CAD AUTOMATION MODULE
BASED ON CELL MOVING ALGORITHM
FOR INCREMENTAL PLACEMENT TIMING OPTIMIZATION

by

KAN MEI WAR

Thesis submitted in fulfillment of the requirement
for the degree of
Master of Science

July 2012

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude and appreciation to my supervisor, Dr. Bakhtiar Affendi Bin Rosdi, for guiding me during my master's study and helping me in discovering an exciting field in the computer-aided design of electronic systems. This study would not be possible without his invaluable advices, suggestions and supports.

Special thanks to Chew Eik Wee, an Intel design engineer for his mentoring, support, knowledge sharing and consistent help to my work over the last two years. I also deeply thank to my friends and colleagues, especially to Wai Mun, Shabagran, Yit Siang, and Saw Beng for their best friendship, help, and support. They made my life in the USM challenging yet colorful and enjoyable. Many thanks also to the engineers in Intel Microelctronics (M) Sdn Bhd in Penang, Sze Wei and Swee Leen for creating a conducive learning and research environment.

I would also like to thank my parents and my family members for their unconditional love during my years in graduate school. Their care always provides the warmest support in my life and work, wherever I am.

Last but not least, I would like to offer my gratitude to Bee Leng, a manager in USAINS Holding Sdn Bhd, who always provides precious suggestions and consistent support during the last two years, to the completion of this research and thesis.

TABLES OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

LIST OF SYMBOLS

LIST OF TERMINOLOGIES

TCR	Total cells on right side of optimal position	79
TCL	Total cells on left side of optimal position	79
CCW	Critical cell's width	79
NRS	Next right branch solution	79
NLS	Next left branch solution	79
TMC	Total move cells of initial solution	79

MODUL AUTOMASI CAD BERDASARKAN ALGORITMA PERGERAKAN SEL UNTUK PENGOPTIMUMAN MASA PENEMPATAN TOKOKAN

ABSTRAK

Engineering Change Order (ECO) ialah satu proses untuk menangani perubahan logik dalam rekabentuk litar. Dalam era *deep sub-micron* (DSM), perubahan logik dalam rekabentuk litar adalah tidak dapat dielakkan. Perubahan dalam rekabentuk litar diperlukan untuk pelbagai sebab. Antara sebab-sebab adalah untuk memperbaiki fungsi litar, memenuhi keperluan pelanggan, atau mengoptimumkan persembahan litar seperti pengunaan kuasa. Penempatan tokokan yang mempunyai keupayaan untuk menangani peubahan logik dengan cekap dapat mengurangkan masa dan kos. Inilah sebab mengapa ECO merupakan salah satu peringkat yang mustahak dalam rekabentuk *Very Large Scale Integration* (VLSI). Tesis ini menghuraikan penempatan tokokan yang menggunakan teknik piawai sel bergerak untuk membaiki masa laluan rekabentuk. Fungsi penapisan sel ditambah masuk ke dalam penempatan tokokan untuk meningkatkan peluang memperbaiki masa rekabentuk. Fungsi ini menentukan sel-sel yang mana perlu dipindahkan untuk mencapai objektif pembaikan mengatur masa laluan. Modul automasi Co*mputer Aided Design* (CAD) dibuat untuk mengintegrasikan penempatan tokokan ini. Modul automasi ini menjadi penyelesaian bagi mengoptimumkan masa pasca penempatan yang juga menyediakan strategi pelarasan sel penempatan di mana tiada pentindihan antara sel berlaku dan memastikan tiada perubahan yang nyata dari penempatan awal. Terdapat lima belas litar-litar tanda aras yang telah digunakan untuk mengesahkan keberkesanan CAD modul automasi ini. Keputusan daripada eksperimen menunjukkan ciptaan ini dapat mengurangkan masa kendur negatif maksimum sehingga 54.18 peratus. Purata sebanyak 5.64 peratus pembaikan masa berbanding dengan teknik piawai sel bergerak direkodkan dan ciri penempatan awal dapat dikekalkan dengan lebih baik. Ini menunjukkan ciptaan tersebut dapat mengurangkan masa kendur negatif maksimum dengan lebih berkesan berbanding dengan teknik sel bergerak.

