 12g_{m0}$. But since M_6 and M_0 are different in type, a coefficient (4~4.5) is given to compensate the difference between NMOS and PMOS in μC_{ox} (process dependent) when we calculate W/L for M_6 based on M_0. As a result, large g_{m6} and already-known $(W/L)_{M6}$ lead to a relatively small V* (<0.1V) and ascertain the output stage current I_{M6} or I_{M7}. As 1st stage gain degrades due to many higher order effects such as channel length modulation (λ) and so on, the 2nd stage must provide enough gain to meet the SPEC. Therefore, this setting for M_6 (large g_{m6} under a proper I_{M6} for power concern leads to a small V*) is quite reasonable.

Knowing dc current I_{M7}, then, according to V* of M_3, $W_7 \cdot nf_7$ can be determined while l_2 and l_7 are already set by l_3. After that, W_2, nf_2, W_3, nf_3, W_7, nf_7, and I_{BIAS} are finally set in such a manner that there are 1 odd number among nf_2, nf_3, and nf_7 at the most (for common centroid layout).

For M_4 and M_5, they can no longer be set based on ICMR information due to the poor-saturation issue, but based on design experience. In our case, V* for current mirror load is initially set quite similar to the differential pair, leaving enough room for both sides to remain in deep saturation. As for the channel lengths of differential pair and current mirror load, they are the same and set mainly by process information and experience, usually at least 5 times the minimum size (since area is not the primary concern and large size is good for low noise).

Fig. 39 summarizes all the deduction threads mentioned above. The red ones are the user-input SPECs while the purple ones are the outcomes for parameters to be used in simulations. The green ones are process or design expertise determined parameters, and the blue ones stand for the principle ("ccl" stands for "common centroid layout"). For

NMOS differential-pair case, the initialization for the 2-stage op amp with Miller Compensation is quite similar, which we won't mention too much.

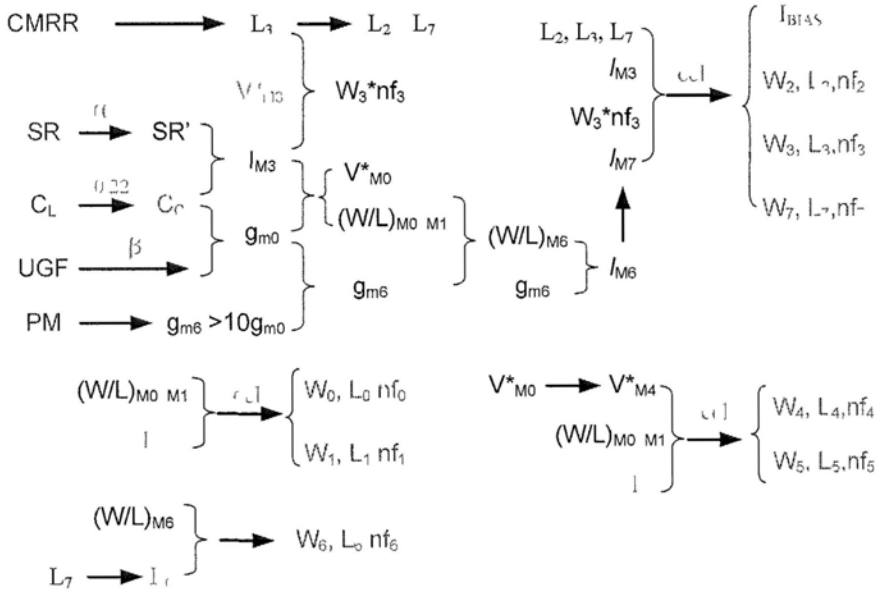


Fig. 39 Deduction flow for initialization of parameters (2-stage case)

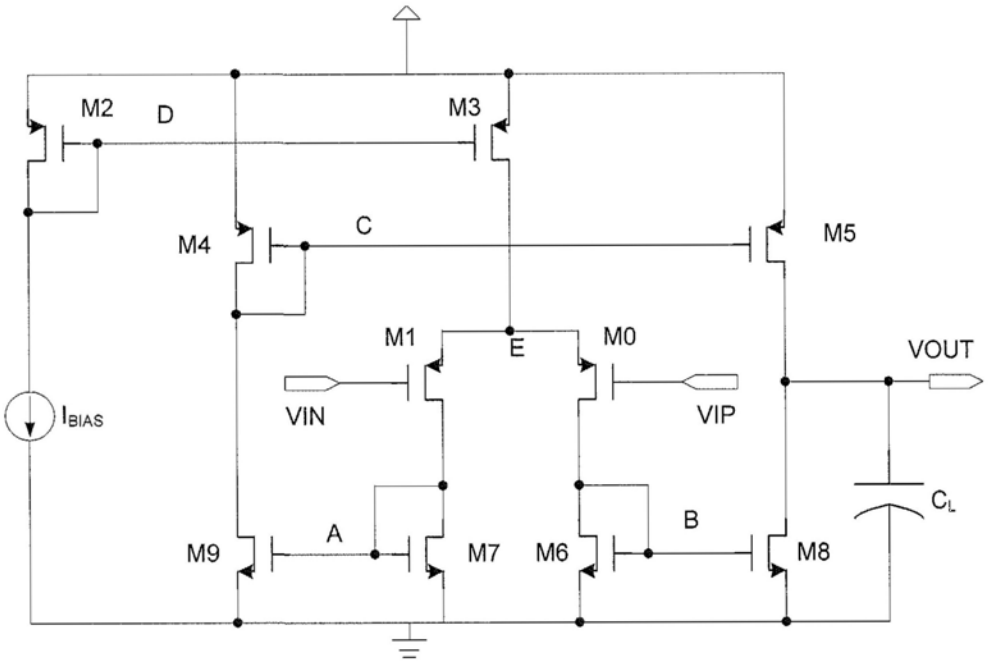


Fig. 40 Schematic of single-stage current mirror op amp

Besides the 2-stage op amp topology mentioned above, our proposed platform also supports single-stage current mirror op amp design as shown in Fig.40, which has a larger output swing and is suitable for general applications that don't ask for a very high gain.

The design thread starts from SR and CL, which determines the output stage dc current (the similar transition from average SR to maximum SR that used in theoretical calculation is almost the same story as the 2-stage op amp case with a compensating coefficient α). Then through design experience and process, the initial values of V* can be chosen. Based on these data and process, W/L for M_5 and M_8 can be determined. After that, the sizes for M4, M6, M7 and M9 can be written in a function of n1, n2 and n3, where

$$n_1 = \frac{W_8/L_8}{W_6/L_6} \qquad n_2 = \frac{W_9/L_9}{W_7/L_7} \qquad n_3 = \frac{W_5/L_5}{W_4/L_4} \tag{7}$$

Usually due to the symmetry of differential pair, M_6 and M_7 are the same. And for the whole structure's symmetry to minimize the systematic offset, M_4 & M_5, M_8 & M_9 are correspondingly matched, so $n_1 = n_2 \cdot n_3$.

According to power requirement, n_1 & n_3 can be determined based on known I_{M8} and I_{TOT}. Then the current for each transistor (except M_2) is already known. Based on biasing current of M_3, UGF, and C_L, also considering the process information, the W/L of M_0~M_1 can be obtained as UGF is highly dependent on the transconductance of the differential pair, which is shown below.

$$UGF = \beta \cdot \frac{n_1}{C_L} \cdot \sqrt{I_{M3} \cdot \mu_{diff} C_{ox} \cdot \left(\frac{W}{L}\right)_{diff}} \tag{8}$$

Where β is a coefficient to compensate the degradation effects due to small size and ignorance of parasitic capacitance of output stage. Up to now, W/L has been clear for all the transistors except M_2.

To determine the exact value for their widths and lengths, we need to start from dc gain. As the gain depends on the output resistance as show in (9), the channel length for M_5 and M_8 can be found according to process data collected by specific testing program.

$$A_V \propto \frac{1}{\lambda_N + \lambda_P} \qquad (9)$$

Once L_5 and L_8 are known, L_4~L_9 can be determined. Unlike L_2 and L_3 are selected by CMRR requirement, the channel length for differential pair is selected mainly by experience. For low noise, they are often relatively large or similar with other MOSs if no other special SPEC is proposed. As for M_2, as its V* and length are determined by M_3, its primary concern is to simplifying the nf_2 to nf_3 ratio for common centroid layout (power is not the major concern as W_2 is usually much smaller than W_3).

Fig.41 shows the total thread for single-stage current mirror op amp design. The red ones are the user-input SPECs while the purple ones are the outcomes for parameters to be used in simulations. The green ones are process or design expertise determined parameters or deduction procedure, and the blue ones stand for the principle ("ccl" stands for "common centroid layout"). As the circuit is single-stage with only 1 node being high impedance and having relatively large capacitance (output node), phase margin is not a major issue, especially for low frequency applications as common use. As for the NMOS type, it's almost the same.

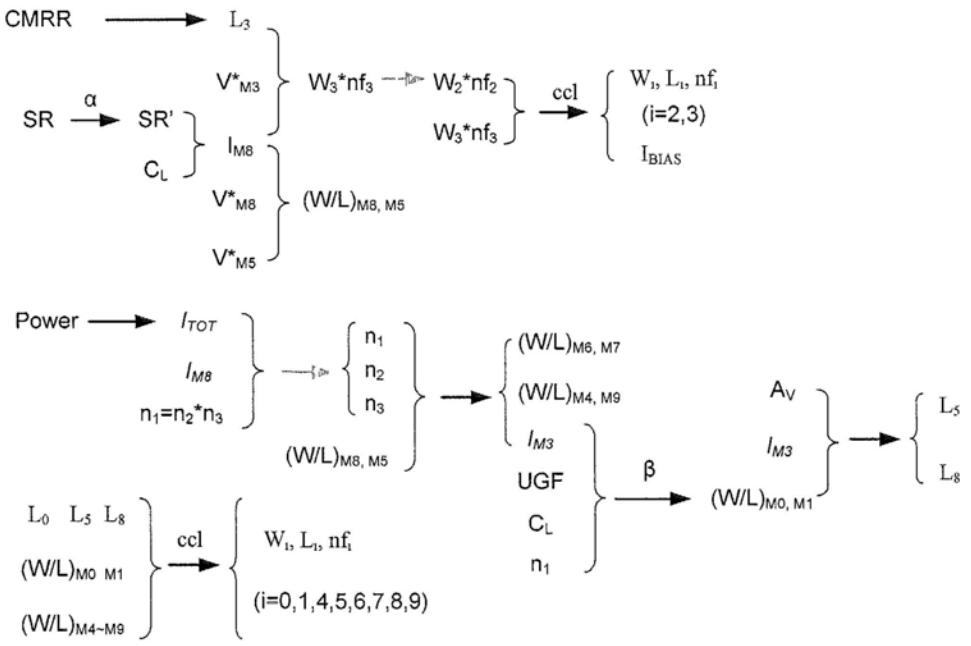


Fig. 41 Deduction flow for initialization of parameters (1-stage case)

Obviously nearly all the key SPECs are covered in initialization except the power of 2-stage op amp case. As initialization has already taken many SPECs, process and design expertise into account, the coming possible iterations can be greatly reduced.

5.2.2 Arbitration and Iteration

So far, arbitration focuses on the key SPECs listed above, namely Av, UGF, PM, SR, CMRR and Power (CL is already set by user). If any of these figures of merit doesn't meet the corresponding SPEC, the program will call the corresponding iteration programs to do specific change for circuit sizing as shown in Fig.42.

```
If (Simulated_SR<SR_SPEC) {
    & SR_improve (α1*SR_SPEC/Simulated_SR); $ite++;}
If (Simulated_CMRR<CMRR_SPEC) {
    & CMRR_improve (α2, @wi, @li); $ite++;}
If (Simulated_Av<Av_SPEC) {
    & Av_improve (α3*Av_SPEC/Simulated_Av); $ite++}
If (Simulated_UGF<UGF_SPEC) {
    & UGF_improve (α4*UGF_SPEC/Simulated_UGF, CC, CL, IBIAS, @wi, @nfi); $ite++}
If (Simulated_PM<PM_SPEC) {
    & PM_improve (α5*freq_p60/Simulated_UGF, CC, CL, IBIAS, @wi, @nfi); $ite++}
& power_check;
If ($ite<= iteration_limit) {
    & update_netlist(CC, CL, IBIAS, @wi, @li, @nfi);
    & re-run_simulation;
}
```

Fig. 42 SPEC-specific iteration program (Perl)

Where α_1~α_5 are the experience-based coefficient used to give a margin over each detailed SPEC for reliability since the data are just from pre-layout simulations. By our default settings, these coefficients are given a value of 1.05~1.1, leaving some margin for post-layout simulation that follows. The parameter "freq_p60" refers to the frequency at which phase margin is 60° (obtained by HSPICE simulation). "@w_i, @l_i, @nf_i" covers all the sizing information for each transistor.

Each subroutine for update & iteration focuses on improving the corresponding figure of merit with the least affect on other unmentioned SPECs. However, some subroutines will unavoidably affect other SPECs, such as the SR improvement program will lead to higher output stage current, larger sizes, which will increase the UGF. But usually they are either independent (almost no effect on other figures of merit) or helpful (contributing to the improvement of other figure of merits).

However, as a matter of fact, any iteration process will encounter some non-convergent cases. As our goal is to achieve a fast & reliable design assistant, we can't sacrifice speed

too much. So we have a setting on iteration limit (12 as default), which of course can be changed by user. Moreover, direct manual intervention is provided as shown below.

5.2.3 Manual Intervention

As any iteration process can't avoid the possible non-convergent issues, for fast and easy use, our platform provides manual intervention. User can manually change the generated netlist file by clicking the "Edit Netlist" button shown in Fig. 37 or directly generate one netlist file to replace the auto-generated one. After clicking that button, our platform will open the generated netlist file as shown in Fig.43. It's a common CDL netlist, which analog designers are quite familiar with.

Note that once user chooses the manual intervention, the system will no longer use any iteration that may change user's manual-input sizes, but just report the corresponding simulation results. It's designed for users to check their ideas. If they want further iteration based on their manual initialization, so far they have to manually input the command illustrated in the application note.

```
.SUBCKT  pmc_op  vdd  vss  vip  vin  VOUT
M0       C    vip      E     E    P_12_SPL130E   W=13.98u    L=1u     M=2
M1       A    vin      E     E    P_12_SPL130E   W=13.98u    L=1u     M=2
M2       B      B    vdd   vdd   P_12_SPL130E   W=3.26u     L=0.5u   M=1
M3       E      B    vdd   vdd   P_12_SPL130E   W=3.26u     L=0.5u   M=10
M4       C      A    vss   vss   N_12_SPL130E   W=3.5u      L=1u     M=2
M5       A      A    vss   vss   N_12_SPL130E   W=3.5u      L=1u     M=2
M6    VOUT      C    vss   vss   N_12_SPL130E   W=4.08u     L=1u     M=16
M7    VOUT      B    vdd   vdd   P_12_SPL130E   W=3.26u     L=0.5u   M=40
C0    VOUT      C   3.13p
I0       B    vss  ibias
C1    VOUT      0  cload
.ENDS
```

Fig. 43 Opened netlist ready for manual change

For user's better understanding, files containing information about typical values, format of outputs and application notes will be automatically opened once user clicks the "Edit Netlist" button. Fig.44 shows the typical ranges for the two supported topologies in both N-type and P-type configurations (4 combinations totally).

pmc	nmc	pcm	ncm		
<=70	<=70	<=45	<=50	Av	V/V
<=22	<=7.5	<=3.5	<=3.5	UGF	MHz
<=9.8	<=5.0	<=1.0	<=1.0	SR	V/us
1~5	1~5	1~5	1~5	CL	pF
~0.3	~0.5	~0.3	~0.5	ICMR-L	V
~0.6	~1.0	~0.6	~1.0	ICMR-H	V
<=60	<=60	<=60	<=60	CMRR	dB
>=40	>=50	>=5	>=5	POWER	uW
~0.3	~0.3	~0.2	~0.3	OSL	V
~0.9	~0.9	~0.9	~1.0	OSH	V
<=73	<=71	<=46	<=53	PSRR	dB
>=0.3	>=0.3	>=0.3	>=0.3	IRN	uV/sqrt(Hz)
>=1	>=1	>=5	>=5	ERR	%
>=0.5	>=0.5	>=0.5	>=0.5	FBF	dimensionless
>=15000	>=20000	>=6000	>=3000	AREA	um**2

NOTE: due to tradeoff, extreme values can't be realized simultaneously

Fig. 44 Typical range of op amps in 4 combinations

Note that ICMR is calculated based on simulated data about V_T and $V_{DS,SAT}$ while PSRR is the minimum value of PSRR+ and PSRR- at either 1kHz or 10KHz (minimum value among the 4 combinations) .

After circuit synthesis, output results about sizing and simulation results will be shown to user via GUI interface as shown in Fig.45.

After that, the next step is to generate layout by clicking the "Generate Layout and Run PV" button. But before that, we have to set up a critical basis to facilitate the layout PR process. That's why Section 5.3 comes next describing the details on Module Generators (MGs).

