AUTOMATED PLACEMENT OF A TRANSISTOR PAIR FOR ANALOGUE DESIGN

SARAVANAN BALAKRISHNAN

UNIVERSITI SAINS MALAYSIA

2012

AUTOMATED PLACEMENT OF A TRANSISTOR PAIR FOR ANALOGUE DESIGN

by

SARAVANAN BALAKRISHNAN

Thesis submitted in fulfilment of the requirements for the degree of Master of Science

UNIVERSITI SAINS MALAYSIA

May 2012

ACKNOWLEDGEMENT

It is with a deep sense of gratitude I wish to record here my most sincere thanks to my supervisor Dr. Mohd Tafir Mustaffa, School of Electrical and Electronic Engineering, Universiti Sains Malaysia, for the encouragement, personal guidance, assistance and valuable suggestions enabling me to steer my research work efficiently and effectively. His wide knowledge in the field of electronics engineering and logical approach for research work had been extremely useful for my research work and provided excellent basis for my thesis at the end. I am very grateful to my co-supervisor, Heng Chin Chuan, Physical Design Engineer of Intel Malaysia, for his detailed and constructive comments and support extended to me throughout my work. Most importantly my heartfelt sincere thanks should go to my beloved parents who have been instrumental in raising me up to the heights that I am in at present with their love, courage and support. I dedicate this thesis to them.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF PUBLICATIONS

Balakrishnan, S., Mustaffa, M.T. and Heng Chin Chuan, (2011) "Analog layout design optimization and verification for SoC". 2011 IEEE Regional Symposium on Micro and Nanoelectronics (RSM), 28-30th September 2011, Sabah, Malaysia.

PEMADANAN AUTOMASI TRANSISTOR BERPASANGAN UNTUK REKABENTUK ANALOG

ABSTRAK

Pengoperasian litar-litar analog terjejas oleh keadaan sekeliling seperti tahap-tahap bunyi bising, cerun terma litar, dan kesan-kesan negatif yang terhasil daripada faktor kerintangan dan kapasitif. Disebabkan tiada pendekatan-pendekatan yang berkesan untuk menangani kekangan analog seperti yang disebut di atas, tumpuan membangunkan aturcara rekabentuk litar bersepadu diberikan secara meluas ke arah litar-litar digital. Tujuan penyelidikan ini adalah untuk memperkenalkan satu kaedah lengkap untuk pemadanan transistor berpasangan rekabentuk analog menggunakan konsep sel dan susunan sel berteraskan penghijrahan dan penggunaan semula. Keseluruhan kaedah ini mengandungi komponen-komponen seperti Pembentuk Sel Piawai untuk menghasilkan sel piawai bagi litar analog, Pembentuk Pemadanan untuk tujuan pemadanan transistor secara teras-dasar *(common-centroid)*, dan Pengelaluan Automasi untuk pengelaluan global *(global routing)*. Kaedah ini diautomasikan oleh paparan penggunaan bergrafik *(GUI)* untuk menghasilkan bentangan analog yang berfungsi penuh dalam beberapa langkah mudah dalam masa yang singkat. Penyelidikan ini mampu menghasilkan satu bentangan yang boleh ditatarajah seperti satu set blok-blok binaan yang memenuhi kesemua spesifikasi rekabentuk analog. Dengan membandingkan segala pendekatan berbeza yang telah dibincangkan dan dianalisa sebelum penyelidikan ini, aliran rekabentuk baru untuk automasi bentangan analog dibina berlandaskan kekangan pemadanan transistor.

AUTOMATED PLACEMENT OF A TRANSISTOR PAIR FOR ANALOGUE DESIGN

ABSTRACT

The performances of analogue circuits are affected by surrounding parameters such as levels of noise, thermal gradients of a circuit, and parasitic effects from both resistive and capacitive part. As there are no effective approaches to handle these analogue constraints as mentioned above, the focuses to develop IC design tools are bended towards digital circuits. The purpose of this research is to introduce a complete methodology for transistor pair placement for analogue layout using a concept of cells and arrays based on migration and reuse. The entire process consists of Standard Cell Generation to produce standard cell for analogue circuits, Matching Generator with array alignment to generate transistor matching of common-centroid arrangement, and Auto Routing for global routing. The methodology is translated into automation by a graphical user interface to render a fully functional layout designs in a few steps and fraction of time. This research describes such a system in obtaining a layout that can be configured like a set of building blocks that meets all design specifications. In comparison to all the different approaches that have been discussed and analysed prior to this research, a new design flow for analogue layout combined with automation is constructed by considering transistor matching as a constraint.

CHAPTER 1

INTRODUCTION

1.0 Introduction

In recent times, integrated circuits (IC) have been carved into almost all of consumer-related products stretching from media players to communication devices. ICs in common consumer-engaged applications have been very complex that it evolves exponentially. As the ever-demanding market for integrated circuits increases, market readiness becomes a grave issue. This is due to the modern technology that sprouting swiftly that the time taken to design has now become a key attention. As such, various computer tools have been introduced in designing process with the aim of reducing the design's cycle time. Introductions of computer aided design tools or popularly known as CAD tools, intensifies productivity in IC design area. IC designs have become so multifarious that it is now impracticable to conduct a circuit design without any assistance from CAD tools (Dastidar, April 2005). Usage of these tools has effectively lowered the time taken for layout designs (Dastidar, April 2005). Due to that, issues surrounding block placements and routing interconnections of the transistors are drastically reduced.

Though the importance of such programs in IC generations cannot be denied, CAD tools are only used widely in designing digital circuits. A traditional practice in circuit designing is by using reusable design libraries in which the functions are pre-configured. Another practice is by applying cells that are pre-characterised with specifications that have to be implemented in any design. These cells are referred as

standard cells. The standard cells are organised in a horizontal manner where, by using a CAD tools, the cells are placed and routed in a way to reduce the area of the IC. The ultimate aim of any IC production is to produce IC that is minimum in design size as much as possible. ICs with smaller print-size helps to avoid the possibility of a single chip intersecting any defect that appears on the silicon wafer during IC development process. Additionally, the smaller the integrated circuit, the larger the amount of chips can occupy in a single silicon wafer. Apart from the digital standard cell method mentioned previously, a replacement method is by designing the integrated circuits using what is referred as interconnected transistors, as the basis to the design. It is known as macro-based cells. The macro-based cells are not required to be organised in horizontal manner. Therefore, the amount of transistors that are interconnected in any unit of area for a macro-based cell design type is larger compared to a standard cell type. However, with a low reusability factor, the time essential to build the macro-based cells is very costly. Figure 1.1 and Figure 1.2 shows these design methodologies (Waller, 2009).

Current design tools personalised for only one type of the above methodologies. There is no design tools developed to produce combination of the two topologies discussed above. But a few developments have made a positive progress to pursue mixed approaches (Waller, 2009). The introductions of programs such as cell generator have tremendously decreased the restrictions on designing time for huge circuit structures like as programmable logic arrays. A common knowledge is that by using a computer aided design tools alone does not guarantees circuit functionality, even after all the necessary placements and routings. These two para-processes influence the timing constraints of circuit signals but current design tools discard the timing

complications during the layout process. For the circuit to meet its specifications, the circuit designer often has to make many modifications and adjustments. Design flow could be reduced only if design tools understand the timing constraints (Stevens, et al., Feb 2003).