CAD AUTOMATION MODULE BASED ON CELL MOVING ALGORIHTM FOR INCREMENTAL PLACEMENT TIMING OPTIMIZATION

ABSTRACT

Engineering Change Order (ECO) is a process to handle logic changes in circuit design. In deep sub-micron era, logic change in design happens inevitably. Design changes are required for numerous reasons. The reasons may be to fix design bugs, meeting design functionality change due to customer's requirement or optimize design performance such as power consumption. An incremental placement that has the capability to handle design changes efficiently manages to save time and cost. This is why ECO remains one of the most influential steps in Very Large Scale Integration (VLSI) design. This thesis describes timing driven incremental placement that uses standard-cell move technique to improve timing of the layout design. A cell filtering function is added in the incremental placement to enhance chances of layout design timing improvement. This function finalizes which cells need to be moved to achieve timing path improvement objective. A Computer Aided Design (CAD) automation module is developed to integrate the incremental placement. This automation module serves as a post-placement timing optimization solution that also provides a cells position adjustment strategy such that no cells overlap occur and ensure no significant deviation from initial placement. There are fifteen benchmark circuits that have been used to verify the functionality of the developed CAD automation module. Experimental results show that this approach can effectively reduce maximum negative slack timing up to 54.18 percent. An average of 5.64 percent timing improvement compare to standard-cell move technique is recorded and preservation on initial placement characteristic is better. This shows that the approach can effectively reduce maximum negative slack timing better compare to standard-cell move technique.

CHAPTER 1

INTRODUCTION

This thesis proposes a Computer Aided Design (CAD) automation module for timing driven incremental placement on Very Large Scale Integration (VLSI) circuit design. The CAD automation module is coded in Tool Command Language (TCL). The module runs using an Intel in-house tool in deep submicron environment setup. This chapter presents the project background, problem statement, research objectives, scope of work, project approach and tools, research contributions, and thesis organization.

1.1 Project Background

Figure 1.1 shows a VLSI design flow. The flow separates into behavioral, logic, circuit and layout representations (Nagi, 2010). The design flow starts from a system specification that defines the functionality and architecture of the design. Then, this behavioral representation is converted into logic representation at logic design step. The logic representation includes logic operation, arithmetic operation and control flow of the design. Logic design step will be followed by circuit design step. Circuit design is a step which outputs the schematics of the design. After that, the circuit representations are converted into geometrical shapes during physical design step. These geometrical shapes which also called layout representation will be manufactured in the corresponding layer to ensure the functionality of the design. Note that the verification of design plays a very important role in every step of the flow. Failure to properly verify a design in its early stages typically causes significant and expensive re-design at a later stage.