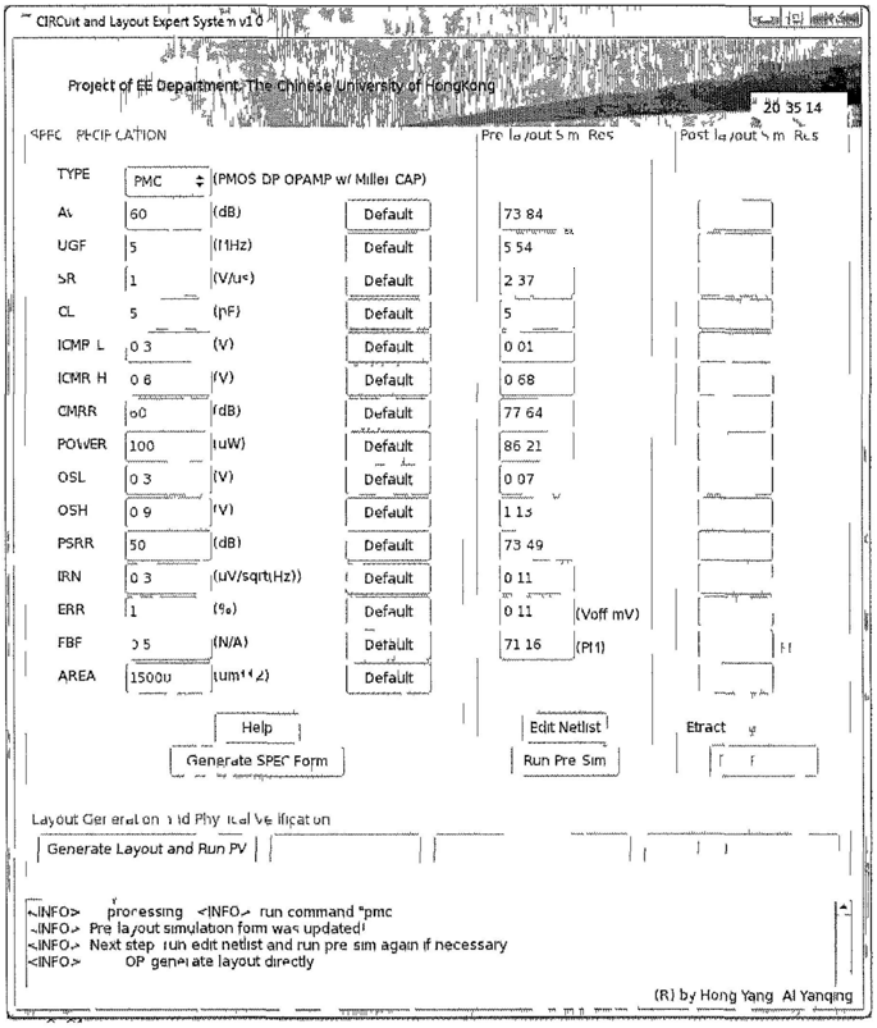
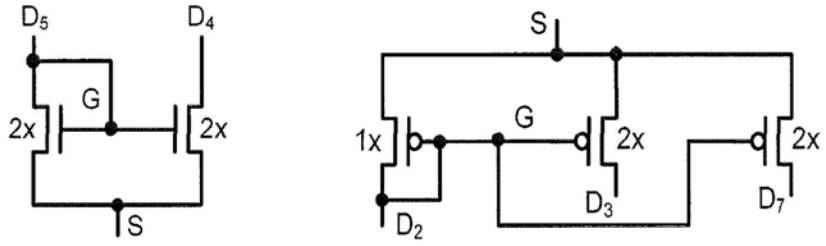


Fig. 45 Pre-layout simulation results shown in GUI interface

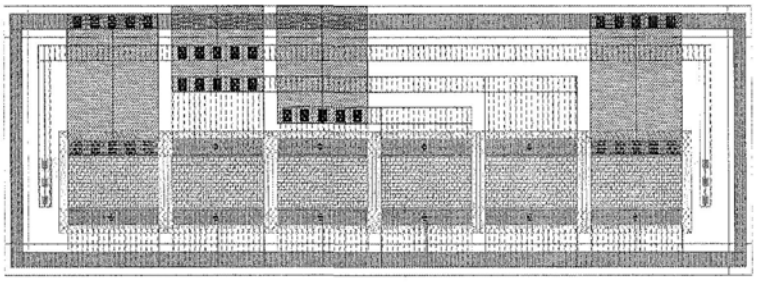
5.3 Module Generator

To simplify the layout PR issue and make the automatic layout generation more flexible, extensive use of module layouts is needed. For classical circuit structures or basic components that are frequently-used in common applications, i.e. current mirror, differential pair, capacitor array, layer-to-layer (L2L) connection and so on, our platform (CIRCLES) provides the corresponding Module Generators (MGs), which is both highly

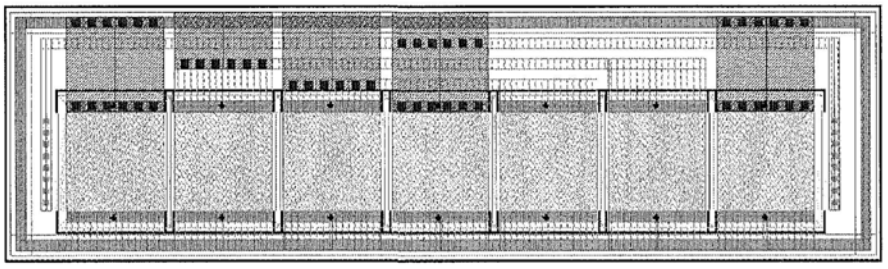
parameterized to generate flexible module layouts and layout-quality-oriented as the PR algorithms of modules are guided by build-in design expertise with many considerations accommodating appropriate analog layout rules of thumb for quality and reliability concerns.



(a) (b)



(c)

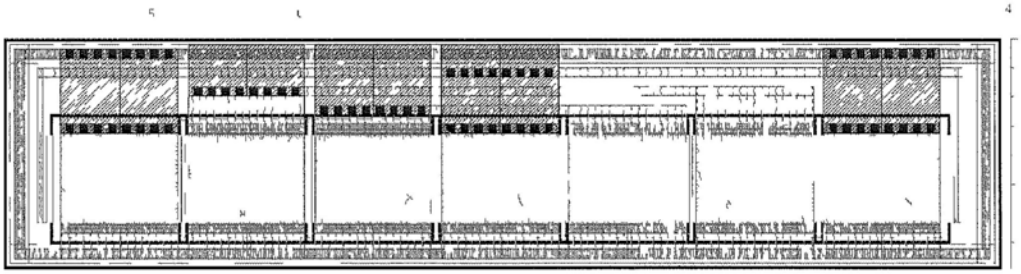


(d)

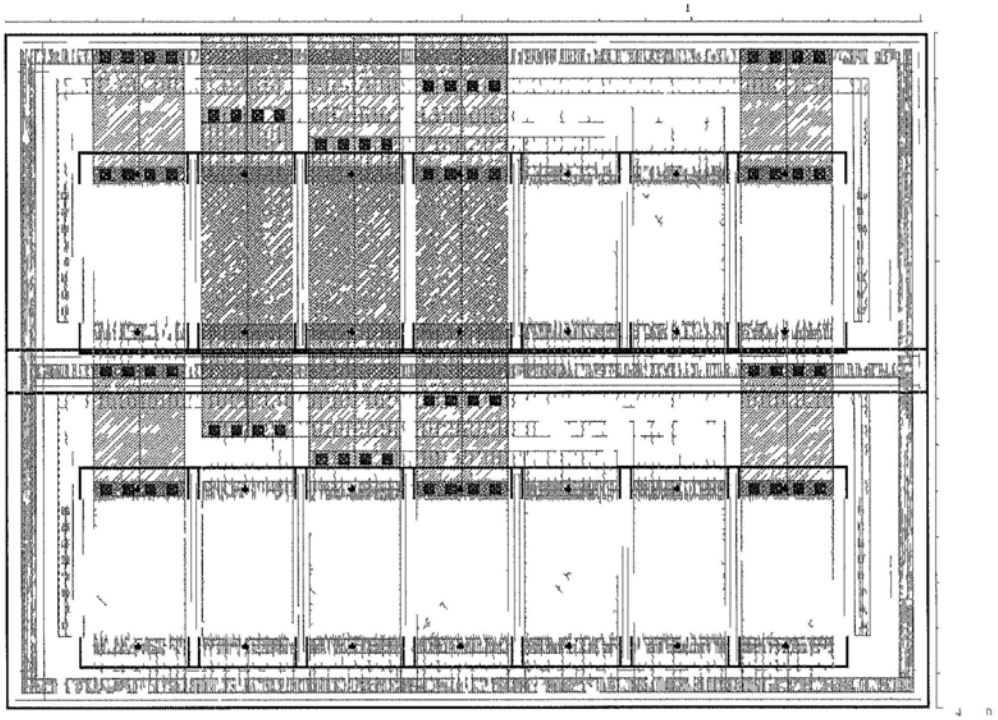
Fig. 46 (a) Schematic of current mirror (NMOS) (b) Schematic of current mirror (PMOS) (c) Generated layout for current mirror of Fig.46 (a) (d) Generated layout for current mirror of Fig.46 (b)

Fig. 46(a) and (b) show the schematics of NMOS and PMOS current mirrors that are often called in general-purpose 2-stage op amp design. Unlike the inter-digitized structure as shown in Fig. 26(b), the placements in Fig.46(c) and (d) not only guarantee the exact common centroid layout and unit matching (every MOS finger has the same width, including the dummy ones) but also achieve a uniform current flow direction, which is quite important and helpful for current mirror matching. Of course, to get the common centroid layout, there is at most one transistor that can have an odd finger number in the current mirror structure. For better matching, two dummy transistors are added on both sides to make the surroundings almost the same while the whole nearby PWR/GND rings facilitate the coming global connection, minimize the bulk resistance, and shield the inner circuits from any outside unpredictable disturbance. Moreover, the shape of each generated current mirror structure is regulated to be a standard rectangle, which greatly improves regularity and facilitates floorplanning that follows.

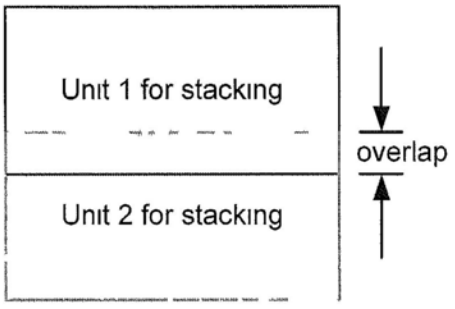
As common centroid layout puts the constraint on finger number ratio for each current mirror layout, stacked structure becomes a good choice to deal with the arbitrary ratio that user may order, increasing the flexibility. At the same time, good regularity for modules makes the stacking quite easy to achieve and still guarantees a good regularity for the whole stacked structure as shown in Fig.47. This is very helpful in optimizing the aspect ratio for total layout when the area is no longer negligible compared with other blocks.



(a) Original layout: horizontally too long & vertically too short



(b) Stacked layout with a better aspect ratio



(c) Precise overlap for stacked calling in Fig 47 (b)

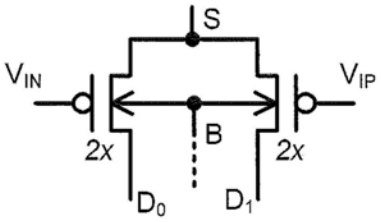
Fig. 47 stacked structure of current mirror for better aspect ratio

The red and green rectangles shown in Fig.47 (c) corresponds to the big top and bottom black rectangles in Fig.47 (b), each of which contains a current mirror structure. As the generated layouts are symmetrical with respect to the center in horizontal direction, just directly overlapping the blocks with top-down wiring is needed for stacked structure. As our module programs are written in Perl and generate SKILL™ codes of the corresponding layout for Cadence ™ Virtuoso ™, the overlap distance can be precisely calculated in *Perl* program before the total stacked layout is generated through loading SKILL™ codes into Cadence ™ Design Environment. So, flexibility is further enhanced and iterations can be launched to check PR issue such as overlapping in code form with no need to generate real Virtuoso ™ layout until the final solution is ascertained.

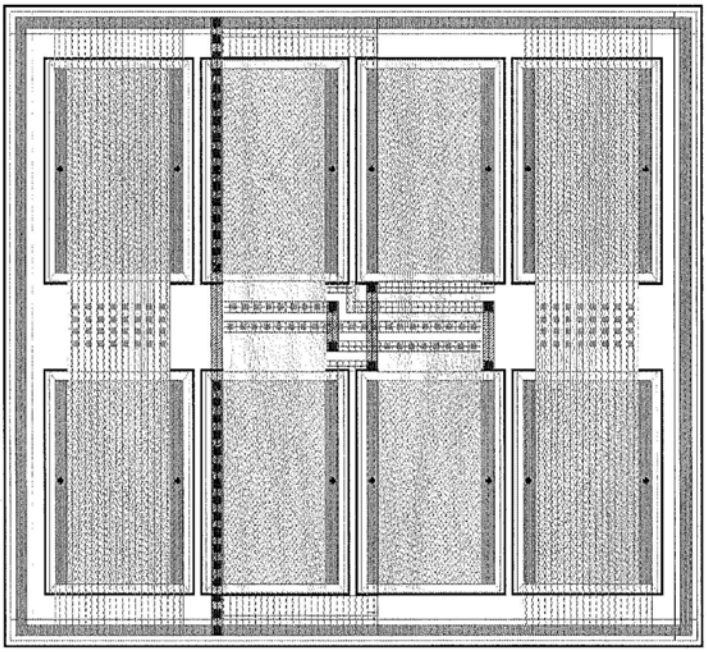
To facilitate the coming global placement and routing, module generator programs also automatically complete local routings with the terminals being right on the boundary, and record the information of size, aspect ratio, all I/O ports' relative locations and their properties for each of the generated blocks (in Perl code for PR evaluation). When these module layouts are called as an instance (usually also in the form of Perl code), the top-level P&R program can quickly determine their orientations to get the best aspect ratio and the best port-related global routings.

Fig.48 shows the schematic and layout of a typical PMOS differential pair. Besides unit matching and common centroid layout, to guarantee uniform current flow direction for better matching, the fingers do not share any common drain or source area with each other. As for local routing, the module generator provides an option to select whether or not a source is connected to the bulk for different cases (Source/Bulk connected by default to eliminate the body effect). Four dummy fingers lie on the two sides with PWR/GND rings

all around to create the same environment for each finger to minimize mismatch. For local routing, wire widths are case-dependent (actually quite conservative) and checked against the lower bound imposed by the given electromigration rule. To facilitate the following top-level P&R, parameterized differential-pair module generator, like the current-mirror one already mentioned, records the generated block's size, aspect ratio together with all the I/O ports' information (ports' properties, relative locations …) in the form of Perl code, extremely useful for PR engines.

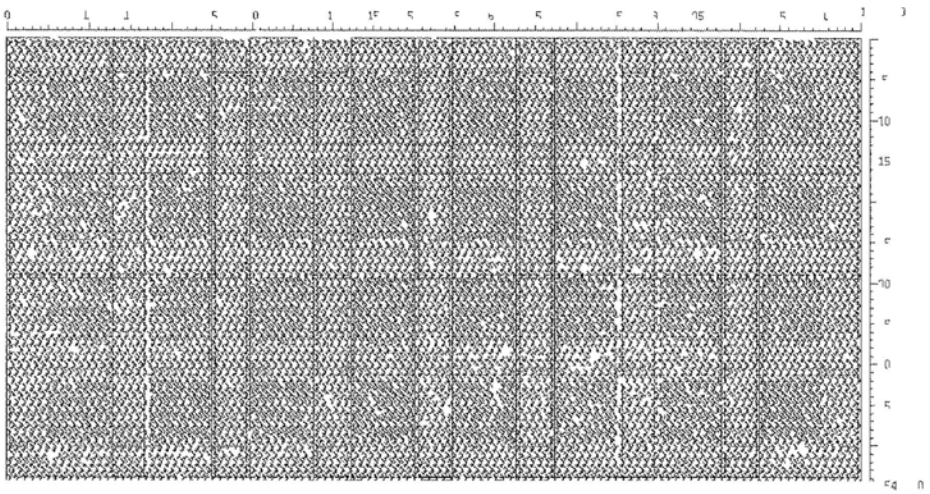


(a) Schematic of a typical PMOS differential pair

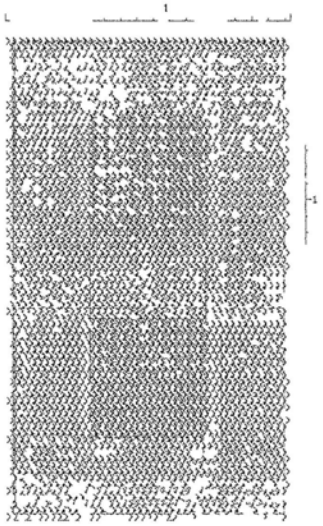


(b) Layout of the PMOS differential pair with S/B connected

Fig. 48 Schematic and Layout of a typical PMOS differential pair



(a) Stacked 2x8 of the unit shown in (b) with the given height



(b) Unit (MIM) capacitor layout

Fig. 49 module layout for (a) capacitor array (b) unit capacitor

Fig.49 (a) shows the layout for a MIM capacitor array (4x8) that is highly elastic in its size to fit with any adjacent blocks while Fig.49 (b) shows the unit cap layout that can be as small as 15umx25um, small enough to fit all kinds of vacant places in common layout to get both compactness and good regularity. The height of the capacitor array is elastic enough to achieve a better aspect ratio fitting with other parts. The horizontal length of the

capacitor array is then determined by both the ordered height and the total target capacitance. The wire widths, similar to the MG case for current mirror or differential pair, are case-dependent and checked against the design rules which are usually set conservatively to guarantee reliability and performance. Similarly, properties of I/O ports are also recorded to facilitate the coming global routing.

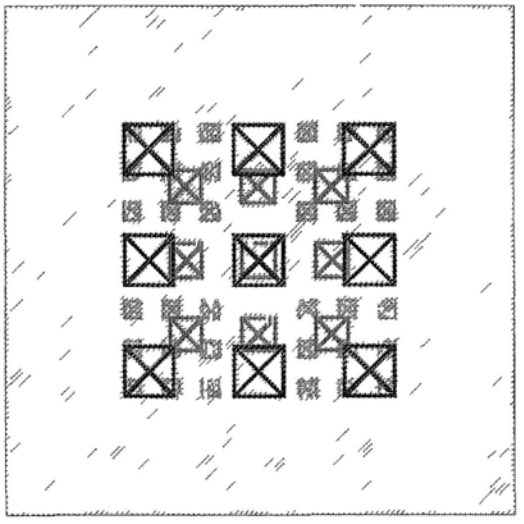


Fig. 50 Generated Layout for Layer-to-Layer (L2L) connection

Fig.50 shows the layer-to-layer connection part containing VIA arrays and different layers in a pre-determined regular shape (usually rectangular). As wire connection is always needed in both local routing (fulfilled by module generator) and global one (done by higher-level PR), via arrays for connecting different layers are frequently used. To simplify & accelerate the layout generation process, a module generator for layer-to-layer connection part is proposed to fulfill the layout formation for any connections between two metal layers in a pre-determined (usually rectangular) area. The detailed arrangement for via arrays is set beforehand to meet some specific rules, such as via density rules in our 130nm technology case imposed by foundry.