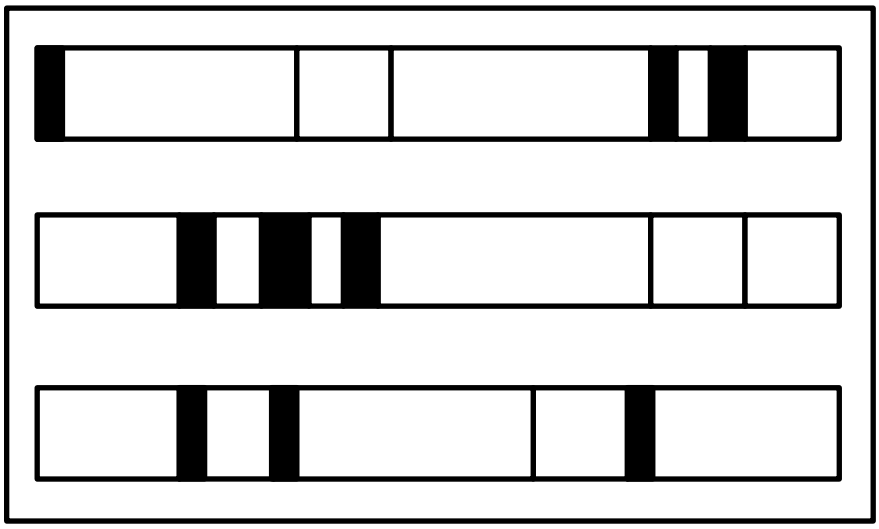

Figure 1.1: Standard cell design method

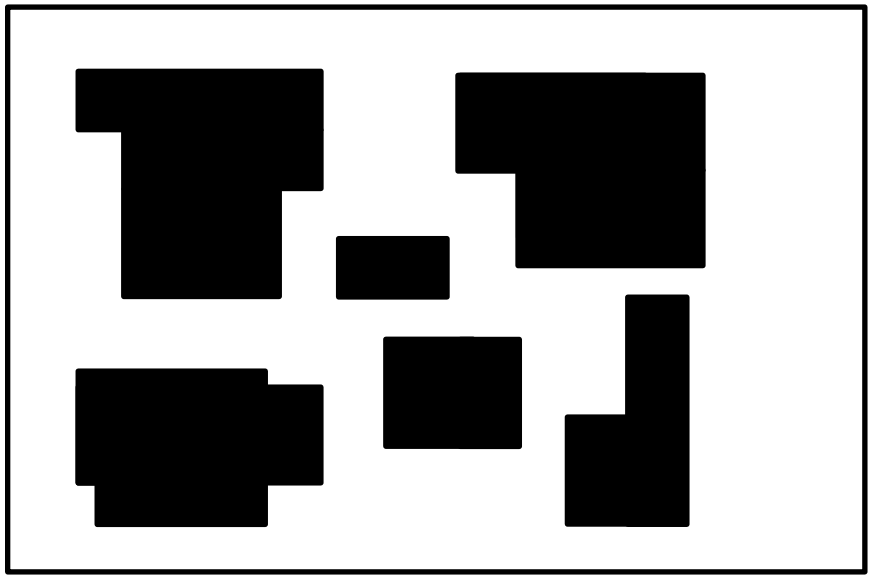

Figure 1.2: Macro-based cells design method

An IC design flow contains a few physical design phases, as shown in Figure 1.3 (Sadiq M. Sait, 1999). Until design target is achieved, back-cyclings and repetitions will be performed for each individual stage. To start the process, a structural representation, which actually recites routing of the physical components, will be used as the input of the physical design phase. The physical design process ends by generating output that contains the symmetrical information to accomplish IC fabrications. Physical components can be separated into three main layers, which are cell layer, device layer, and module layer. The bottommost level is the device layer, which defines components classified as physical devices. Diodes, rectifiers and transistors fall into this category. The cell layer, which is the middle level, describes a collection of devices with symmetrical data and has passed through the physical design phase at lower stages. Digital components such as logic gates and flip-flops, and analogue components such as differential amplifiers are member of cell level. The highest level is module layer, which contains a larger collection of devices. Associates of this layer have also passed through the physical design phase. Microprocessor, arithmetic-logic unit, and DAC/ADC unit can be grouped as module layer.

A netlist file is a textual representation of a circuit diagram or schematic (Wisconsin, 2011). The descriptions on structural of physical components and the interconnections are such information found in a netlist. It is generated by using logical design stage or circuitry design stage which comprises information regarding physical components and physical routings such as cell instances and network signals; referred as nets. In circuit level, the netlist is depicted in textual representation. The netlist also can appear in schematic representation in this level. A schematic representation contains