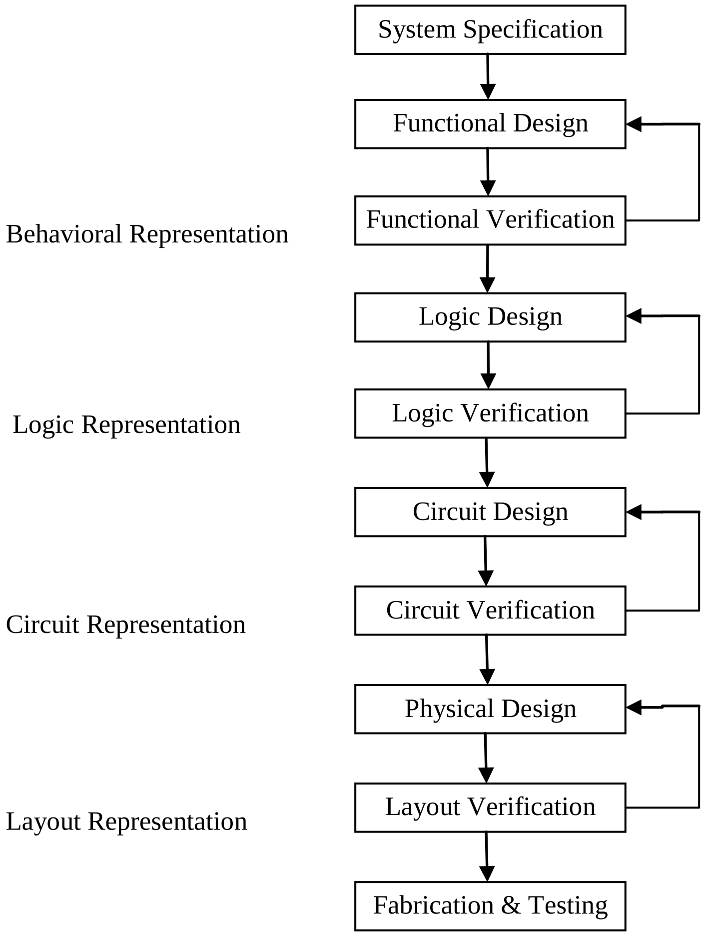


Figure 1.1: VLSI design flow

The last couple of decades witnessed explosive growth in the electronics industry due to the rapid advances in technologies integration of large scale electronic design. With the economy of large scale electronic systems blossom, the design process of these systems is undergoing a revolution. Integrated circuits today consist of billions of transistors. System designers of integrated circuits face challenge in both design complexity and capability to deliver to meet time-to-market requirement. Due to the massive increase in complexity, automating the design process not only becomes difficult but also getting huge demand globally.

As a result, almost all stages of design process extensively use CAD tools and many phases have already been partially or fully automated. The task of circuit design automation using CAD tools is called Electronic Design Automation (EDA). The objective of the EDA research field is to fully automate the tasks of every aspect of the development cycle, from design entry to the layout generation, verification, and performance analysis. The design automation is delivered in a systematic stage-by-stage manner especially in physical design process as large number of components required during the process simply beyond humans' eyes capability to deliver in this demanding industry. Physical design process is accomplished in several stages such as partitioning, floorplanning, placement, routing, extraction and verification. The physical design process is shown in Figure 1.2.

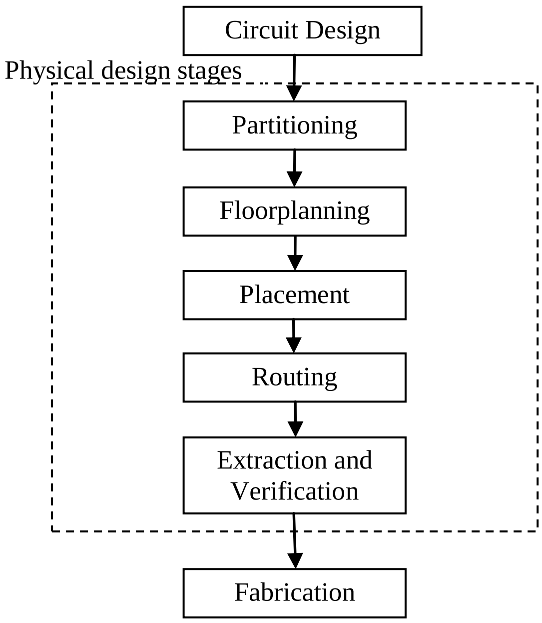


Figure 1.2: Physical design stages

Design may be modified during physical design stage to satisfy design requirements such as design functionality. The modification may involve replacement of large instance cells with variants in term of power consumption, cell delay, shape, and connectivity. New logic may be added in as well. Design timing may deteriorate from these changes and typically create overlaps. To support such changes which are known as Engineering Change Order (ECO) after placement stage, one could not afford to run general placement again because they are designed to generate a complete placement from scratch and thus time consuming. An efficient automation system is needed to obtain good incremental solutions in reasonable time. The automation system should ensure no overlap occurs while preserve as much initial placement as possible to maintain the performance of the circuit. It is because the initial placement should be optimizing in terms of wire length and other performances such as timing, area, and power consumption. To address this requirement, an incremental placement tool or ECO placement tool is required.