So far, we've introduced the module layout generators for some of the commonly-used classical circuit structures or basic components. All the module generator programs are written in Perl to generate SKILL codes for corresponding layout, which are quite easy to change, update and maintain. All the layout prototypes and their expanding algorithms (according to different user inputs) have already been tested to be free of physical verification errors.

With guaranteed layout quality for each called module block and comprehensive information on each block's property (size, aspect ratio, property & position of I/O ports), the rest tasks for achieving the ADA is greatly simplified and eased, namely higher-level PR and total physical verification. And as the library of module generators gets richer and richer, it can support more and more applications with the system flexibility being greater than ever.

5.4 Layout Synthesis

As the module generators achieve the module layout with guaranteed quality and report all the PR-related information (such as block size, I/O port position with orientation), our proposed PR algorithms will do the floorplan in an application-specific manner.

5.4.1 Evaluation Function

Unlike the conventional placement focusing on the area usage and total routing length as the top 2 priorities (Equation (2)) or evaluating more figures of merit (Equation (1)) in a linear combination style, a novel knowledge-based approach is proposed with serial consideration mode and further classification of each major figure of merit, handling the

issue in a style just similar to what analog designers consciously do. The ***equivalent*** evaluation function is shown as follows:

$$E = R(CA)[1 + \alpha \cdot AR(CA)][1 + \beta \cdot AR(CA, NCA) + \gamma \cdot TC(CA, NCA)] \qquad (10)$$

Where CA refers to Critical Area and NCA refers to Non-Critical Area. Normalized Functions *R*, *AR*, and *TC* evaluate *Routing quality*, *Aspect Ratio*, and *Total Compactness* (*Area Usage*). Coefficients α, β, and γ are application-dependent, which highly relies on input usage description. Usually, α, β, and γ are much smaller than 1. Relatively speaking, α is often set to be much larger than β while β is relatively larger than γ.

5.4.2 Proposed PR Flow

The huge difference among α, β and γ makes (10) nearly a serial flow in design consideration as described below:

Step 1: further classify each major figure of merit, namely *area*, *routing* and *aspect ratio* here. Based on application-specific design expertise, PR engine will divide *area* into *critical area* (*CA*) and *non-critical area* (*NCA*), which simultaneously defines *critical routings* (*CR*) and the non-critical ones. As for aspect ratio, the critical part and the rest are also correspondingly defined. But before the total layout, partial aspect ratios are often not that important as *CA* and *CR*.

Step 2: start *CR*-oriented floorplan for *CA*. An area containing several blocks is defined *critical* mainly due to each block's inner requirement and the routings among them. As each block layout generated by corresponding module generators (MGs) meets the requirements, routing concern is the top priority for PR in getting the critical area layout.

Step 3: on the premise of achieving good routings within *CA*, try optimizing the aspect ratio. For the initial run of step 3, as there is no clue for the shape & aspect ratio of *NCA*, few things will be done.

Step 4: Viewing the critical area as a whole, do the total PR that merges CA and NCA together. This is an Aspect-Ratio-oriented (when $\beta > \gamma$) or Area-oriented (when $\beta < \gamma$) flow. The default is aspect-ratio-oriented flow. Of course, on the premise of achieving good aspect ratio or area usage, try optimizing another is favored. However, in our case, due to reliability issue and migration (in the future), area usage or compactness is not deliberately cared. In this step, routings and layout for non-critical parts are to remain *acceptable*, which is guided by application-specific design expertise.

Step 5: Evaluate the total aspect ratio. If necessary, go to Step 3. If critical routings can't be changed drastically, check whether any block in critical area can be regenerated in a different aspect ratio. If not, stop the optimization trial and go to Step 4 and Step 5 with a poor-aspect-ratio flag, telling Step 5 there is no need to go back to Step 3 again.

In sum, for our proposed PR flow:

Priority: Layout quality (CA: CR; NCA: acceptable) >> Aspect Ratio > Area (default)

User can make $\gamma > \beta$ by setting a harsh requirement on Area. Then it may be:

Priority: Layout quality (CA: (MG+CR); NCA: acceptable) >> Area > Aspect Ratio

Fig.51 shows all these threads in a diagram. The principle is: compared with traditional linear combination style that optimizing any figure of merit means (more or less) sacrificing others, in our approach, layout quality (MG+CR) will never get sacrificed for either smaller area or better aspect ratio.

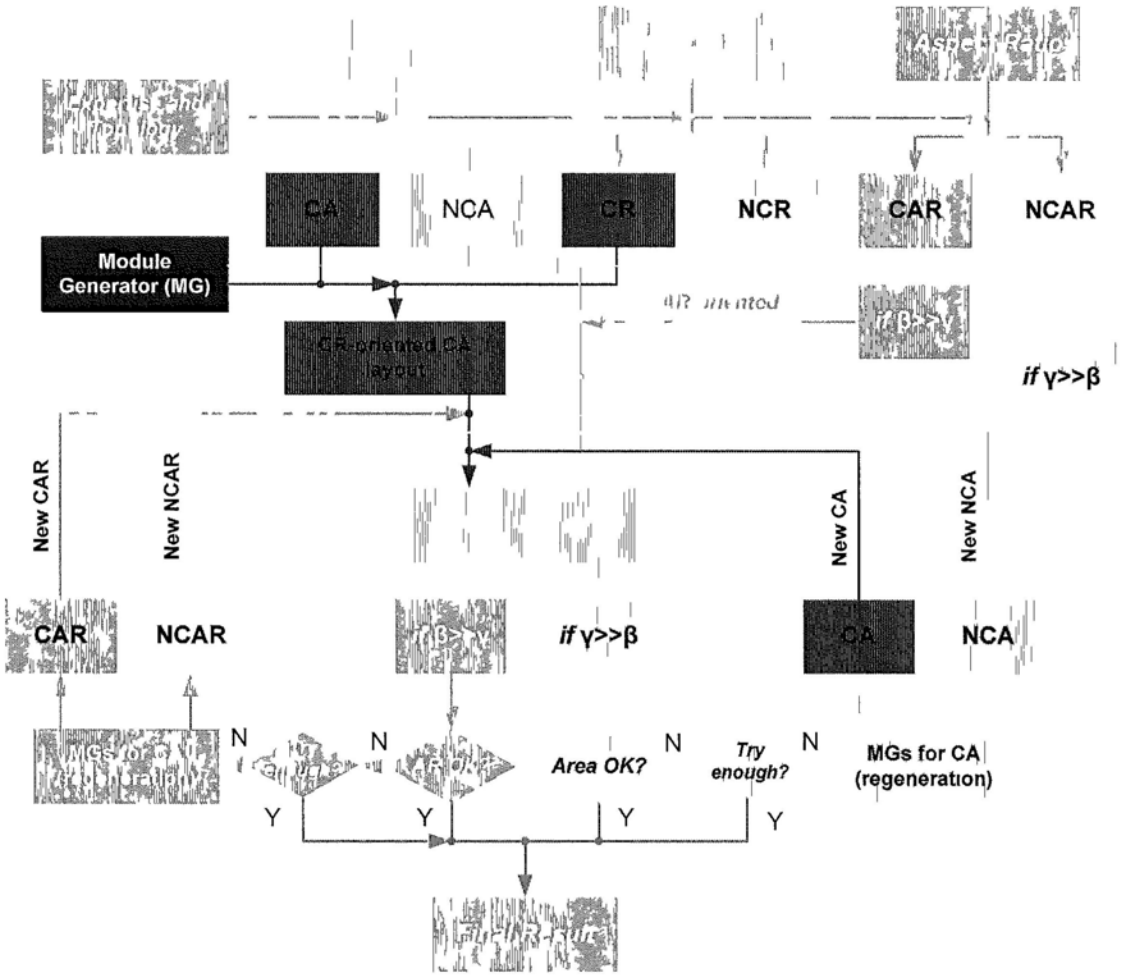


Fig. 51 The diagram of proposed PR flow

Although it may not be suitable for all the cases, it's reasonable for most common applications, which our proposed flow is ready to serve. Meanwhile, as the proposed flow doesn't consider the possible compromise between layout quality and other figures of merit, the platform can provide a very high speed (usually few minutes for "SPEC in, GDS out"), quite important for any design assistant.

Fig. 52 is the illustrative example of the proposed PR flow that divides critical part and non-critical part for both area and aspect ratio. Here, suppose that: 1) Block A, B, C and

D are generated by MG with guaranteed quality; 2) Relative positions of the I/O ports of block A, B, C, and D are known due to MGs' reports.

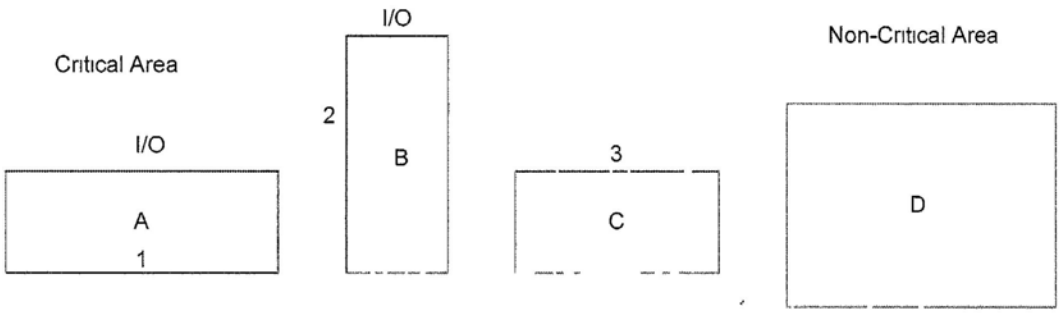


Fig. 52 Original generation: critical area and boundary identification (Step 1)

Fig.52 shows the original module blocks generation with critical area and boundary identification. As **Step 1** describes, Block A, B and C are identified as *CA* with 1 or 2 sides (red) regarded as the boundary where corresponding I/O ports are located. And block D and all the rest (not shown) are regarded as the NCA. Red "I/O" denotes that the ports on the corresponding boundaries are for I/O interconnection with the outside world while the red numbers mean the priority for inter-block routings inside the CA.

According to **Step 2** mentioned above, after knowing *CA* and *NCA*, *CA* will be placed in a *CR*-oriented manner. That means Block A, B, and C will be clustered in a form that the red boundaries are adjacent to each other according to the priority list while red "I/O"s are to be located at the boundaries of the total layout. Therefore, we have Fig.53.

Then we go to **Step 3** to optimize the total layout combining CA and NCA together. As the NCA part is relatively simple, aligning block D with other blocks is OK as already shown in Fig.50. So there are no further steps when user doesn't stress on aspect ratio.

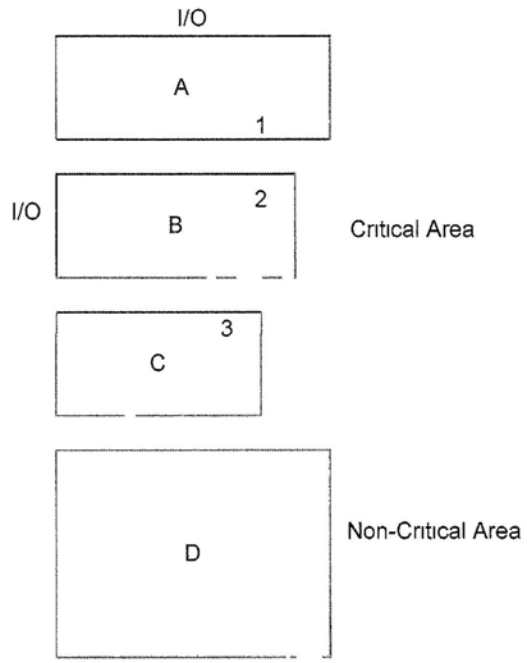


Fig. 53 CR-oriented CA with NCA (Step 2)

5.4.3 Proposed Flow for 2-stage Op Amp with PMOS-input

Now let's analyze the 2-stage op amp case with PMOS differential pair (DP) input and Miller compensation capacitor realized by MIM capacitance. Herein we don't use MOS capacitance due to its huge volatility (voltage-dependent).

Schematic of the 2-stage op amp case with PMOS DP input and Miller cap is already shown in Fig.38. As the MIM capacitor is quite large, usually in the magnitude of at least a few hundred um^2 ($1fF/um^2$ based on process information), it can be put aside when we optimize other parts' layout. Also, as the capacitor can be generated quite regularly with the size in one dimension being custom-built, the rest part is better to be rectangular.

According to Fig.38, the other parts (besides C_C, no R_Z) of a 2-stage op amp can be divided into the following blocks:

1) 1^{st} current mirror (CM1) structure containing M_2, M_3 and M_7.

2) Differential pair (DP) structure consisting of M_0 and M_1.

3) 2nd current mirror (CM2) structure that includes M_4 and M_5.

4) Output stage MOS transistor M_6.

Based on the current mirror generator and the typical prototype shown in Fig.46 and Fig.47, the shape and the boundary where I/O ports are located are quite similar to Block C in Fig.52 and Fig.53. Considering CM1 needs to provide I_{BIAS} and the op amp output (V_{OUT}, drain of M_7), CM1 needs to have one side located at the boundary of total layout for stretching out I/O ports for I_{BIAS}, V_{OUT} and the possible Power/Ground. In addition, it has to leave one side adjacent to the DP. Considering the inner structure of current mirror, these two sides are better to be top and bottom ones with top-down wirings, similar to the blue wirings in Fig.47(b) and Fig.47(c), which both serve the I/O ports linkage and the inter-block routings.

Therefore, the CM1 is obviously the same case as Block A described above, a rectangle with top & bottom sides to be wiring-related. CM2 is quite similar to Block C. And for DP layout, according to Fig.48 (b), its common source line is on the left while the two gated inputs can come in either from the top or from the bottom. So just rotating it by 90° clockwise will make it just the same as B, where two gated inputs stretch out from the left while the other inter-block routings come from the top or bottom sides. Finally, for M_6, as its gate will be tied to Miller cap (while its drain will be tied to V_{OUT} with C_L) and it's relatively large in size, the routing is not as critical as others for minimizing the parasitic capacitance. For routing length, as mentioned above, we don't rely on R_Z to achieve PM. Therefore, a little parasitic resistance caused by routing won't affect too much since large size of M_6 means wide routings are feasible and nonzero R_Z may even improve the PM.

In sum, based on related design expertise, CM1 (M_2, M_3 and M_7), CM2 (M_4 and M_5) and DP (M_0 and M_1) are defined as critical parts while M6 and Miller cap are the rest. For critical routings, just similar to the distribution of red lines in Fig.53, the I/O ports, the linkage between CM1 and DP, the one between DP and CM2 are the critical routings with their priority list shown below:

1) I/O ports (mainly due to the two gated inputs for DP)

2) Linkage between CM1 and DP (CMRR/ICMR-related)

3) Linkage between DP and CM2 (symmetry needed*)

**Note: as stage 1 is single-output, actually the linkage between M_1 and M_5 should be shorter than the one between M_0 and M_4. Actually, no exact symmetry requirement comes for 3), which is why 3) bottoms in the priority list.*

All these considerations above seemingly suggest a similar floorplan for the 2-stage PMOS input op amp with Miller Compensation. And based on the netlist of Fig.54 gained through circuit synthesis, Fig. 55 shows the original layout of CA part when its floorplan is simply copied from Fig.53.

```
.SUBCKT  pmc_op  vdd  vss  vip  vin  VOUT
M0       C   vip      E     E   P_12_SPL130E   W=13.98u    L=1u     M=2
M1       A   vin      E     E   P_12_SPL130E   W=13.98u    L=1u     M=2
M2       B     B    vdd   vdd   P_12_SPL130E   W=3.26u     L=0.5u   M=1
M3       E     B    vdd   vdd   P_12_SPL130E   W=3.26u     L=0.5u   M=10
M4       C     A    vss   vss   N_12_SPL130E   W=3.5u      L=1u     M=2
M5       A     A    vss   vss   N_12_SPL130E   W=3.5u      L=1u     M=2
M6    VOUT     C    vss   vss   N_12_SPL130E   W=4.08u     L=1u     M=16
M7    VOUT     B    vdd   vdd   P_12_SPL130E   W=3.26u     L=0.5u   M=40
C0    VOUT     C   3.13p
I0       B   vss  ibias
C1    VOUT     0  cload
.ENDS
```

Fig. 54 Netlist gained by circuit synthesis (where ibias=1.4uA, cload=5p)

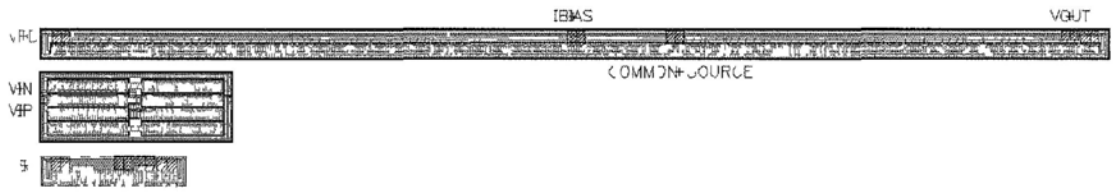


Fig. 55 Original placement for CA of 2-stage op amp

Obviously, the aspect ratio is too far away from acceptable. Herein, our module generators record all the sizes and relative position information. When the modules are placed with their left side aligned, as show in Fig.55, our program will check each block's horizontal length. If any length is more than double the maximum of others, our program will go back to Step 3, checking whether or not the corresponding module generator can re-generate that module layout with a better aspect ratio.