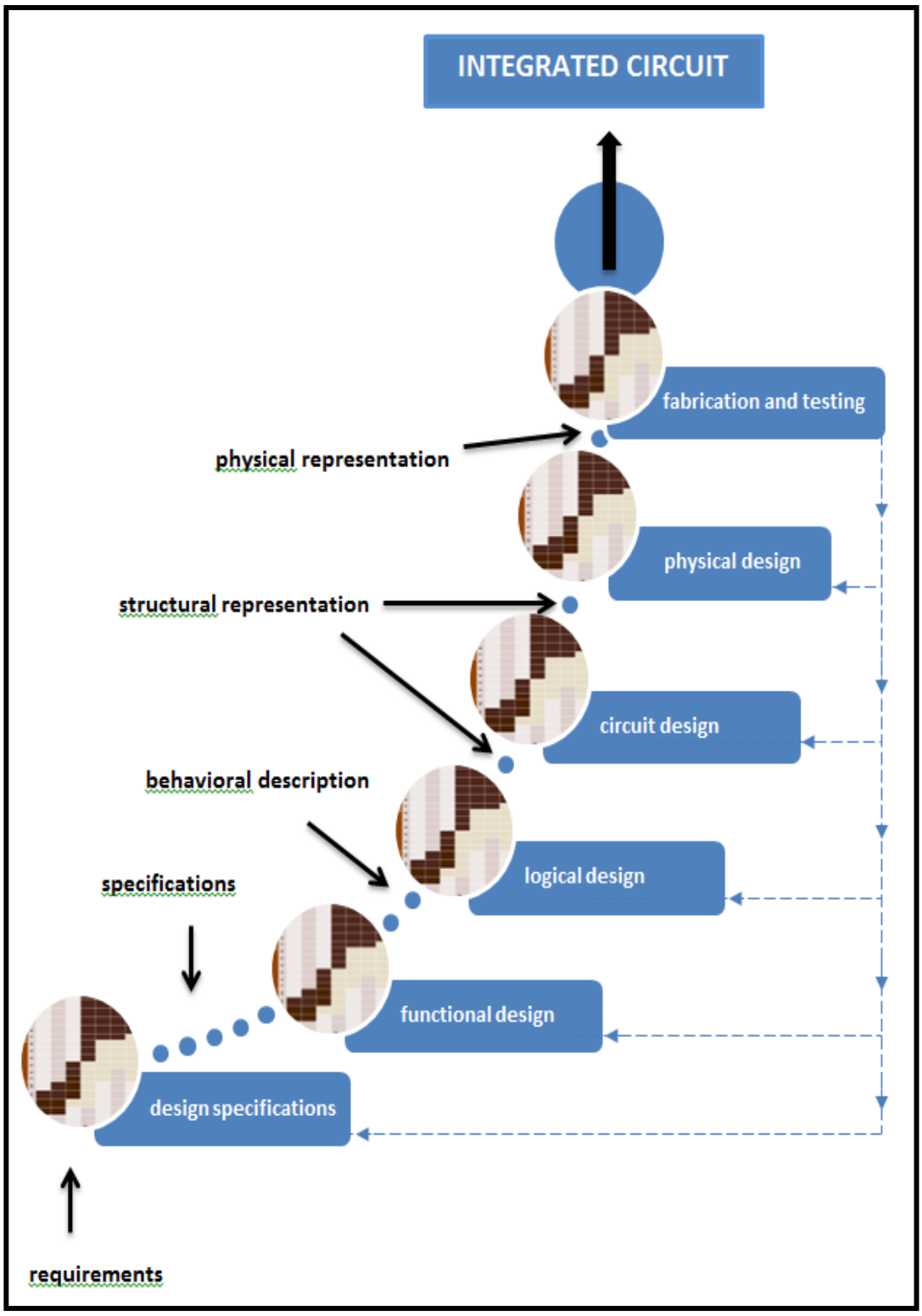

Figure 1.3: Phases in integrated circuit design.

symbols to signify physical components and lines to suggest the signals while in textual representations, physical components are portrait in text first. In schematic representation signal connects to a component's input and output pin.

In textual representations, the signals arranged in text format according to position starting from lower level of the hierarchy. Figure 1.4 and Figure 1.5 below shows a sample illustration of circuit level describing schematic representation and textual representation. On the other hand, Figure 1.6 shows a representation in the physical level. In the shown figure, the physical regions representation comprises wires that connect the components. These wires are constructed by depositing various materials such as polysilicon or copper on the silicon wafer. These materials are consecutively deposited on the silicon wafer during photolithography process.

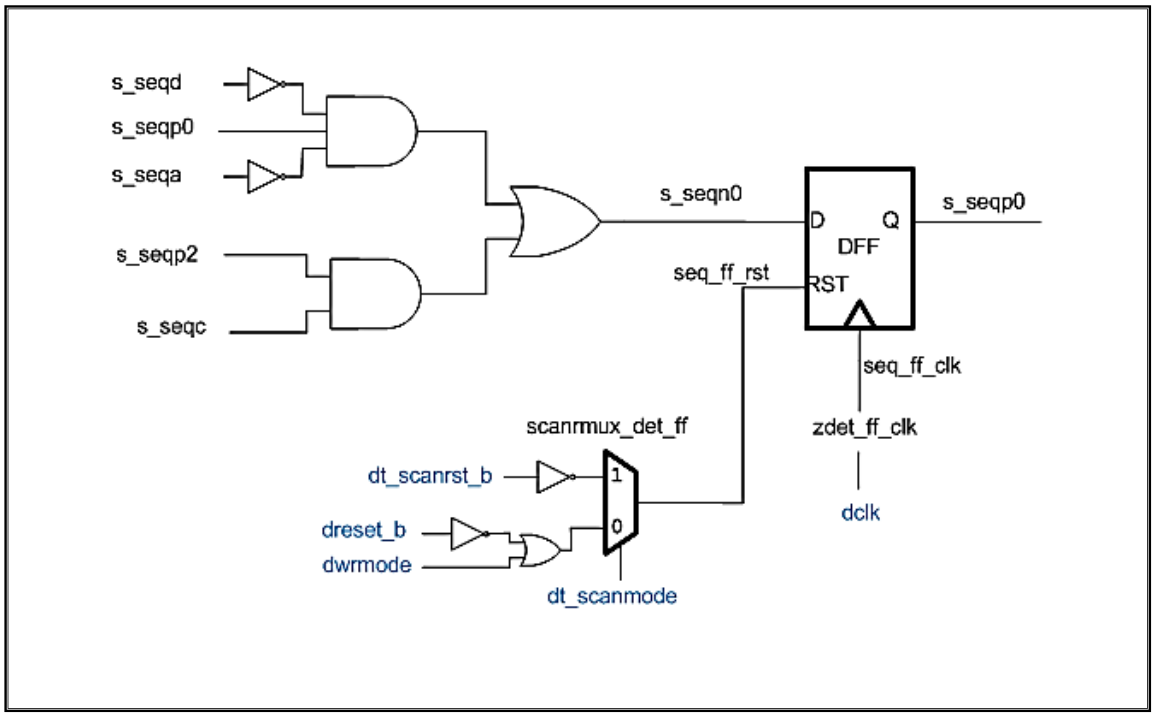

Figure 1.4: Schematic representations in circuit level

```
module toplevel(clock,reset);
  input clock;
  input reset;

  reg flop1;
  reg flop2;

  always @ (posedge reset or posedge clock)
    if (reset)
      begin
        flop1 <= 0;
        flop2 <= 1;
      end
    else
      begin
        flop1 <= flop2;
        flop2 <= flop1;
      end
endmodule
```

```
module Div20x (rst, clk, cet, cep, count, tc);
// TITLE 'Divide-by-20 Counter with enables'
// enable CEP is a clock enable only
// enable CET is a clock enable and
// enables the TC output
// a counter using the Verilog language

parameter size = 5;
parameter length = 20;

input rst; // These inputs/outputs represent
input clk; // connections to the module.
input cet;
input cep;

output [size-1:0] count;
output tc;

reg [size-1:0] count; // Signals assigned
                      // within an always
```