In this case, the program calculates and finds out that the length ratio (LR) is about 5. As it's a current mirror, the system will check the set of finger numbers and the finger width. If the finger numbers' GCF (Greatest Common Factor) can be the multiples of LR (herein, 5), then the program will re-generate the CM layout with all the finger numbers divided by the LR (must be rounded to an integer), and called it LR times by using the stacked structure to make the aspect ratio sound. If GCF is not the multiples of LR but finger width is large enough (process dependent), then the program will use a thinner width and also call it more times in a stacked structure to compensate.

Herein, GCF of M_2, M_3 and M_7 (1:10:40) is not the multiples of LR if we include M_2. But as power SPEC is not too harsh and GCF of M_3 and M_7 (10:40) does be the multiple of LR (5), we can multiply nf_2 and I_{BIAS} by 5 times and then the GCF for M_2, M_3 and M_7 can be the multiples of LR. Shrinking the finger width is not used because the divided width would be small (<1u) enough to cause some nonlinear effects.

Fig 56 shows the re-generated placement for CA part of the 2-stage op amp, also with the critical routings And Fig 57 shows the layout covering both CA part and one of the NCA parts, the M_6 arrays

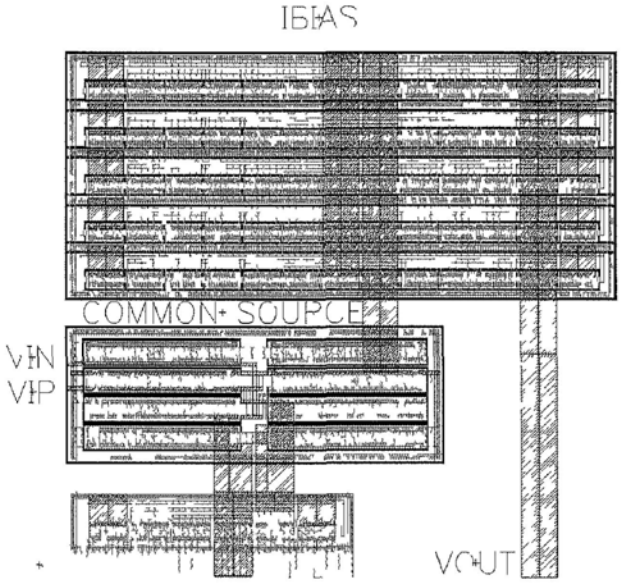


Fig 56 CA layout for 2-stage op amp with routings

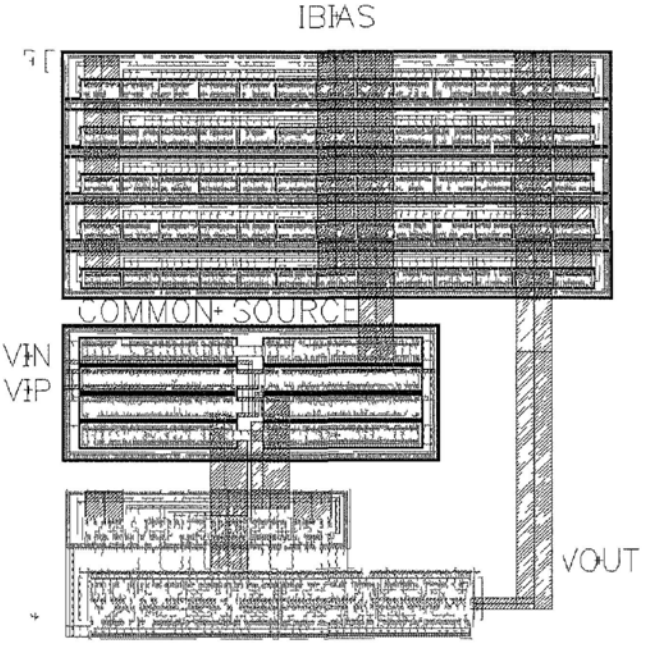


Fig 57 Layout for 2-stage op amp except Miller Capacitor

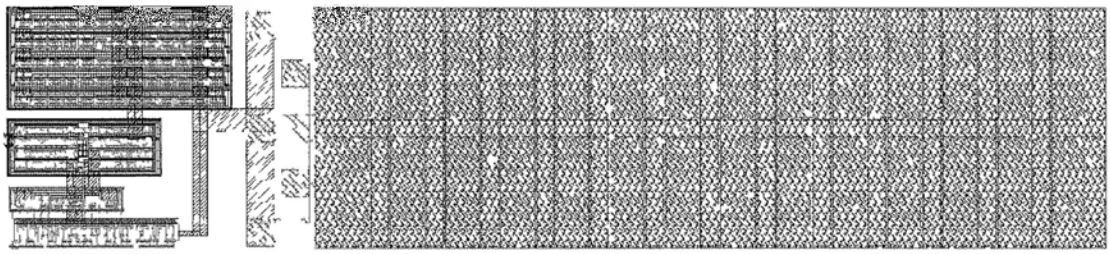
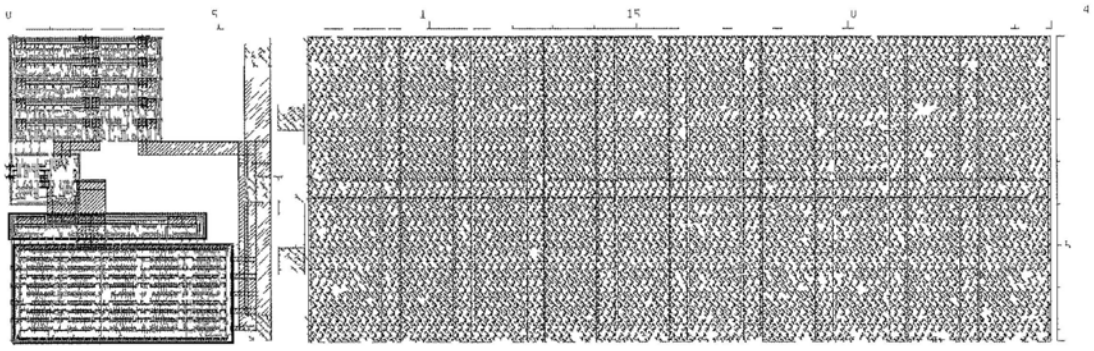


Fig. 58 Total layout of 2-stage op amp with Miller compensation (PMOS-input)

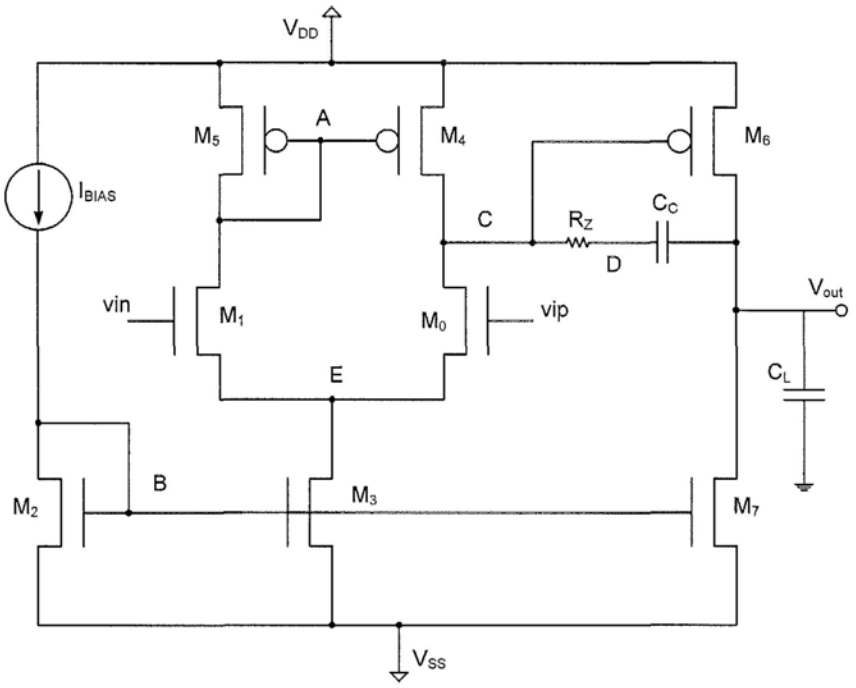
The total layout of the 2-stage op amp with PMOS differential-pair input and Miller compensation is shown in Fig.58. The area is about 250x55um^2. Obviously, the flexible capacitor array is adjusted to the same height as the MOS part (like standard-cell style), making the total layout quite regular with a reasonable aspect ratio. All the I/O ports are placed near the boundary: I_{BIAS}, V_{OUT} and V_{DD} at the left top, V_{SS} at the left bottom and V_{IN} & V_{IP} on the left side with a ground line in between as a shield (intra-block structure).

5.4.4 Proposed Flow for other Op Amps

So far, the layout for 2-stage op amp (PMOS differential pair input) is finished. Besides this type of applications, of course, our platform supports the corresponding NMOS type for 2-stage op amp that is as shown in Fig 59, and the NMOS/PMOS-differential-pair input single stage current mirror op amps that are shown in Fig.60 & Fig.61 respectively.



(a)



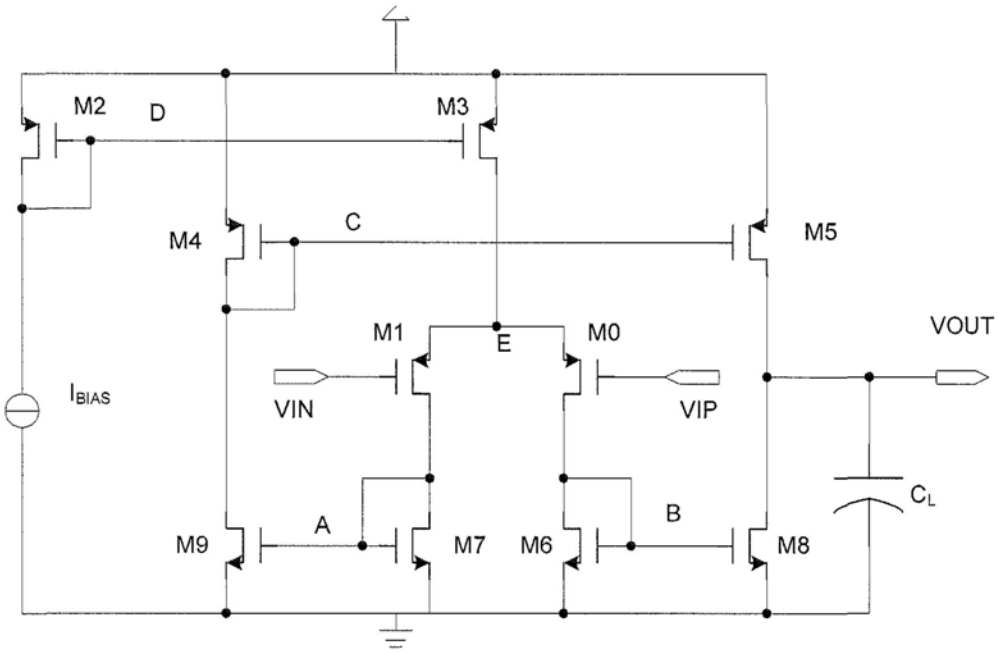
(b)

Fig. 59 (a) total layout of 2-stage op amp with Miller Compensation (NMOS-input) (b) the corresponding schematic

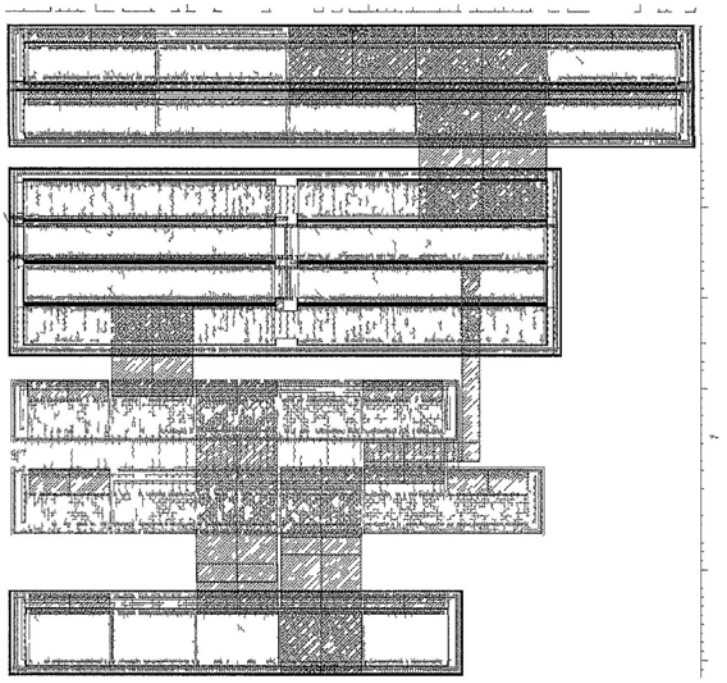
Due to the low mobility of PMOS, the NMOS input differential pair can be much smaller than the PMOS current mirror load. Because of the same reason, the output stage PMOS transistor has to be very large in order to achieve $g_{m6} > 10g_{m0}$.

As the PMOS current mirror load differs from the NMOS differential pair too much in horizontal width while they are both smaller than the corresponding CM1 (M_2, M_3, and M_7), after using the stacked structure, there are still some margins, which shows there is still something left to be improved. Of course, the tradeoff between speed and area usage has to be considered.

Fig. 60 (a) and (b) show the schematic and corresponding layout for single-stage current mirror op amp with PMOS input while Fig.61 shows the corresponding schematic and layout for single-stage current mirror op amp with NMOS input.

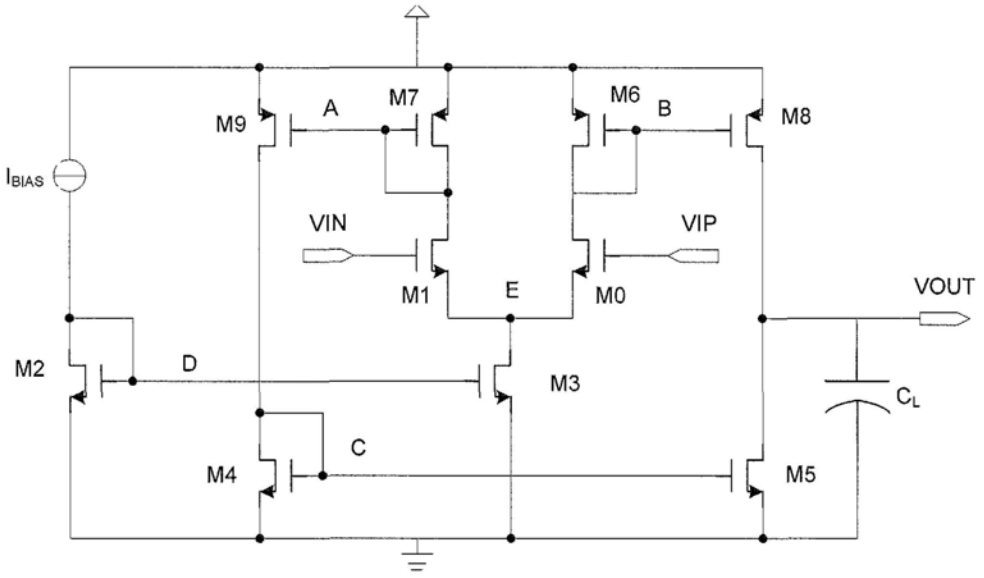


(a)

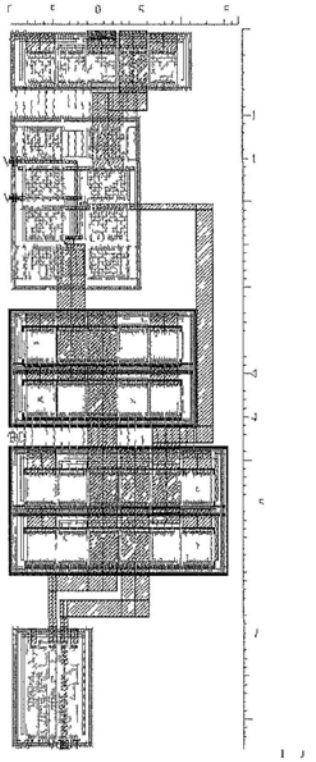


(b)

Fig 60 (a) schematic of single-stage current mirror op amp (PMOS-input) (b) the corresponding layout



(a)



(b)

Fig. 61 (a) Schematic of single-stage current mirror op amp (NMOS-input) (b) the corresponding layout

For single-stage current mirror op amp, due to the absence of on-chip capacitor, the area is quite small compared with the previous 2-stage ones. Therefore, the aspect ratio is no longer an important figure of merit. CR-oriented layout for CA parts is also applicable for these cases with stacked structure similarly used in Fig.61 (b) for better regularity.

Up to now, the layout synthesis is finished. Before going to post-layout simulation, our system will launch the automatic physical verification to check the results.

5.5 Physical Verification

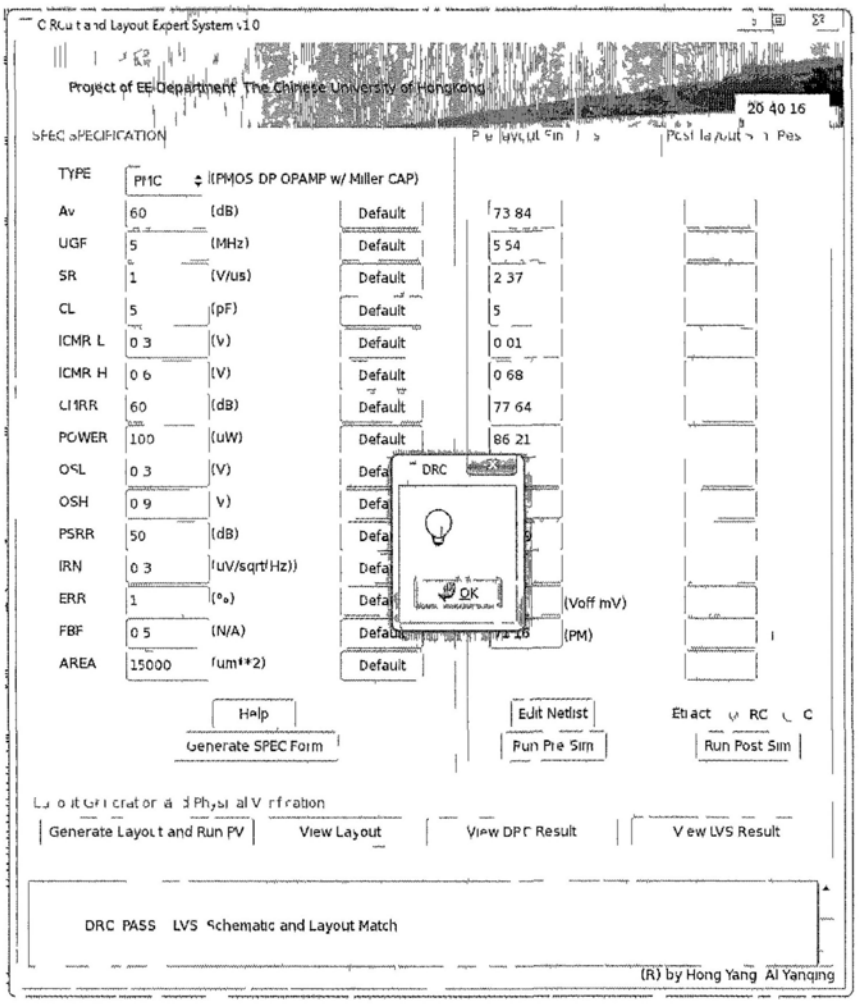


Fig 62 Automatic DRC results for 2-stage op amp (PMOS-type)

Fig.62 shows the automatic DRC results for 2-stage op amp case with PMOS differential pair input and Miller compensation while the automatic LVS results are shown in Fig 63 Since our platform checks the error file (*.err) that contains all the errors, a DRC-clean check will show no content as there is nothing in the error file

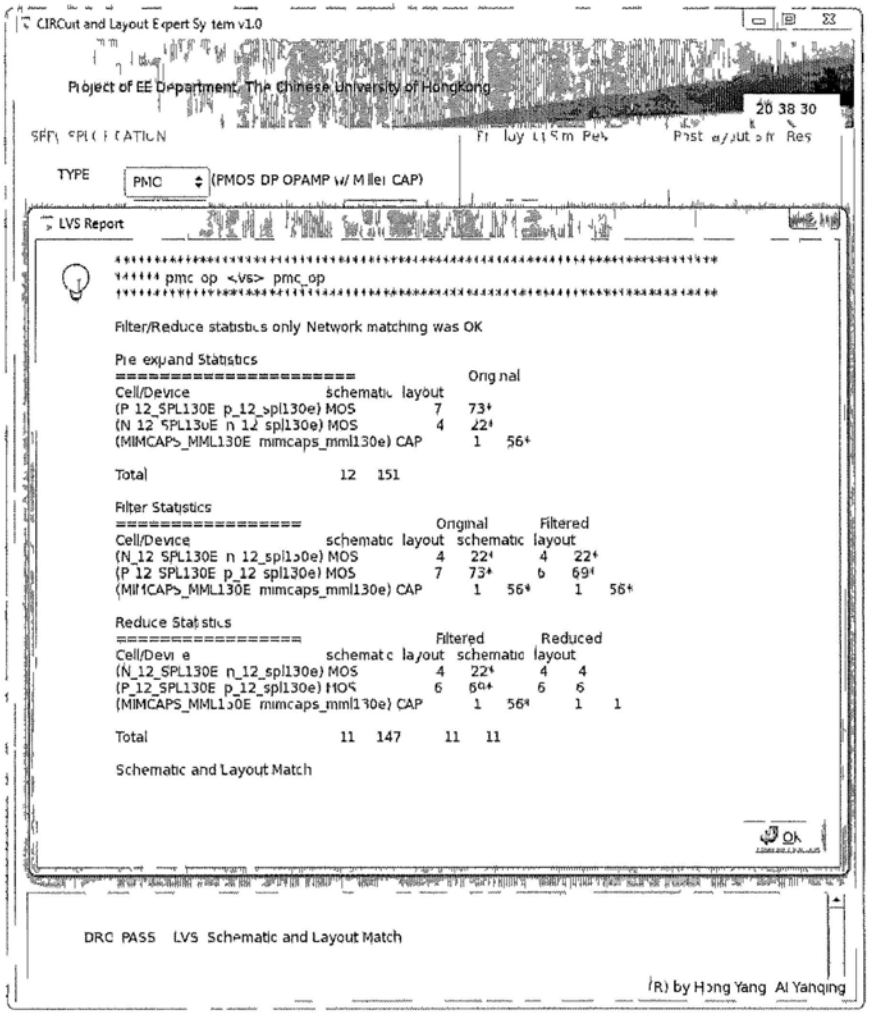


Fig. 63 Automatic LVS results for 2-stage op amp (PMOS-type)

As all the module generators are relatively conservative in local PR and the top-level PR engines are also conservative due to the relatively low priority for compactness, automatic check for DRC and LVS is quite easy to get clear. As it's almost the same case for other op amps, the similar results won't be shown here tediously.

5.6 Post-Layout Simulation

After the automatic layout generation and physical verification, the layout information is ready to support the post-layout simulation with layout parasitic parameters. For our system, it provides two kinds of settings for post-layout simulation: extraction of parasitic capacitance only (C) or extraction of both the parasitic capacitance and the parasitic resistance (RC).

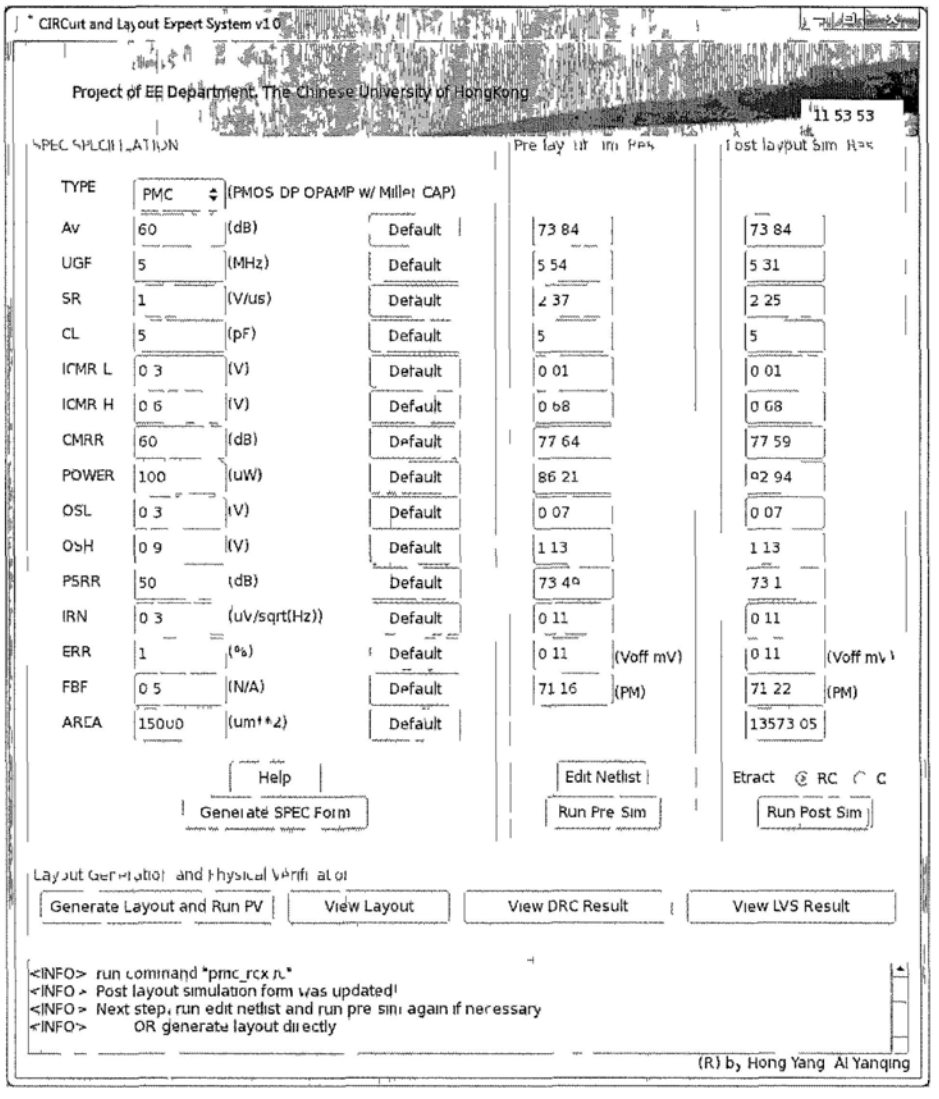


Fig. 64 Comparison of results after Post-layout Simulation

Once user sets his or her choice and clicks "Run Post-Sim", our system will automatically call Assura^{TM} to do the corresponding RC extraction and then start HSPICE post-layout simulation based on the generated netlist full of C/RC parasitic information. Finally, the corresponding results will be shown to user via GUI interface, just as Fig.64 shows.

As both circuit synthesis and layout synthesis are application-specific, the engines can be made relatively simple, achieving a very fast speed. The total running from "SPEC in" to "GDS out" costs just a few minutes. Even including the post-layout simulation, ten minutes is more than enough to go through the total flow. With the provided manual intervention and a very fast speed, this design assistant can be even more flexible and attractive.

5.7 Testing Results

In this section, the testing results for our chip (PMC type) are presented as follows.

Fig. 65 shows the schematic of testing circuit for offset voltage, where $R_1 = 1.5k\Omega$ and $R_f = 150k\Omega$.

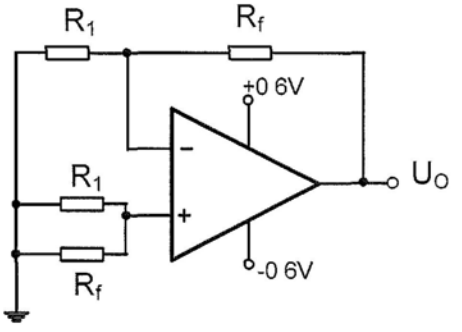


Fig. 65 schematic of testing circuit for Voff

$$V_{OFF} = -\frac{R_1}{R_1 + R_f} \cdot U_O$$

Testing results for V_{OFF} show that it varies from -1.77mV to -0.07mV. It depends on process fluctuation, bonding, PCB design and many other factors.

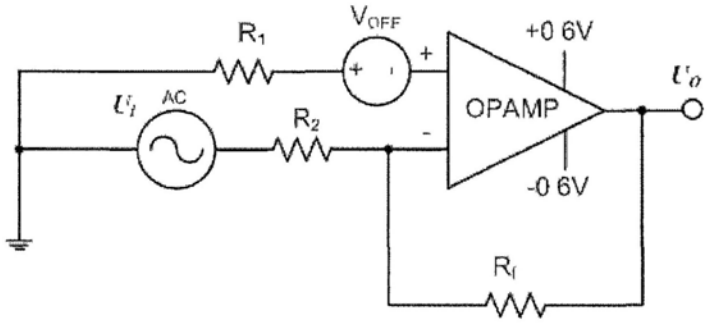


Fig. 66 testing schematic for open loop gain A_V

Fig. 66 shows the circuit for open loop gain test. As the expected open loop gain (A_V) is quite high (around 5000), R_f is chosen to be much larger than R_2 so that 1+AF cannot be rounded as AF (where F is the feedback factor that equals $-R_2/R_f$). Herein, we set

$$R_f = 200k\Omega \quad R_2 = 1k\Omega \quad R_1 = R_2//R_f \approx 1k\Omega$$

The input & output AC signals are shown in Fig.67 for such a test under the close-loop configuration. The green line is the output signal waveform, bearing a peak-to-peak value of around 718mV. The yellow line shows the input ac signal. Although the oscilloscope shows the peak-to-peak value (average) is 6.83mV, the actual peak-to-peak value for this sinusoidal input is much less due to the noise issue (caused by signal generator, routing and oscilloscope itself). Herein the input sine wave is estimated to be 4.5mV (peak-to-peak), therefore we have:

$$\left|\frac{v_{out}}{v_{in}}\right| = \left|\frac{A}{1 + AF}\right| = \frac{718}{4.5} \approx 159.56$$

$$\because F = \frac{R_2}{R_2 + R_F} = \frac{1}{201} \qquad \therefore A \approx 773.82 \approx 57.77dB$$

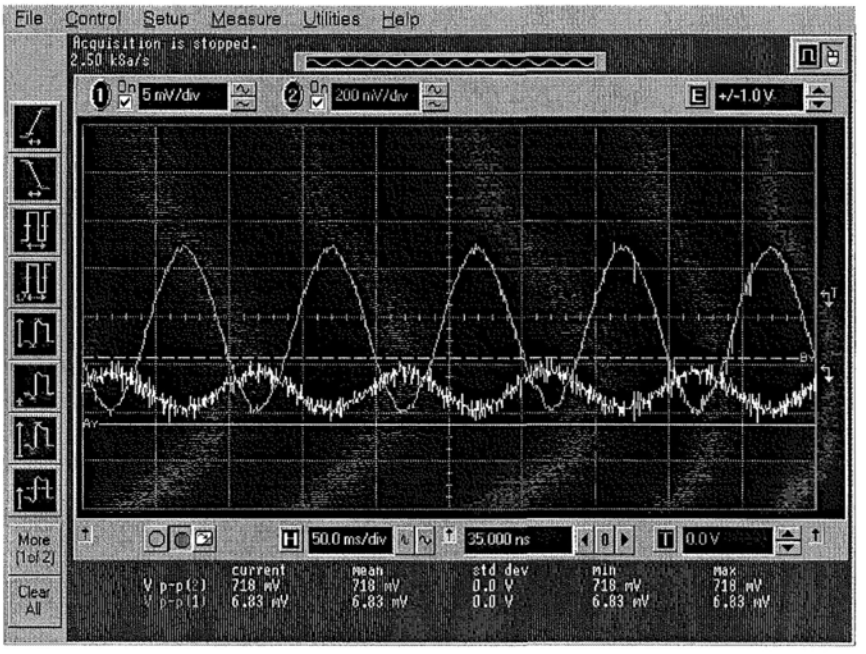


Fig. 67 waveforms of input & output ac signals for A_V measurement

Fig.68 shows the UGF measurement waveform when we increase the frequency until the input and output waves have the nearly the same amplitude. The measured UGF is about 1.02MHz.

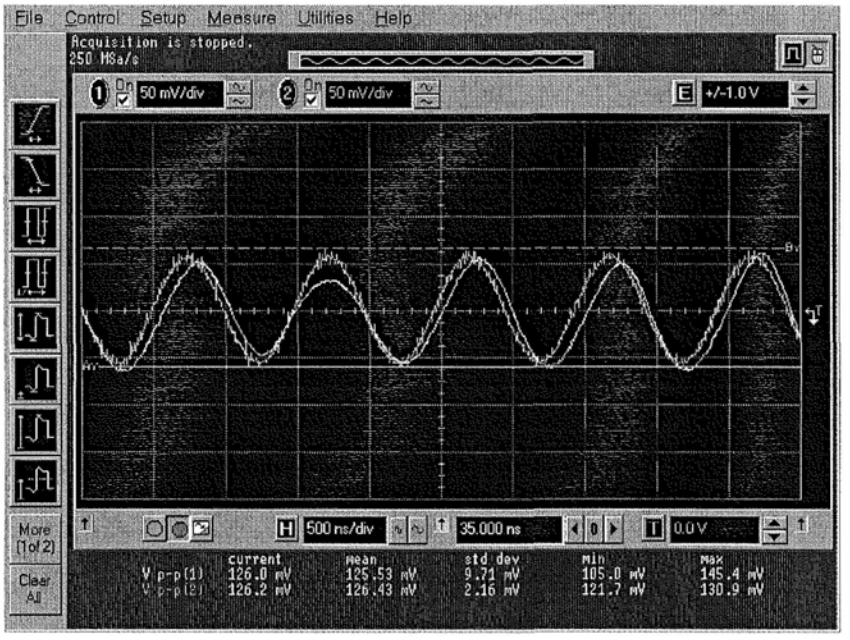


Fig. 68 waveforms for UGF measurement

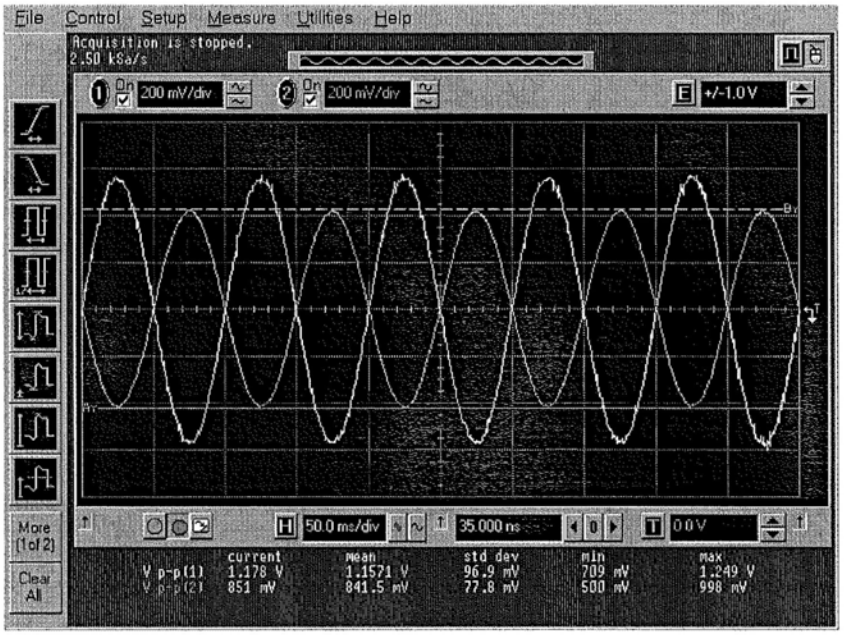


Fig. 69 waveforms for OSW measurement

Fig.69 shows the waveforms of input & output ac signals for the output swing (OSW) measurement. According to the test, the maximum range within which the output curve doesn't get distorted is about 0.05~1.15V with the corresponding input V_{PP} being 850mV.