Figure 1.5: Textual representations in circuit level

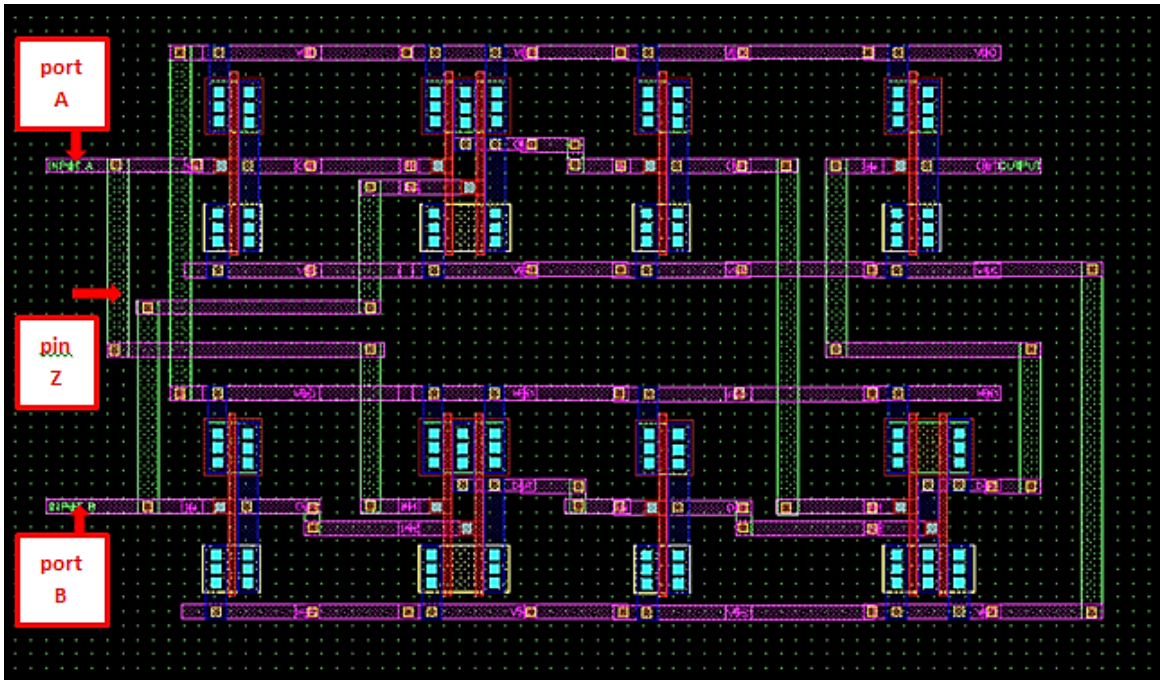

Figure 1.6: Representations in physical level

Photolithography is defined as a consequential process of transferring symmetrical shapes on a mask to the surface of a silicon wafer (Georgia, 2011). In short, it is an optical means for transferring patterns onto a substrate. Photolithography uses photoresist that can be spread out onto a substrate to expose with a desired pattern, and develop into a selectively placed coating for subsequent processing for the materials to be deposited; referred as layers. These layers can be categorised as device layers and route layers. Examples of these categories are diffusions and metals. Device layers used for physical device generations while route layers used for device interconnects. Route layers can cross each other without short-circuiting as they are insulated electrically. The current carrying ability of the layer is subjected to its electrical characteristics and the maximum inter-connection's length before the signal fades.

Metals like polysilicons have a very large resistance that it performs better for a short distances of routing connections. In comparison, metal such as copper has a very low resistance and is proper to use it for long distances of inter-connection. Previous technologies advancement limits circuit designers to single metal and single polysilicon layer but with recent technology achievements, circuit designers are able to construct routings with many metal layers. A functional design can only be produced if the design tool fully comprehends the characteristics of each interconnect layer material.

All these wires connect to a component through ports. A single pin of a circuit can consist of many physical ports. Referring to Figure 1.6, pin Z contains two ports, Port A and Port B. Routing layers are connected together with vias or contacts. Each layer is subjected to design rules. Design rules are guidelines in the fabrication technology. The design rules contain information that conforms to the physical constraints for all the layers. These constraints can be a requirement of the design itself,

such as reducing noise, or a requirement of the process used to produce the design (Agilent Technologies , 2011). A quick check can be performed to ensure preciseness to basic design requirements, such as minimum spacing and minimum width, or any custom check can be run using pre-written rules to ensure that a design meets manufacturing specifications (Agilent Technologies , 2011). In either case, designers can check all or few parts of the design. Whether a quick check or a custom check, the procedure is essentially the same; first define a design rule, then run the design check using the selected rule. Finally load the results and view for any errors to be found. As fabrication technology advances, design rules evolve regularly.

1.1 Problem Statement

However, automated computer-aided physical design tools are purposely-made only for digital circuits. Digital circuits process discrete binaries while analogue circuits, on the other hand, process actual signals which are in the form of continuously signals only. Thus, performances of analogue circuits are affected by surrounding parameters such as levels of noise, thermal gradients of a circuit, and parasitic effects from both resistive and capacitive part. As there are no effective approaches to handle these analogue constraints as mentioned previously, the focuses to develop IC design tools are bended towards digital circuits. This downside becomes the important bottleneck to focus on as whole systems are assimilated on a chip that combines both analogue and digital circuits. Analogue layout automation become even more important to achieve project requirements as the IC industry is venturing into complex designs with many

complicated design and manufacturing rules (APPENDIX A & B). The analogue layout portion of any IC is almost absolutely performed by hand using layout designers. This creates hurdles for layout designers to feed in all the rules and at the same time provide enhanced designs across all metrics. Thus, these designers face challenges to provide comparable results to those of an automated design of their digital counterpart (APPENDIX C). An automated solution for analogue layout design must be made to address a diverse set of criteria, of which includes timing, designing style, layout density, and integration issues. With this, a balance must be reached so that the right methodology of automation is made available to analogue layout designers to address these challenges (APPENDIX D). Furthermore, the methodology should be able to provide near-manual layout quality of results, but in a reduce segment of the time it consumes for the manual layout process.

1.2 Objectives

- To introduce an effective methodology for analogue layout automation that :
  - addresses design constraints (matching), design style, and integration issues of analogue layout.
  - provides alternative for complex analogue layout designs, which are with many complicated design and manufacturing rules.
  - increases productivity associated with analogue layout designs.
  - improves time-to-market of an analogue IC

1.3 Scope of work

- To introduce an automated methodology for layout placement, which can be used in analogue layout designs.
- To use the proposed methodology to reduce analogue matching constraints that exist in analogue layout design process during placement stage.
- To prove the working concept of the proposed methodology by developing related tools/programs.

1.4 Research Contributions

- Provides alternative solution for analogue layout design constraints (matching), design style, and other integration issues.
- Provides alternative method to address limitations on complex designs.
- Increases productivity related with analogue layout designs.
- Reduces time-to-market constraints of an analogue IC.
- Improves analogue design cycle by reducing time taken for designing.

1.5 Thesis Organisation

- Chapter 1 introduces to the research background, problem statement, objectives of this research, and research contributions.

- Chapter 2 is about reviewing other researches on analogue layout automation and design methodology that have been done in recent times, with comparison to this research. This chapter also revolves around phases of physical design involved in analogue circuit. This is to give brief idea on the current designing levels involves, compared to the proposed methodology on this research.

- Chapter 3 focusses on implementation of proposed automation and methodology which includes analogue standard cell concept and new placements approach of analogue layout while addressing the matching constraints involved. This chapter will focus on transistor matching which is one of analogue design constraints during placement that plays important factor for analogue circuit stability. This design constraint will be included as part on proposed methodology.

- Chapter 4 is on results and discussion and final conclusion on this research.

- Chapter 5 is on final conclusion future work of this research.

CHAPTER 2

LITERATURE REVIEW

2.0 Introduction

In this chapter, other researches on analogue layout automation and design methodology that have been done in recent times will be reviewed. This is to make a point on the differences and parallelity of this research compared to other similar researches than have been conducted in recent times. Then the discussion is followed by a detailed review of current physical design stages.

2.1 Recent Developments

2.1.1 Automatic Generation of Hierarchical Placement Rules for Analogue Integrated Circuits

This paper (Eick, et al., 2011) discussed on their research for hierarchical placement rules generation automation. By comparing previous papers such as (Charbon, et al., 1993) and (Sangiovanni-Vincentelli, 1993), the authors have decided that existing approaches can only generate flat constraints, i.e. between the devices of a circuit as hierarchical partitioning are not provided for hierarchical placements. The hierarchical placement can only be done using hierarchical placement rules. Few constraints can be removed from the hierarchy, which in return will be useful to generate improved circuit performances.

The proposed methodology requires three stages to produce the intended hierarchical placement rules. The methodology is shown in Figure 2.1. Initial stage confirms on building blocks and symmetry conditions. Information obtained from stage one will be used in stage two to form a graph that represents the constraints required by the circuit. Then this graph is used to regulate a circuit partitioning into hierarchical groups and to generate the related placement rules. This is done in final stage. These placement rules consider different constraints on proximity, matching, and symmetry in a comprehensive manner.

In comparison to other automation tools, this method has unique features that can distance itself from the rest from tools. Few approaches introduces a hierarchy concept for placement rules that delivers a prioritisation of placement rules, clustered circuit hierarchical and constraint information specific to cluster. A comprehensive hierarchical placement rules suitable to use as input for the placement algorithms. The tool too has an improved constraint generation for better circuit performance according to post-layout simulation compared to existing automation tool.

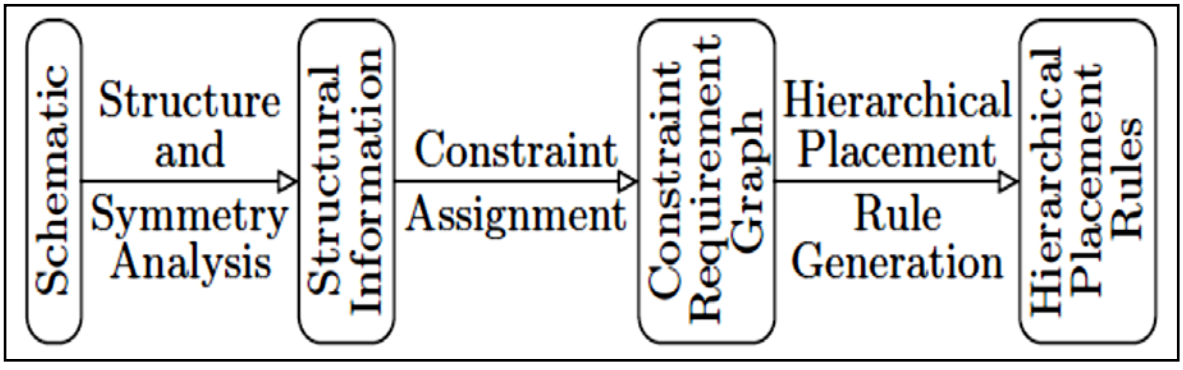

Figure 2.1: Design flow for automatic generation of hierarchical placement rules

(Charbon, et al., 1993)

In conclusion, this paper manages to present a complete method to generate hierarchical placement rules for analogue circuits automatically. Each group represented by hierarchical placement rules for a circuit as hierarchical groups and the placement constraints. Proximity and symmetry are among in the list of constraints managed by this automation tool. Referring to post-layout simulation these placement rules contributes to the circuit improvement and performances.

2.1.2 Analogue Placement and Global Routing Considering Wiring Symmetry

In this paper, the issue of wiring symmetry in analogue layout automation is highlighted (Yang & Jiang, April 2010). The research prioritises device and wiring symmetry, seeing the analogue layout is very much sensitive to the layout geometry. According to the paper, prior placement algorithms are not adequate to eliminate the wiring symmetry limitations. Hence, to maintain the device and wiring symmetry, the researcher proposed an algorithm for analogue placement and routings. Few prior works used stochastic algorithms to solve the problem but in this research, they adopted a deterministic placement approach. The paper mentions that symmetry limitations of device appears on each single device that are connected each other. Device folding technique is utilised to enhance the symmetry with the viable dimensions recorded into the shape curve. It is also stressed that lower signals that travels in analogue designs should not be mixed up with larger signals as the placement follows the signal flow from lower to larger signals.

The signal flow is actually prearranged and its priority determined by value. The layout is then generated by considering the priority values. The symmetry information is

extracted during placement while the global router specifies the symmetry constrained wires and route them first. This helps to conserve the wiring symmetry for each pair of symmetry-constrained devices. The produced results proves the mentioned algorithm can yield a favourable layout for the designers and can be used for further modifications. Two key benefits of this analogue automation approach are wiring symmetry and device folding mechanism. The wiring symmetry performed for routing among interconnected devices and not limited within symmetric devices but also for the routing among them and their connected devices.

For device folding, it separates large devices into small pieces and inserts the pieces into shape curve combination. This increases the flexibility and feasibility of the symmetry. Referring to Figure 2.2 will enable us to get a clearer understanding of the analogue placement and global routing automation. The exact automation starts with analogue placement. The placement is done by prioritising the device and wiring symmetry limitations. A SPICE file containing circuit information will be supplied as an input. On the other hand, the device and wiring symmetry limitations can be inserted into the SPICE code and hierarchical constraints tree is constructed accordingly. The analogue placement algorithm will record feasible dimensions of each device as a shape curve. Device and wiring symmetry constraints and device dimensions defines the placement priority. Any interconnects of symmetry module will be routed first and each interconnects will be created along routing process. This is to maintain the symmetry of the wiring.