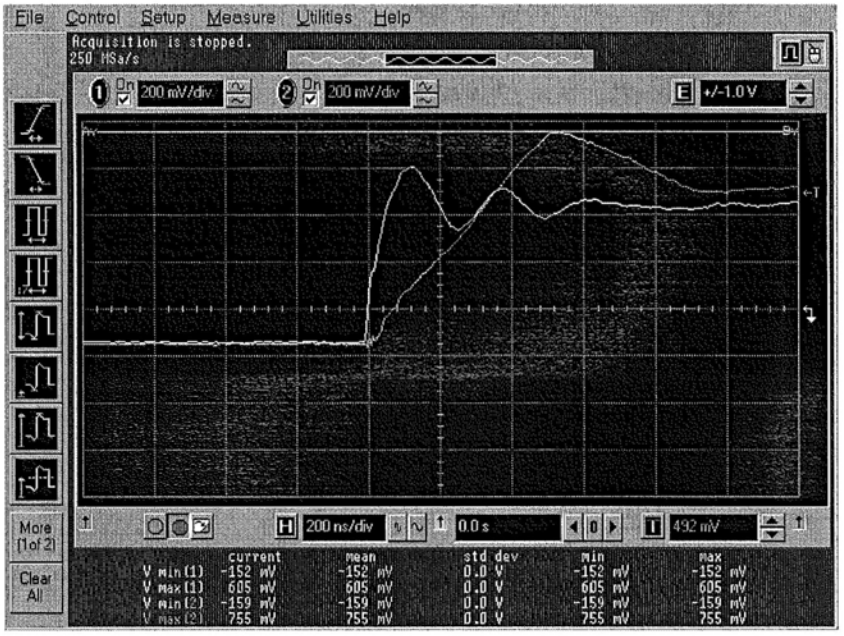


Fig. 70 Rising edge SR test waveform

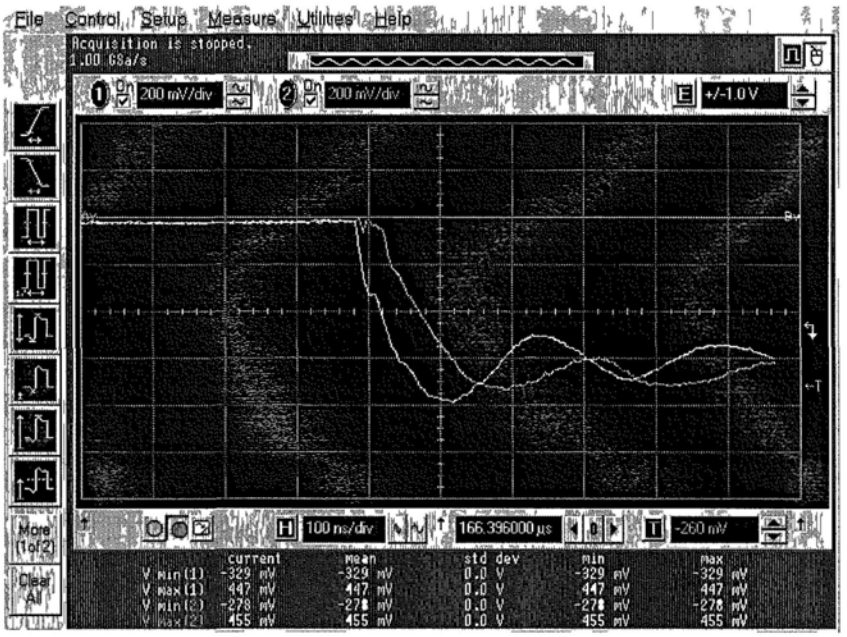


Fig. 71 Falling edge SR test waveform

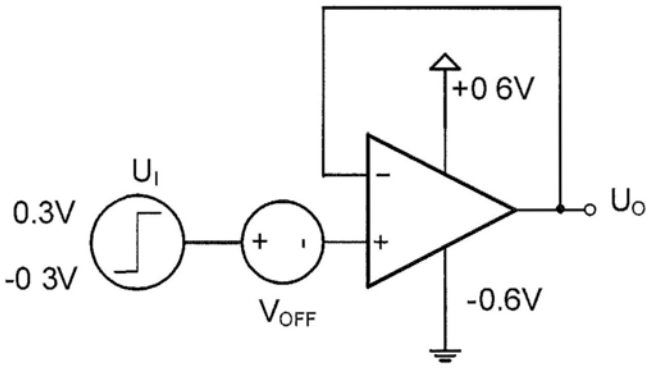


Fig. 72 schematic for SR test

Fig.70 and Fig.71 shows the input and output voltage waveforms for SR measurement while Fig.72 shows the schematic. The yellow and green curves are the input and output voltages respectively in both Fig.70 and Fig.71. Measuring 30%-to-70% rising & falling times, we have the SR results shown as follows:

$$Rising:\ SR \approx 1.8\ (V/us)$$

$$Falling:\ SR \approx 3\ (V/us)$$

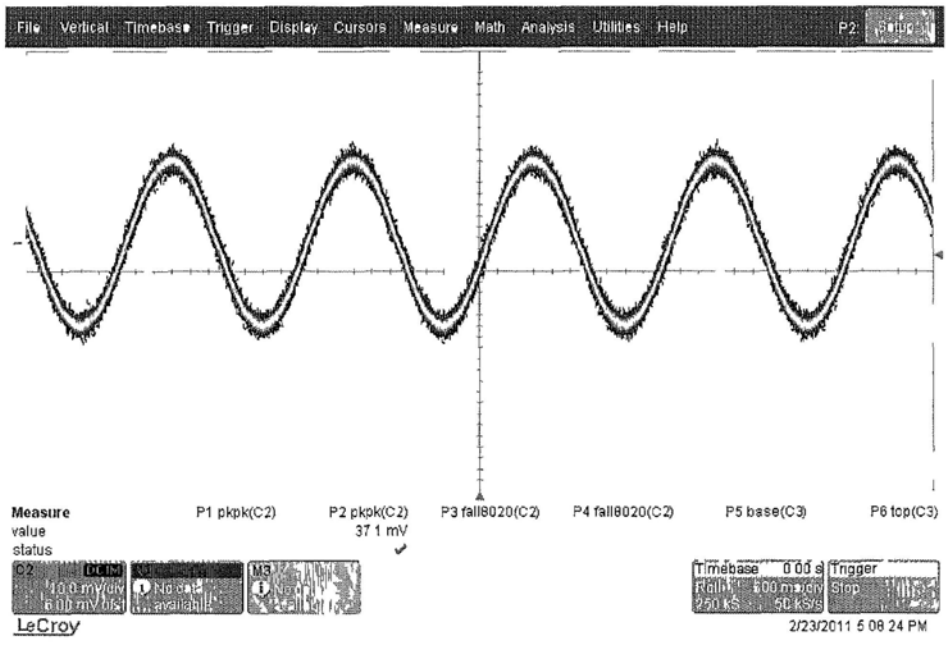


Fig. 73 Common mode gain test waveform

For CMRR test, Fig.73 shows the common mode output waveform under a 300mV (V_{PP}) 1Hz sinusoidal wave input. Therefore, we have the low frequency common mode gain as follows:

$$Common\ Mode\ Gain:\ A_{vc} = \frac{37.1mV}{300mV} \approx 0.12\ V/V$$

$$\therefore CMRR = \frac{A_{vd}}{A_{vc}} \approx 76.2\ dB$$

As for the power, the measured DC current flowing from VDD is 79uA. Therefore, total power consumption is

$$Power = 1.2 \times 79 = 94.8\ (uW)$$

For PSRR, the measurement result is far from the expected results. Just like the open loop gain Av measurement, in PSRR measurement, the small output AC signals are highly affected by noise as these ac signals & noise are in the same magnitude while the latter is quite hard to be completely eliminated.

Fig.74~Fig.76 shows the measured PSRR waveforms for PSRR+@1KHz, PSRR-@1KHz and PSRR@50Hz respectively.

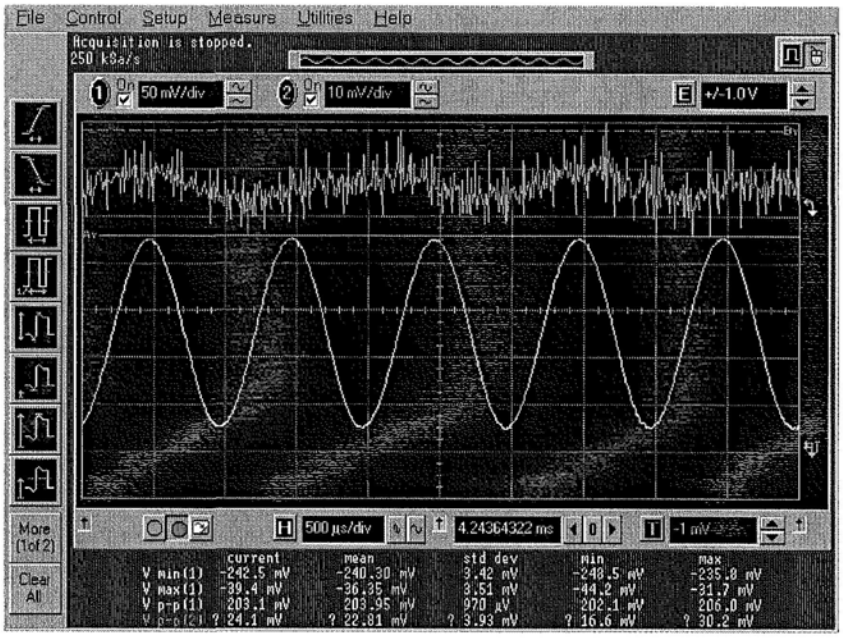


Fig. 74 waveforms for PSRR+ @ 1KHz

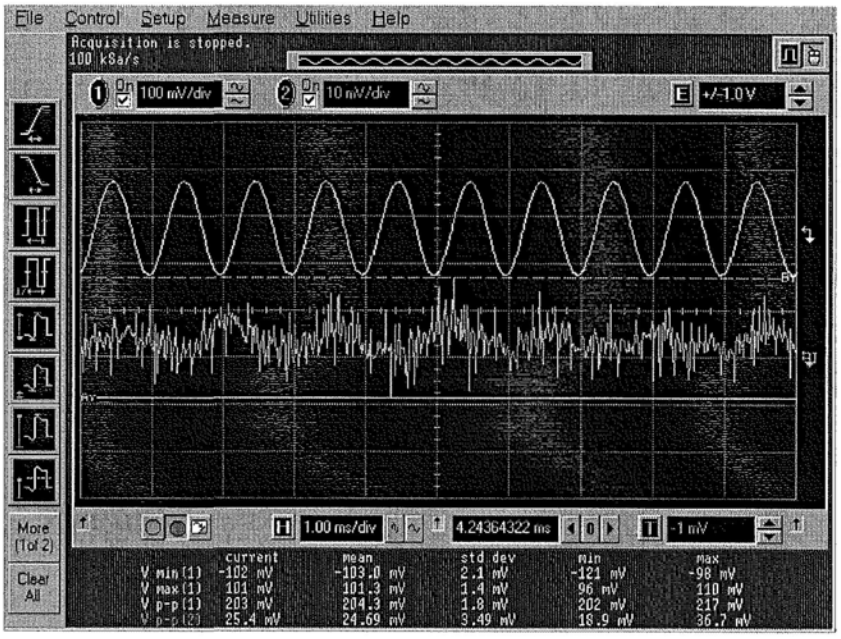


Fig. 75 waveforms for PSRR- @1KHz

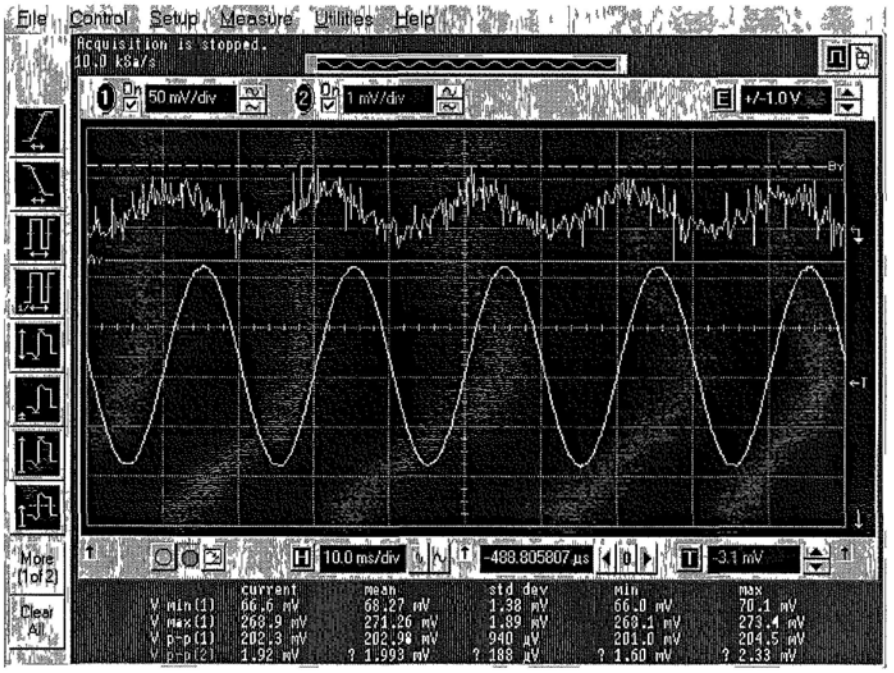


Fig. 76 waveforms for PSRR+@50Hz

As Fig.74~Fig.75 show, if we eliminate the noise from the green Vout curves, the V_{PP} for v_{out} is at most 10mV, meaning that the PSRR+/- @ 1 KHz is larger than 26.1dB. For Fig.76, the similar deduction shows the PSRR+ at low frequency (e.g. 50Hz) must be larger than 46dB. However, due to the noisy v_{out}, the exact values for PSRR+/- are quite hard to get.

Table 3 shows the total measurement results and the comparison between simulation and measurement. From the comparison, we can see that SR, CMRR, Power, V_{OFF} and Area are acceptable while A_V, C_L and Av/C_L related figures of merit (including PSRR, OSW, ICMR, UGF and PM) are highly degraded due to the inaccuracy of small ac signal measurement and uncertain (much larger than expected) parasitic loading capacitance. Under such a case, two figures of merit are not measured, namely Input-Referred Noise (IRN) and Phase Margin (PM). Detailed information is shown in the footnotes.

Table 3 Simulated & measured results for op amp (PMC type)

Figure of merits	User-input SPEC	Expected SPEC	Experimental Results
A_V	60dB	73dB	57 77dB[1]
UGF	5MHz	5.5MHz	1.02MHz[2]
SR	1V/μs	2.3V/us	1.8~3V/us
C_L	5pF	1~5pF (5pF as default)	>10pF[2]
CMRR	60dB	77dB	76.2dB
Power	100μW	93 μW	94.8μW
OSW	0.3~0.9V	0.07~1.13V	0.05~1.15V[1]
V_{OFF}	-	0.11mV	-1.77mV~-0.07mV
ICMR	0.3~0.6V	0.01~0.68V	0.387~0.812V[1]
PSRR	50dB	73dB@1k, 50dB@10k	>26.1dB@1k[1], >46dB@50[1]
Area	15000μm^2	<13573μm^2	<13573μm^2
IRN	0.3μV/$\sqrt{Hz}$	0.11 μV/$\sqrt{Hz}$	N/A
ERR	1%	-	-
FBF	0.5	-	-
PM	60°	71°	N/A[2]

Note:

[1] Due to the resolution constraints & inherent noise caused by signal generator, oscilloscope and environment, small signals lower than 10mV are consistently mixed with obvious noises that can hardly be eliminated. As a result, open loop gain A_V is highly degraded in measurement. So do the figures of merit related to A_V, such as PSRR. OSW & ICMR are also affected. OSW here is the maximum output swing without distortion, not expected high-gain region as what's originally defined. And measured ICMR is the corresponding input range to get the measured maximum undistorted OSW while the simulated ICMR is the calculated results based on simulated data and classical formulas to ensure every MOS is in saturation.

[2] Due to the signal generator and oscilloscope, esp. the latter, the output loading capacitance is much larger than 5pF, out of the supported range. As a result, UGF is highly decreased while PM in such a case seems almost pointless.

5.8 Pros and Cons

Compared with our ideal goal for ADA and other available methods, especially the traditional optimization-based methods, our proposed method and system differs in the following points:

Pros:

1) Fast speed due to the simplicity of each application-specific engine

2) Guaranteed layout quality due to MGs, novel evaluation function/flow and knowledge-driven PR, going in the same way as analog designers consciously do.

3) Ready expansion to assimilate novel design ideas & topologies as all the good manual work can be abstracted into templates.

4) Avoid lengthy heuristic approach asking too much for users.

Cons:

1) Library for MGs and PR engines needs to be enriched for wide application. It's a one-by-one task.

2) Design space exploration when there is overlap for many kinds of solutions, e.g. topology selection. Special knowledge-based engines are needed.

3) How to quantify the usage description and use it as a good guide.

4) Process independent issue: PPI (Process Program Interface) is needed.

Of course, for the traditional optimization-based methods, 2~3) (or the equivalent) of Cons are the same challenges. And for 1) and 4), they are totally feasible, which just a time issue to make further development and refine the codes.

Table 4 and 5 show the comparisons on circuit synthesis and layout synthesis between our proposed approach and several typical or prevailing ones respectively.