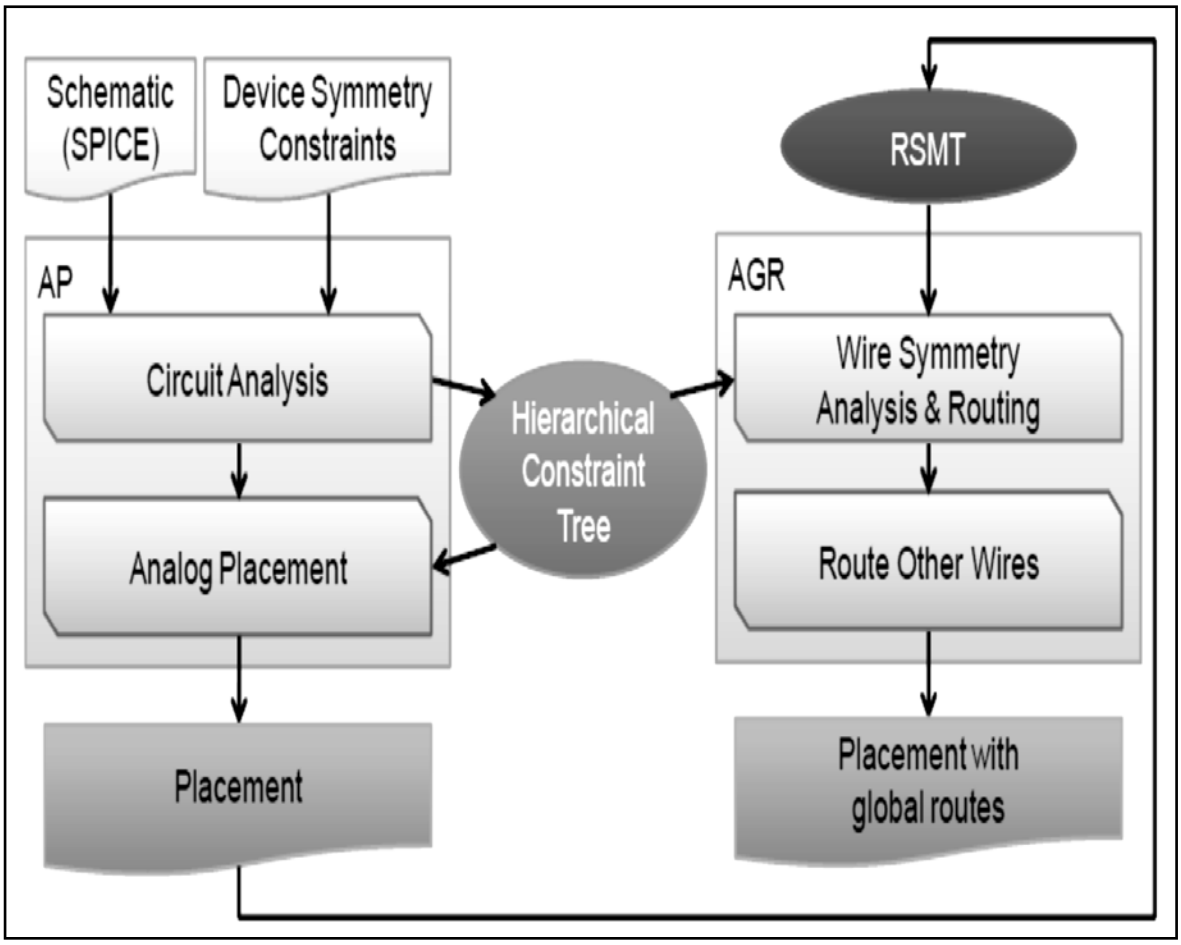

Figure 2.2: Analogue placement and routing automation design flow (Yang & Jiang, April 2010)

2.1.3 Automatic Analogue ICs Layout Generator (LAYGEN)

LAYGEN is an advanced analogue IC layout designing tool developed based on evolutionary computation techniques (Lourengo, et al., 2006). The concept is that the layout designer provides a high-level layout description to a template that will be used to generate analogue layouts. The paper introduces unique layout style to designers. Even with that, the layout style is very consistent with design guidelines. This is true even with some specifications or technological changes. The usage of parametric generators for simple cells is technology-dependent, thus limiting the re-use capability. But for complex cells, parametric generators are not effective in terms of design time and reusability. To overcome these restrictions, LAYGEN stores these design regularities in a layout meta-description that is independent of technology. This will provide the designer with a technological and specification to define some of the most commonly used cells. The complete design flow for LAYGEN is shown in Figure 2.3.

Expertise and experience of a layout designer is very important in dealing with analogue layout constraints such as device matching, parasitic, and thermal gradients. However this expertise and experience does not hover on process technology design details such as minimum poly length, width of contact or maximum length of interconnects. The templates are created based on designer expertise as inputs. These inputs are then used to guide LAYGEN to generate the desired layout without being bounded to process technology and specifications. The template based approach provides the layout designer ability to control the layout automation. This process reduces the solution space by decreasing the computational efforts required to achieve the optimal solution. Therefore it directly increases the desired layout quality.

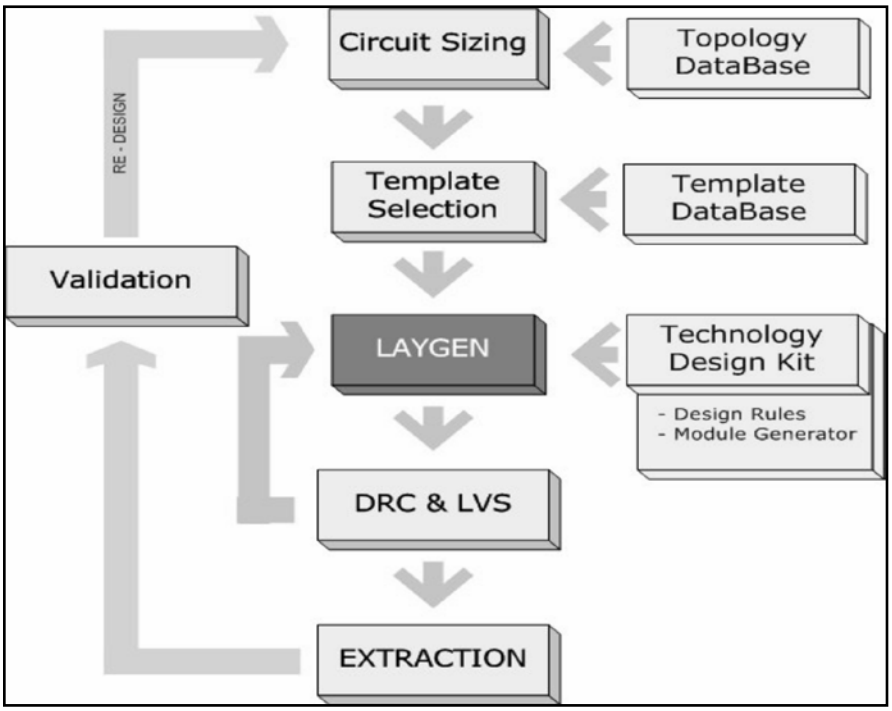

Figure 2.3: LAYGEN design flow (Lourengo, et al., 2006)

The layout generation of LAYGEN are conducted in common steps, which involves placement and followed by routings. The template topological relations are mapped into a non-slicing (F. Balasa, Feb 2004) data structure, enforcing the same topological relations on the desired layout. Then, LAYGEN chooses the best matching of modules that produces finest module placement in according to the topological relations that exist in the template. Apart from placement, the template also contains information on desired routing paths. The routing paths are rearranged to the positions of new contact points and later tuned to produce the desired layout. The information fetch from template that is used for placement is the comparative placement, size, and matching requirements of targeted devices. Modules can also be created in hierarchical way by using the templates. This gives the designer an opportunity to use the templates for simpler cells in the definition of more complex ones.

2.1.4 An Automated Design Tool for Analogue Layouts

Lihong Zhang, Ulrich Kleine, and Yingtao Jiang writes on an automated layout design tool, named as ALADIN. ALADIN (Lihong, et al., Aug 2006) is abbreviation for automatic layout design aid for analogue integrated circuits. It allows analogue circuit designers to apply their knowledge in circuit designing into the synthesis process to create quality layouts. The placement and the routing of modules of the layouts are performed automatically under the custom-made constraints for applications.

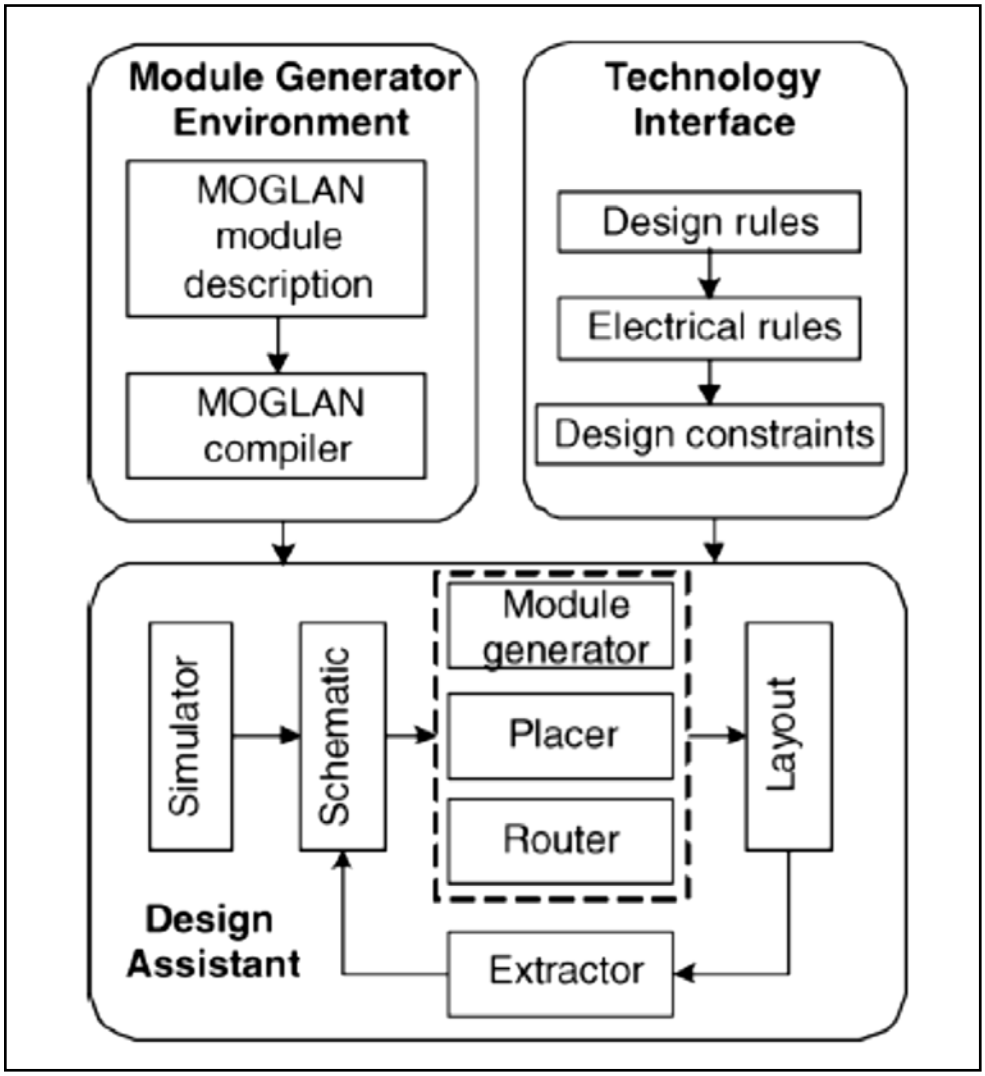

Figure 2.4: Block diagram of ALADIN (Lihong, et al., Aug 2006)

The block diagram of ALADIN is as shown in Figure 2.4. The whole system of ALADIN consists of three major components, which are Module Generator Environment, Design Assistant and Technology Interface. Module Generator Environment (Wolf, et al., 1998) allows circuit designers to write modules as complex as needed. Design Assistant provides graphical user interface to assist on optimising analogue circuits from schematic to layout. The application of Technology Interface (Zhang, et al., 2002) simplicities the input of description rules including the design rules, the electrical rules, and the special design constraints. The circuit-layout synthesis flow of this tool is shown in Figure 2.5.

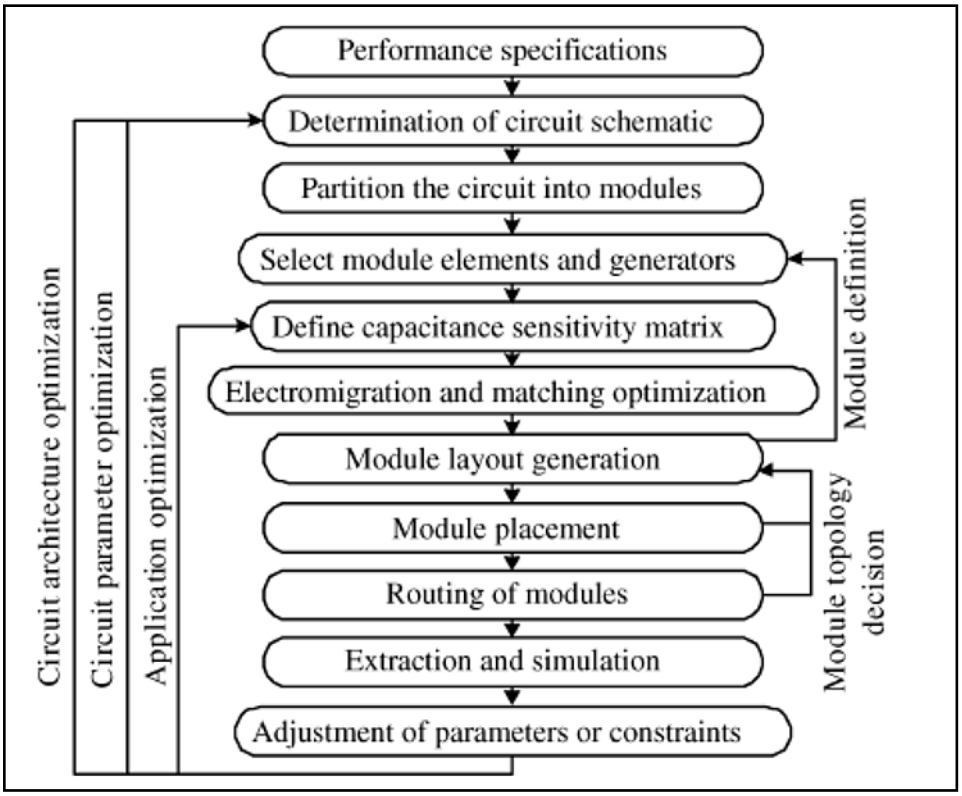

Figure 2.5: Circuit-layout synthesis flow of ALADIN (Lihong, et al., Aug 2006)

In ALADIN, a simulated circuit is partitioned into few modules. The parasitic capacitance and the electrical rules in the module layout are reflected during the module generation. The module selection and the layout generation will be repeated until all the circuit modules are defined. As for the placement of the modules, it is done concurrently with the global routing. An appropriate topology of each module is chosen from all possible alternatives during the process of placement and global routing. The alternatives are located in the design library. Detailed routing finalizes the layout synthesis flow with an embedded layout compaction capability. An extraction is performed after the layout generation.