Table 4 Comparison on circuit synthesis

Circuit synthesis	CIRCLES	NeoCircuit™	GP	IDAC
Year	2010	2004~present	2001	1987
Application specific	Yes	Yes	Yes	Yes
Simulation (S) or Calculation (C)-based	S	S	C	C/S
Process (um)	0.13	≤0.18	0.8	3
Speed	< 60s, OP	lengthy	1~2s, OP	4h, ADC
Opt method	Knowledge & simulation &rule-based	N/A	Geometric programming	Knowledge-based
Math Constraints	No	unknown	polynomial	no
Opt Inclination	Reliability	unknown	One SPEC	Case-dependent
Additional input	Usage description	Many*	No	A lot**
Process independent	No	No	No	No
Heuristic approach	No	Yes	No	Yes
Related Layout Gen.	CIRCLES	NeoCell	No	ILAC

Note: Many*: Cadence NeoCircuit needs many manual tuning for MOS size as an example. Actually it's rarely used by analog designers. So the available information for this commercial tool is quite limited.

A lot**: additional input includes: matching of components, specifying some degrees of freedom which exist in almost all designs. Also, these degrees of freedom are process and application specific with default values at hints.

Table 5 Comparison on layout synthesis

Layout Synthesis	CIRCLES	NeoCell™	Ref [78]	ALG	LAYGEN	ALADIN	ILAC
Year	2010	2004~present	2010	2009	2006	2006	1988
Application specific PR	Y	N/A	N	N	N	N	N
Speed	<5 mins	slow	<1min (PR only)	≤ tens of mins	not mentioned	tens of mins	≤ tens of mins
MG	knowledge-based	knowledge-based/user-defined	N/A	knowledge-based	knowledge-based	knowledge-based	knowledge-based
MG features[1]	PR-related info report; auto-reshape	User-defined	N/A	min nodal parasitic cap	High-level module description	Module description language	no
Cost function	Serial mode; classification	unknown	combination, parallel	linear comb parallel	combination	linear comb parallel	combination, parallel
Weight input	Build-in priority list (knowledge-based)	unknown	User input	User input	User input	User input	User input
PR Features	Classification of figure of merit as area, routing …	Now part of Virtuoso editor w/ enhancement	Symmetric Routing	Tunable (directive to automatic)	Coarse tune & fine tune	Module-slide-based placement	global routing with local fine tune
PR strategy	CA: CR-driven Others & total: Area- & Aspect ratio-driven	Interactive	Symmetry, Area, Aspect Ratio	mixed: custom & automated	Template PR & fine tune	Mixed methods & interactive	Route net-by-net in a given order
PR algorithm	knowledge-based w/ templates	Interactive w/ templates	Defer[87] & RSMT[3]	mixed, interactive	B-tree, SA[5]	GASA[6], STB[7],CB[8]	SA[5], SBICR[9]
Process independent	N[2]	Y	unknown	Platform independent[4]	Y	Y	Y
Size netlist	User input or CIRCLES	User input or NeoCircuit	User input	User input	User input	User input	User input or IDAC

Note:
1: refer to the features besides covering the layout rule of thumb to guarantee the layout quality.
2: now it's process dependent. But in principle it can be process independent, and in the future it will.
3: Rectilinear Steiner Minimal Tree.
4: platform independent but no process information mentioned.
5: Simulated Annealing.
6: Genetic Algorithm & Simulated Annealing.
7: Steiner-Tree-Based routing.
8: Compaction-Based routing.
9: Scanline based incremental channel router.

As shown above, our serial-style cost function differs from all the others that use linear combination or the like. The serial style distinguishing CA and NCA with different figure-of-merit-oriented approaches really reflects what analog designers do in their job, guaranteeing the layout quality. In contrast, although all the prior work has many features

in both MGs and PR methods, not all basic analog layout rules of thumb are covered in their generated layouts (some absence is quite obvious, such as the lack of uniformity of current flow directions for current mirror structures, sometimes even no common centroid layout …). Moreover, the prevailing SA methods are not deterministic, causing some uncertainty.

CHAPTER 6. Future Work

As our proposed platform has already achieved a "SPEC in GDS out" flow for certain types of op amps, there are still some problems left to be solved or improved so far. Once they are solved or improved, based on our proposed principle and this open system, more applications will get supported. All these call for a total overview for system extensibility and future work in order to get the big picture clear.

6.1 To Enrich the library

Up to now, op amps of two topologies in two types (N-type & P-type) have already got support in CIRCLES platform based on our proposed flow. Accordingly, module library and PR template & engine library have already come into being, but still don't have enough components to cover many kinds of usual applications, such as the capacitor pair of the module library and the PR template for folded-cascode op amp.

However, in our approach, module generators, PR templates and engines all come from the available classical analog layouts that are already silicon-proven under the guide of knowledge-based algorithms. As there are a huge literature for these analog layouts that provides plenty of prototypes, enriching the library is just a time issue. In this process, building module generators with maximum flexibility and keeping the program process-independent are the two critical points.

As for the circuit synthesis, the design method can also be abstracted from analog design ideas. But more topologies will cause the decision-making issue within explored design space a bit complicated, which will be discussed in Section 5.2.

6.2 Automatic Topology Selection

Although the issue of topology selection in principle can be solved by our knowledge-based approach, the cost in speed is unavoidable as more and more topologies provide the similar design space, making the overlap issue quite complicated. For design assistant, sometimes it's unacceptable or even fatal. Essentially, it's a tradeoff between preciseness of topology selection and speed.

So far, the prevailing optimization-based methods also have this problem. Although there are some approaches trying to integrate the topology selection into circuit sizing and accelerate the solving by using advanced algorithms, the time-consuming issue still bothers. As a result, both knowledge-based and optimization-based approaches support manual intervention or direct manual input of sized netlist. But comparatively speaking, knowledge-based approach in the future can show more flexibility in selection, but requiring more detailed branches to analyze the usage description.

As now the emphasis of ADA is on automatic layout generation, in our future work, improving topology selection will be relatively low in priority list compared with enriching layout libraries. It's quite obvious especially before a lot of topologies can get support on our platform.

6.3 Interpretation of Usage Description

As mentioned above, this issue actually is part of the requirements posed by topology selection. With detailed knowledge, usage description can tell user's design inclination to the program, greatly simplifying the circuit synthesis flow. Also, it will equivalently set

some weights for certain figures of merits, quite helpful for PR engines to get a proper evaluation function.

For example, suppose if usage description of an op amp is for switched capacitor filter (SCF) application. As SCF will use capacitor pairs and are quite sensitive to parasitic capacitance, the program will pay much attention on the input parasitic capacitance. This feature, as interpreted by our knowledge-based algorithms, will show its inclination in both topology selection and the following circuit sizing. This information will even be passed to module generator and PR engines with possible change and care for critical nodes' parasitic capacitance.

In essence, the usage description and corresponding interpretation, is an advantage for knowledge-based methods that can avoid heuristic input or interactive manual guide in pure optimization-based methods. With build-in knowledge, user input can be simplified with implicit ideas still being well understood.

6.4 To Be Process Independent

As a design assistant or any EDA software, being process independent is obviously one of the targets. However, since different technologies (esp. from different foundries) may differ in some details and new rules come out as technology advances into smaller feature sizes, getting really process independent is not an easy task. Even those available tools or platforms (claiming that they are process independent as shown in Table 5) may still encounter some problems or need manual checks after importing different technologies.

For our case, currently the programs are not guaranteed to be process independent. However, as our program engines are all parameterized, being process independent is

theoretically not a difficult task. However, in reality, as technology file doesn't include all the process information and constraints, we have to resort to rule files.

But in rule files, as they are written in a text form rather than a standard look-up table, translating these rules is quite complicated. And based on our UMC design kits from 180nm, 130nm to 90nm, rule files contain more and more special rules that can hardly be written in a standard look-up table but lots of "if then" clauses. So instead of realizing the fully automatic process migration, the feasible way is to make our platform independent of the basic rules such as spacing, sizing, and enclosure. For the rules or constraints peculiar to any specific process, our platform will try using available experience and relatively conservative settings to avoid these pitfalls as many as possible. To guarantee the quality, manual intervention is necessary.

Considering our UMC 130nm case and some possible cases in the future, we will update our platform to cover at least UMC 180nm and 90nm technologies besides the 130nm one, which is quite feasible according to our investigation. For other technologies, future efforts will be done if necessary.

CHAPTER 7. Conclusion

7.1 Our Contribution

In this paper, our consistent efforts on quick tools assisted analog design have been presented. The original idea comes from design reuse, and then goes into the reconfigurable usage. To further pursuing the flexibility, the idea of Analog Design Automation (ADA) has been proposed with the corresponding design flow and examples shown above.

Our contribution mainly focuses on the following points:

1) Template-based PR algorithms with module generators for flexible components achieve the compact layout in a simple & fast flow, which can suffice the usual applications that call for limited flexibility.
2) For applications with more automation and design freedom, application-specific design style under the guide of build-in knowledge-based algorithms is used for both circuit synthesis and layout synthesis in ADA design. It achieves a fast and reliable design flow while keeping the system open and ready to accept more potential flexibility, quite attractive as a design assistant.

The Novelty of our work includes:

1) Original cost/evaluation function in serial mode to reflect what analog designers really care and how they care:

   Unlike traditional cost function in linear combination with user-defined weights, our approach goes in the same way that analog designers consciously do: first thing first, then go on in priority order.

Exclude the possible sacrificing the first for the second, which is more than possible in linear combination (parallel) mode.

2) Further classification of each major figure of merit to stress the design emphasis

   This makes what designers care even clearer, reflecting the real design thread much better the traditional ones. It's also one of the bases for serial-mode cost function.

3) Critical Routing (CR) oriented PR design style for optimizing the layout of Critical Area (CA).

   This exactly reflects what designers care after module generators provide guaranteed module layout.

4) Powerful Module Generator (MG) that covers all the analog layout rules of thumb and provides all the related information for global routing. It also provides the re-shaping function for global placement.

   Powerful and reliable MGs are the basis for CR-oriented PR style for CA.

5) Reliability and Performance driven style for circuit synthesis rather than the latter alone

   This stresses on the reality of analog design in small feature sizes while traditional optimization-based ones get undermined, esp. for pre-layout simulation in small sizes.

In essence, the difference between the traditional optimization-based methods (with generalized algorithms) and our application-specific approach is the difference between the automatic digital design style and the manual analog one. The traditional solutions try to transplant the mature digital flow to analog domain while our approach tries to abstract analog design experience, habit and designers' implicit ideas into clear flows to build a process cut out for analog circuit its own.

7.2 The Prospect of ADA

As technology advances, analog design has to cover more and more concerns as all kinds of higher order effects get much bigger. The complexity of design target setting and design considerations pose a great challenge to both the manual design and the EDA tools, including the ADA tools. This challenge, as it may lead to more design efforts on manual work, it will also make analog design automation even more attractive since it's the bottleneck now for AMS design.

As the challenge comes with more complicated issues, knowledge-based application-specific ADA design flow will distinguish itself from others as it provides a fast and very flexible approach to absorb any novel design ideas in solving all the new problems. Its coverage can quickly widen since the continuous enrichment of the libraries for both modules and PR engines have lots of ready and silicon-proven templates (huge literature of analog circuits and layouts).

Moreover, with ever-increasing computing power, paralleled processing structure using application-specific approach can greatly accelerate the design space exploration and by using knowledge-based algorithms poor candidates can be quickly excluded to further simplify the total flow. All these accelerate the speed and shorten the time to market, very attractive as a design assistant for analog designers.

Appendix

How to Install and Use the platform?

First, open the installation package as shown in Fig.77.

Fig. 77 Files in installation package

Second, run the installer named "CIRCLES_installer". Usually we use root to do that. Of course, you have to choose a different directory as the destination directory for this installation, not the same as the source destination (as shown in Fig.78).

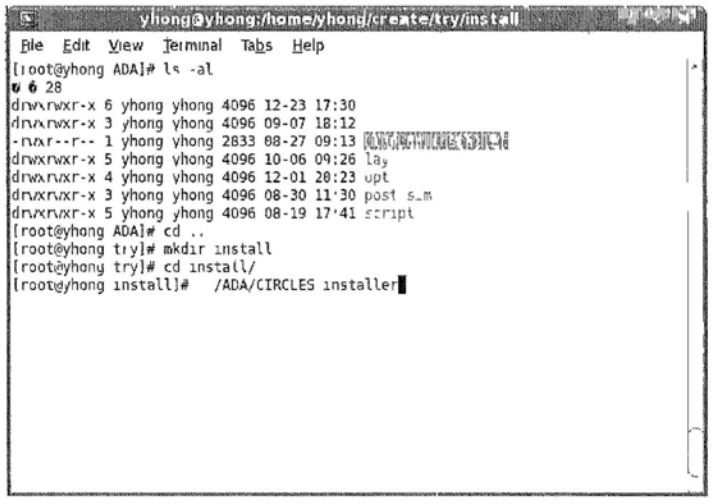


Fig. 78 Run the Installer

Then you can see the response as shown in Fig.79. After that, set your environment variables to include the following directory into your variable $PATH:

$INSTALL_PATH/lay

$INSTALL_PATH/opt

$INSTALL_PATH/post_sim

For different shells, the grammar may be different. For BASH that is used here, it's "$PATH=$directory:$PATH" where $directory is the new directory to be added in.

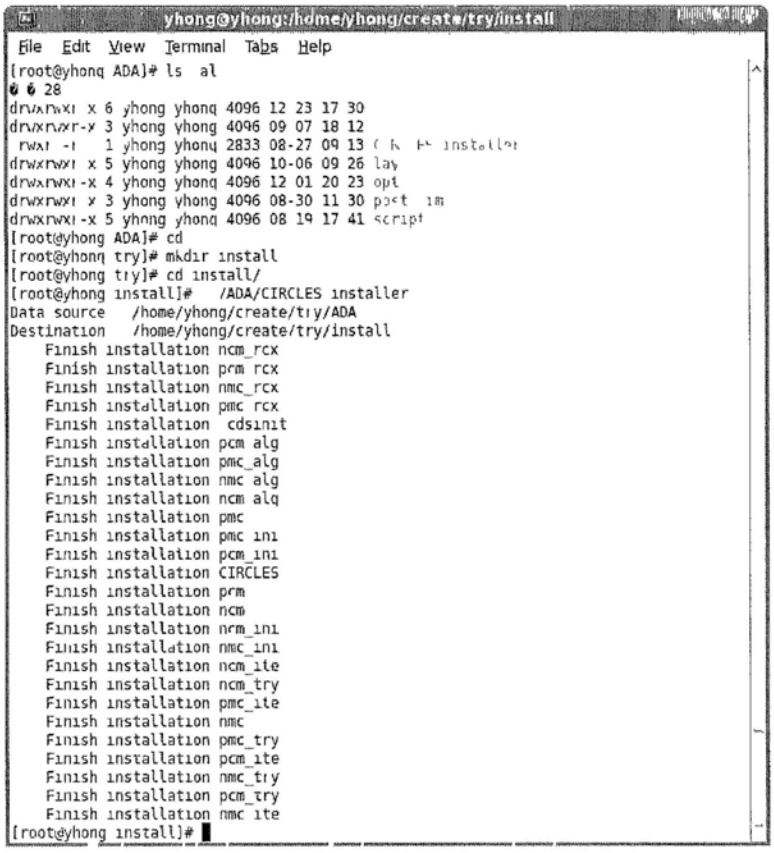


Fig. 79 Finish the installation

Then you can go to the place where you want to start CIRCLES, and simply click "CIRCLES" to get the GUI interface shown in Fig.34. Note that the GUI interface uses wxwidgets, which is a C++ library that lets developers create applications for Windows,

OS X, Linux and UNIX on 32-bit and 64-bit architectures as well as several mobile platforms including Windows Mobile, iPhone SDK and embedded GTK+. The related introduction can be found at http://www.wxwidgets.org/ and for Perl the related patch can be found at http://search.cpan.org/.

In the directory of $INSTALL_PATH/opt as shown in Fig.80, file "info" and "help" show the application note and each SPEC's typical range. File "format_try" is used to help user to manually set the pre-layout simulation netlist, which will be automatically popped up after user clicks "Edit Netlist" shown in Fig.45.

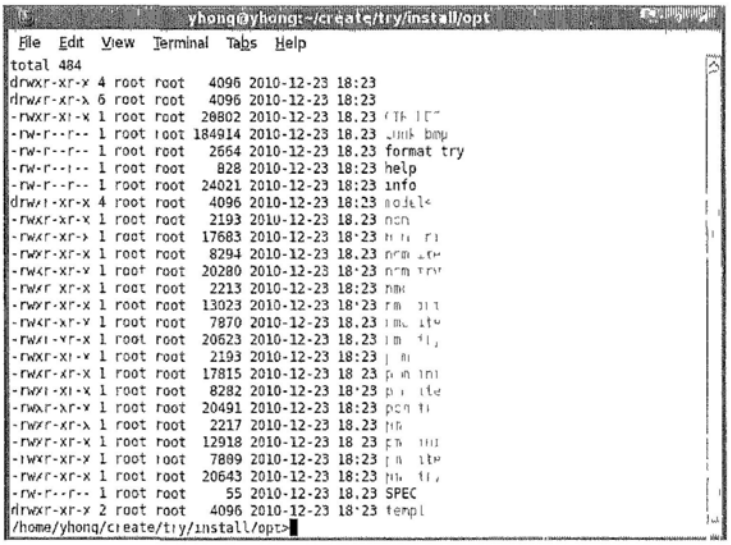


Fig. 80 Files in $INSTALL_PATH/opt folder

As for using it, first enter the SPECs and choose the topology, which is already shown in Fig.37. Then click "Generate SPEC Form". After that, click "Run Pre-Sim" button to get Fig.45. If you want to manually input the netlist or change the generated netlist, click "Edit Netlist" as mentioned above. Then click "Generate Layout and Run PV" to get the layout and physical verification report. If you want to do post-layout simulation, choose the extraction type: "C" or "RC", then click "Run Post-Sim" to see the results.