The extracted parasitic elements are marked back automatically into the schematic. The whole circuit is then run in simulation with parasitic estimation, and the feedback optimisation loop starts. These feedback optimisations include three important techniques which are application optimisation, parameter optimisation and circuit architecture optimization. Application optimization redefines electrical constraints to update the detailed module generation and the routing process. The parameter optimisation changes circuit parameters and performs the layout generation for the impacted part of the circuit. Lastly, circuit architecture optimisation changes the structure of the circuit and, thus, undergoes a major update of the layout generation.

To summaries the performance of ALADIN, the microphotograph of the ALADIN final result of the rail-to-rail Op-amp fabricated using 0.8um n-well CMOS technology is shown in Figure 2.6. As a comparison, the microphotograph of a manually-drawn layout fabricated using the same technology is shown in Figure 2.7. One can see that the performance of the layout generated by ALADIN is comparable to that of the manual layout.

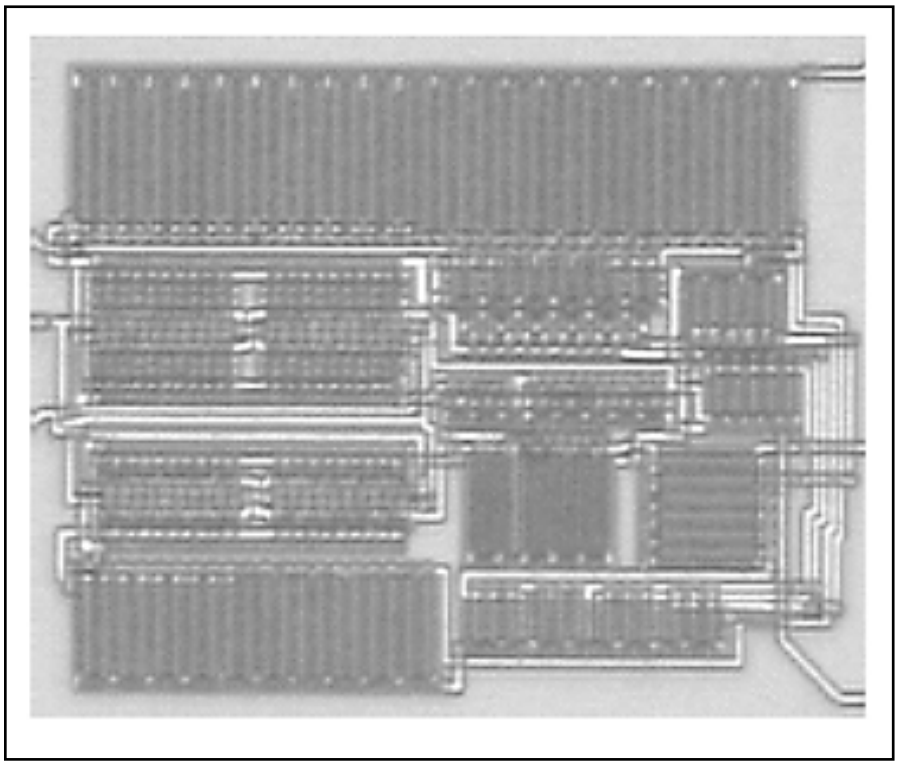

Figure 2.6: ALADIN layout (Lihong, et al., Aug 2006)

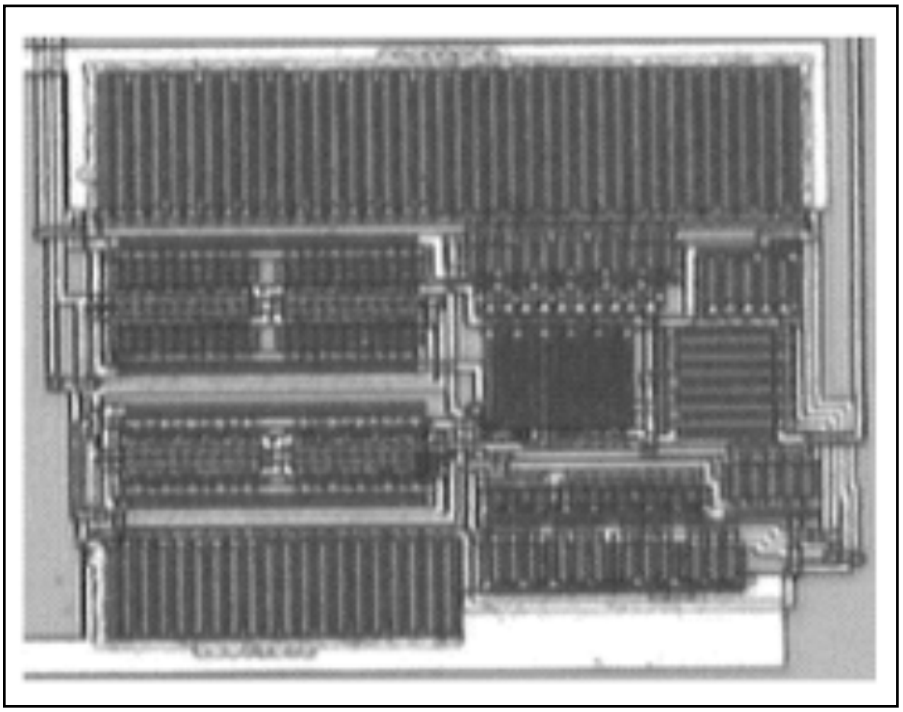

Figure 2.7: Manually-drawn layout (Lihong, et al., Aug 2006)

2.1.5 Automatic Device Layout Generation for Analogue Layout Retargeting

A new automatic design reuse methodology for analogue integrated circuits is presented through this research paper (Hartono, et al., 2005). It highlights a technique for automatic active device layout generation for analogue integrated circuits, which is combined together with a layout retargeting tool. The overall layout topology and layout symmetries are maintained by the use of a graph-based symbolic template. The device generator allows optimisation of active devices in terms of device parameters such as width, length, and finger variables. This is done through template modification. Referring to the result from their research, multiple high quality analogue circuits corresponding to different target specifications are synthesized in less than an hour. The layouts with different device sizes and structures are generated in less than a minute of testing CPU time. The automatic device layout generation permits the retargeting tool to vary the device sizes and fingers. This provides superior control over the layout parasitic effects that helps achieve more demanding design specifications. The use of design space exploration engine coupled with layout optimisation helps avoid costly design iterations and greatly reduces the manual efforts and solution time. The methodology and tools have successfully generated various highly optimised analogue layouts targeted at different specifications.

In this research, a netlist is obtained from an existing layout. This includes device parameters such as transistor widths, lengths, and number of fingers as variables. The netlist is optimised according to the design space exploration engine. It finds device geometry variables for each specification. By using the layout retargeting engine, target layouts are then generated. This is where transistor layouts are rebuilt through a series of