REFERENCES

[1] G.G.E.Gielen, R.A. Rutenbar, "Computer-Aided Design of Analog and Mixed-Signal Integrated Circuits", *Proc. of IEEE*, vol. 88, pp. 1825-1854, Dec 2000.

[2] E.Yilmaz, G.Dundar, "A NEW PERFORMANCE ORIENTED MODULE GENERATOR", *MIXDES 2006*, Page(s): 202 – 206.

[3] Xu Yang, L.T. Pileggi, S.P.Boyd, "ORACLE: Optimization with Recourse of Analog Circuits including Layout Extraction", *Proc. of DAC* 2004, pp.151-154.

[4] Xu Yang, C.Boone, L.T.Pileggi, "Metal-Mask Configurable RF Front-End Circuits", *IEEE J. Solid-State Circuits*, vol.39, Aug 2004, pp.1347-1351.

[5] C. Labrecque, "Near-Term Industrial Perspective of Analog CAD", *Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD), 2006,* pp. 456 – 457.

[6] B. Martin, "Automation Comes to Analog", *IEEE spectrum*, vol. 38, issue. 6, pp.70 – 75, 2001.

[7] P.R.Wilson, R.Wilcock, "Yield improvement using configurable analogue transistors", *Electronics Letters*, vol. 44, Sept. 2008, pp.1132-1134.

[8] K.Gulati, H-S Lee, "A Low-Power Reconfigurable Analog-to-Digital Converter", *IEEE J. Solid-State Circuits*, vol.36, Dec 2001, pp.1900-1911.

[9] R.Want, "RFID explained: A primer on radio frequency identification technologies", *Synthesis Lectures on Mobile and Pervasive Computing*, vol.1, Jan. 2006, pp.1-94

[10] J.Landt, "The History of RFID", *IEEE Potentials*, vol.24, no.4, Oct.-Nov. 2005, pp.8-11.

[11] U.Karthaus, M.Fischer, "Fully Integrated Passive UHF RFID Transponder IC with 16.7-μW Minimum RF Input Power", *IEEE J. Solid-State Circuits*, vol.38, Oct. 2003, pp.1602-1608.

[12] H. Kitayoshi and K. Sawaya, "Development of a passive RFID-tag with 10-m reading distance under RCR STD-1 specification," in *Proc. ISAP'04*, Aug. 2004, pp. 969–972.

[13] P. Harrop, "Near field UHF vs. HF for item level tagging", IDTechEx article, available at http://www.eurotag.org/?Articles_and_Publications

[14] D. Desmons, "UHF Gen2 for item-level tagging", presentation at RFID World 2006, available at www.impinj.com/files/Impinj_ILT_RFID_World.pdf

[15] "UHF Gen 2 for Item-level Tagging" Impinj RFID technology series paper, available at http://www.impinj.com/files/MR_GP_ED_00003_ILT.pdf

[16] C. Ajluni, "Item-level RFID takes off", RF Design magazine, Sept. 2006

[17] "Item-level visibility in the pharmaceutical supply chain: a comparison of HF and UHF RFID technologies", white paper by Philips, TAGSYS, and Texas Instruments, available at http://www.tagsysrfid.com/modules/tagsys/upload/news/TAGSYSTI-Philips-White-Paper.pdf

[18] Gen 2 Tag Clock Rate — What You Need to Know, http://www.impinj.com.

[19] R.A.Rutenbar, "Design Automation for Analog: The Next Generation of Tool Challenges", *Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD), 2006,* pp. 458 – 460.

[20] A.H.Fung, D.J.Chen, Y.N.Lai, B.J.Sheu, "Knowledge-Based Analog Circuit Synthesis with Flexible Architecture", *Proc. IEEE Int. Conf. Computer Design: VLSI in Computers and Processors, ICCD 1988*, pp.48~51.

[21] R.Harjani, R.Rutenbar, and L.R.Carley, "OASYS: A framework for analog circuit synthesis", *IEEE Trans. Computer-Aided Design*, vol. 8, pp.1247-1265, Dec, 1989.

[22] F.El-Turky and E.Perry, "BLADES: An artificial intelligence approach to analog circuit design", *IEEE Trans. Computer-Aided Design*, vol.8, pp.680-691, June 1989.

[23] H.Koh, C.Séquin, and P.Gray, "OPASYN: A compiler for CMOS operational amplifiers", *IEEE Trans. Computer-Aided Design*, vol.9, pp.113-125, Feb.1990.

[24] P. Veselinovic *et al.*, "A flexible topology selection program as part of an analog synthesis system," in *Proc. IEEE Eur. Design Test Conf. (ED&TC)*, 1995, pp. 119–123.

[25] R. Harjani and J. Shao, "Feasibility and performance region modeling of analog and digital circuits," *Kluwer Int. J. Analog Integrated Circuits Signal Processing*, vol. 10, pp. 23–43, Jan. 1996.

[26] P. Maulik, L. R. Carley, and R. Rutenbar, "Simultaneous topology selection and sizing of cell-level analog circuits," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 401–412, Apr. 1995.

[27] Z. Ning *et al.*, "SEAS: A simulated evolution approach for analog circuit synthesis," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, 1991, pp. 5.2.1–5.2.4.

[28] W. Kruiskamp and D. Leenaerts, "DARWIN: CMOS opamp synthesis by means of a genetic algorithm," in *Proc. ACM/IEEE Design Automation Conf. (DAC)*, 1995, pp. 550–553.

[29] L. R. Carley, G. Gielen, R. Rutenbar, and W. Sansen, "Synthesis tools for mixed-signal ICs: Progress on frontend and backend strategies," in *Proc. ACM/IEEE Design Automation Conf. (DAC)*, 1996, pp. 298–303.

[30] G. Gielen and W. Sansen, *Symbolic Analysis for Automated Design of Analog Integrated Circuits*. Norwell, MA: Kluwer, 1991.

[31] M. Degrauwe *et al.*, "IDAC: An interactive design tool for analog CMOS circuits," *IEEE J. Solid-State Circuits*, vol. 22, pp.1106–1115, Dec. 1987.

[32] R. Harjani, R. Rutenbar, and L. R. Carley, "OASYS: A framework for analog circuit synthesis," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 1247–1265, Dec. 1989.

[33] G. Beenker, J. Conway, G. Schrooten, and A. Slenter, "Analog CAD for consumer ICs," in *Analog Circuit Design*, J. Huijsing, R. van der Plassche, and W. Sansen, Eds. Norwell, MA: Kluwer, 1993, pp.347–367.

[34] N. Horta, J. Franca, and C. Leme, "Automated high level synthesis of data conversion systems," in *Analogue–Digital ASICs—Circuit Techniques, Design Tools and Applications*, Soin, Maloberti, and Franca, Eds. Stevenage, U. K.: Peregrinus, 1991.

[35] J. Vital and J. Franca, "Synthesis of high-speed A/D converter architectures with flexible functional simulation capabilities," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 1992, pp. 2156–2159.

[36] C. Toumazou and C. Makris, "Analog IC design automation—I: Automated circuit generation: New concepts and methods," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 218–238, Feb. 1995.

[37] C. Makris and C. Toumazou, "Analog IC design automation—II: Automated circuit correction by qualitative reasoning," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 239–254, Feb. 1995.

[38] B. Sheu, A. Fung, and Y.-N. Lai, "A knowledge-based approach to analog IC design," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 256–258, Feb. 1988.

[39] R. K. Brayton, G. D. Hachtel, and A. Sangiovanni-Vincentelli, "A survey of optimization techniques for integrated-circuit design," *Proc. IEEE*, vol. 69, pp. 1334–1362, Oct. 1981.

[40] P. E. Gill,W. Murray,M. A. Saunders, and M. H. Wright, "User's guide for NPSOL (Version 4.0): A FORTRAN package for nonlinear programming," Operations Res. Dept., Stanford Univ., Standford, CA, Tech. Rept. SOL 86-2, Jan. 1986.

[41] B. A. Murtagh and M. A. Saunders, "Minos 5.4 user's guide," Systems Optimization Lab., Stanford Univ., Stanford, CA, Tech. Rept. SOL 83-20R, Dec. 1983.

[42] H. Chang *et al.*, "A top-down constraint-driven design methodology for analog integrated circuits," in *Proc. IEEE Custom Integrated Circuit Conf.*, 1992, pp. 8.4.1–8.4.6.

[43] C. Maulik and L. R. Carley, "High-performance analog module generation using nonlinear optimization," in *Proc. 4th Annu. IEEE Int. ASIC Conf. Exhibit*, 1991, pp. T13-5.1–T13-5.2.

[44] P. C. Maulik, M. J. Flynn, D. J. Allstot, and L. R. Carley, "Rapid redesign of analog standard cells using constrained optimization techniques," in *Proc. IEEE Custom Integrated Circuit Conf.*, 1992, pp. 8.1.1–8.1.3.

[45] *A. R. Conn, N. I M. Gould, and PH. L. Toint, LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A).* New York: Springer-Verlag, vol. 17, 1992.

[46] A. R. Conn *et al.*, "Optimization of custom CMOS circuits by transistor sizing," in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1996, pp. 174–180.

[47] P. Heikkilã, M. Valtonen, and K. Mannersalo, "CMOS op-amp dimensioning using multiphase optimization," in *Proc. IEEE Int. Symp. Circuits Systems*, 1988, pp. 167–170.

[48] F. Leyn, W. Daems, G. Gielen, and W. Sansen, "Analog circuit sizing with constraint programming modeling and minimax optimization," in *Proc. IEEE Int. Symp. Circuits Systems*, vol. 3, 1997, pp. 1500–1503.

[49] K. Madsen, O. Niedseln, H. Schjaer-Jakobsen, and H. Tharne, "Efficient minimax design of networks without using derivatives," *IEEE Trans. Microwave Theory Techn.*, vol. MTT-23, pp. 803–809, Oct. 1975.

[50] H. Onodera, H. Kanbara, and K. Tamaru, "Operational amplifier compilation with performance optimization," *IEEE J. Solid-State Circuits*, vol. 25, pp. 466–473, Apr. 1990.

[51] G. Jusuf, P. R. Gray, and A. Sangiovanni-Vincentelli, "CADICS: Cyclic analog-to-digital converter synthesis," in *Proc. IEEE Int. Conf. Computer-Aided Design*, 1990, pp. 286–289.

[52] R. Chadha *et al.*, "WATOPT: An optimizer for circuit applications," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pp. 472–479, May 1987.

[53] J. P. Harvey, M. I. Elmasry, and B. Leung, "STAIC: An interactive framework for synthesizing CMOS and BiCMOS analog circuits," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 1402–1417, Nov.1992.

[54] W. Nye, D. C. Riley, A. Sangiovanni-Vincentelli, and A. L. Tits, "DELIGHT.SPICE: An optimization-based system for the design of integrated circuits," *IEEE Trans. Computer-Aided Design*, vol. 7, pp.501–518, Apr. 1988.

[55] W. T. Nye, E. Polak, and A. Sangiovanni-Vincentelli, "DELIGHT: An optimization-based computer-aided design system," in *Proc. IEEE Int. Symp. Circuits and Systems*, 1981, pp. 851–855.

[56] J. Shyu and A. Sangiovanni-Vincentelli, "ECSTASY: A new environment for IC design optimization," in *Dig. Tech. Papers IEEE Int. Conf. Computer-Aided Design*, 1988, pp. 484–487.

[57] C. Xinghao and M. L. Bushnell, *Efficient Branch and Bound Search with Application to Computer-Aided Design*. Norwell, MA: Kluwer, 1996.

[58] P. J. M. van Laarhoven and E. H. L. Aarts, *Simulated Annealing: Theory and Applications*. Amsterdam, The Netherlands: Reidel, 1987.

[59] D. F. Wong, H. W. Leong, and C. L. Liu, *Simulated Annealing for VLSI design*. Norwell, MA: Kluwer, 1988.

[60] P. C. Maulik, L. R. Carley, and R. A. Rutenbar, "Integer programming based topology selection of cell-level analog circuits," *IEEE Trans.Computer-Aided Design*, vol. 14, pp. 401–412, Apr. 1995.

[61] E. S. Ochotta, R. A. Rutenbar, and L. R. Carley, "Synthesis of high-performance analog circuits in ASTRX/OBLX," *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 273–293, Mar. 1996.

[62] G.G.E.Gielen, H.C.C.Walscharts, W.M.C.Sansen, "Analog circuit design optimization based on symbolic simulation and simulated annealing," *IEEE J. Solid-State Circuits*, vol. 25, pp.707–713, June 1990.

[63] F. Medeiro, F. V. Fernández, R. Domínguez-Castro, and A. Rodríguez- Vázquez, "A statistical optimization-based approach for automated sizing of analog cells," in *Proc. 31st Annu. Design Automation Conf.*, 1994, pp. 594–597.

[64] H. Z. Yang, C. Z. Fan, H. Wang, and R. S. Liu, "Simulated annealing algorithm with multi-molecule: An approach to analog synthesis," in *Proc. 1996 European Design & Test Conf.*, 1996, pp. 571–575.

[65] J. Chávez, M. A. Aguirre, and A. Torralba, "Analog design optimization: A case study," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 3, 1993, pp.2083–2085.

[66] Y. Nesterov and A. Nemirovsky, *Interior-Point Polynomial Methods in Convex Programming*. Philadelphia, PA: SIAM, 1994, vol. 13, Studies in Applied Mathematics.

[67] S. J. Wright, *Primal–Dual Interior-Point Methods*. Philadelphia, PA: SIAM, 1997.

[68] M.del.M.Hershenson, S.P.Boyd, T.H.Lee, "Optimal Design of a CMOS Op-Amp via Geometric Programming", *IEEE Trans. Computer-Aided Design*, vol. 20, Jan. 2001, pp. 1-21.

[69] M.del.M.Hershenson, S.P.Boyd, T.H.Lee, "Automated Design of Folded-Cascode Op-Amps with Sensitivity Analysis", *IEEE Int. Conf. Electronics, Circuits and Systems*, vol. 1, 1998, pp. 121-124.

[70] M.del.M.Hershenson, "Design of pipeline analog-to-digital converters via geometric programming", *ACM/IEEE Int. Conf. Computer-Aided Design (ICAD),* 2002, pp. 317-324.

[71] L. Zhang, U. Kleine, and Y. Jiang, "An automated design tool for analog layouts," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 14, no. 8, pp. 881–894, Aug. 2006.

[72] L. Zhang, U. Kleine, R. Raut, and Y. Jiang, "ALADIN: A layout synthesis tool for analog integrated circuits," *Analog Integr. Circuits Signal Process,* vol. 46, no. 3, pp. 215–230, Mar. 2006.

[73] E. Yilmaz, G. Dundar, "New Analog Layout Generator for CMOS circuits", *European Conference on Circuit Theory and Design* (*ECCTD*), 2007, pp.36-39.

[74] J.M.Cohn, D.J.Garrod, R.A.Rutenbar, L.R.Carley, "KOAN/ANAGRAM II: new tools for device-level analog placement and routing", *IEEE J. Solid-State Circuits*, vol. 26, pp. 330–342, Apr. 1990.

[75] J. Rijmenants, J. B. Litsios, T. R. Schwarz, and M. G. R. Degrauwe, "ILAC: An automated layout tool for analog CMOS circuits," *IEEE J. Solid-State Circuits*, vol. 24, no. 4, pp. 417–425, Apr. 1989.

[76] N. Lourengo, M. Vianello, J. Guilherme, N. Horta, "LAYGEN – Automatic Layout Generation of Analog ICs from Hierarchical Template Descriptions", *Research in Microelectronics and Electronics 2006, Ph.D.* 2006, pp.213-216.

[77] M. Dessouky, M.-M. Louerat, J. Porte, "Layout-oriented synthesis of high performance analog circuits", Proc. *Design, Automation and Test in Europe Conference and Exhibition, 2000*, pp.53-57.

[78] Yu-Ming Yang, I.H-R.Jiang, "Analog Placement and Global Routing Considering Wiring Symmetry", 11th International Symposium on Quality Electronic Design (ISQED), 2010, pp.618-623.

[79] E. Yilmaz, G. Dundar, "Analog Layout Generator for CMOS circuits", *IEEE trans. Computer-Aided Design*, vol. 28, Jan. 2009, pp. 32-45.

[80] M. Strasser, M. Eick, H. Grab, U. Schlichtmann, F.M. Johannes, "Deterministic Analog Circuit Placement using Hierarchically Bounded Enumeration and Enhanced Shape Functions", *ACM/IEEE International Conference on Computer-Aided Design* (*ICCAD*), 2008, pp. 306 – 313.

[81] S. Nakatake, "Structured Placement with Topological Regularity Evaluation", *Asia and South Pacific Design Automation Conference* (*ASP-DAC*), 2007, pp.215-220.

[82] S. Nakatake, K. Fujiyoshi, H. Murata, Y.Kajitani, "Module packing based on the BSG-structure and IC layout applications", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1998, pp.519-530.

[83] S. Director, W. Maly, and A. Strojwas, *VLSI Design for Manufacturing: Yield Enhancement*. Norwell, MA: Kluwer, 1990.

[84] T. Mukherjee, L. R. Carley, and R. Rutenbar, "Synthesis of manufacturable analog circuits," in *Proc. ACM/IEEE Int. Conf. Computer-Aided Design (ICCAD)*, Nov. 1995, pp. 586–593.

[85] G. Debyser and G. Gielen, "Efficient analog circuit synthesis with simultaneous yield and robustness optimization," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design (ICCAD)*, Nov.1998, pp. 308–311.

[86] Po-Hung Lin, Ho-Che Yu, Tian-Hau Tsai, Shyh-Chang Lin, "A Matching-based Placement and Routing System for Analog Design", *International Symposium on VLSI Design, Automation and Test* (*VLSI-DAT*), 2007, pp.1-4.

[87] J.Z.Yan, C.Chu, "Defer: Deferred decision making enabled fixed-online floorplanner", *45th ACM/IEEE Design Automation Conference* (*DAC*), 2008, pp.161~166.