1.2 Problem Statement

Incremental placement is an iterations in physical design flow to correct design mistakes and accommodate changes made later in the flow. For example, a correction of timing violations discovered late in the physical design steps requires iterations in the design flow (Chen, 2005). Comprehensive study of incremental placement algorithms in the context of CAD tool development is an open area of research with a great deal of potential. Complete understanding and active participation in research and development in area of incremental placement algorithms would help to cope with the complexity of present day VLSI design. In present and future VLSI designs, geometries become smaller, operating frequency increases, and on-chip interconnect gains increased importance (Cong, 2000). At the same time, time-to-market pressure is driving the electronic automation strategists to reconsider design methodologies (Cong, 2000). Design processes require efficient incremental placement algorithms and methodologies. Incremental placement algorithm generally targets to optimize design metrics such as to reduce overall design's frequency, design power consumption, design area congestion or to meet design netlist changes. Incremental placement should be stable and a placement solution after incremental changes should be similar to the original placement with minimal perturbation so as to preserve the high quality of the original placement and maintain the physical information that physical synthesis uses for optimization (Chen, 2005).

An existing good placement with respect to a given metric may be modified to improve other metrics. Generally, it is extremely hard to modify a given "good" placement to meet particular objective without degrading previously minimized objectives. Given a placement produced by wire length optimization technique, it is unlikely that a timing improvement process can deliver without increasing total wire length. For instance, an algorithm in (Li, 2003b) manages to improve maximum negative slack timing using its applied timing model (Srinivasan, 1991) but total wire length is increased. Faster clock frequency, smaller device geometry, larger chip size and the demand of low power consumption have made timing related issues increasingly critical in VLSI circuits. In short, requirements for high performance and high speed VLSI circuit design have posed challenges to CAD systems especially for timing driven incremental placement. As VLSI design reached deep submicron era, timing driven incremental placement algorithm alone in (Li, 2003b) no longer applicable to improve negative slack timing of today design. An efficient scheme is needed to support the timing driven incremental placement algorithm in deep submicron era.

Responding to the problem stated above, a CAD automation module specifically for timing driven incremental placement is proposed. By providing a design placement layout with data paths propagation delay as inputs, a new timing driven incremental placement algorithm is executed.

1.3 Research Objective

The objective of this research is to:

a) Propose a timing driven incremental placement algorithm for initial placement that consists of negative slack timing.

b) Propose a CAD automation module that consists of timing driven incremental placement algorithm.

1.4 Scope of Work

The CAD automation module should be performed on the post-placement layout data. The performance evaluation of our CAD automation module is restricted to:

a) Test circuits are from International Symposium on Circuits and Systems (ISCAS) benchmark (Maksim, 2007). The benchmark circuits are synthesized and placements are generated through EDA industrial tool.

b) The required data such as data path delays is extracted from the EDA industrial tool. The data paths consist of negative slack timing. Then, the data will be used in our CAD automation module to initiate timing driven incremental placement.

c) Compare maximum negative slack timing with timing driven incremental placement algorithm in (Li, 2003b). The maximum negative slack timing gives an idea about maximum operation frequency of a design.

d) Reduce maximum negative slack timing of an initial placement. The placement after implemented the CAD automation module also ensure no cells overlap occur and no significant deviation from initial placement.

e) The output of CAD automation module is a placement, in a form that can be read in the EDA industrial tool.

f) The accuracy of the data path delays is dependent on the accuracy of the timing model in EDA industrial tool.

1.5 Project Approach and Tools

Figure 1.3 illustrates the overview of project workflow of the proposed CAD automation module development. There are three automation sub-modules that have been integrated and tested. The tests are conducted in (i) nanoscale process technology environment and (ii) using ISCAS85 and ISCAS89 benchmark circuits. An EDA industrial and Intel in-house placement tools have been used for demonstration of the results. Satisfactory outcomes from the performance evaluation successfully conclude the research.

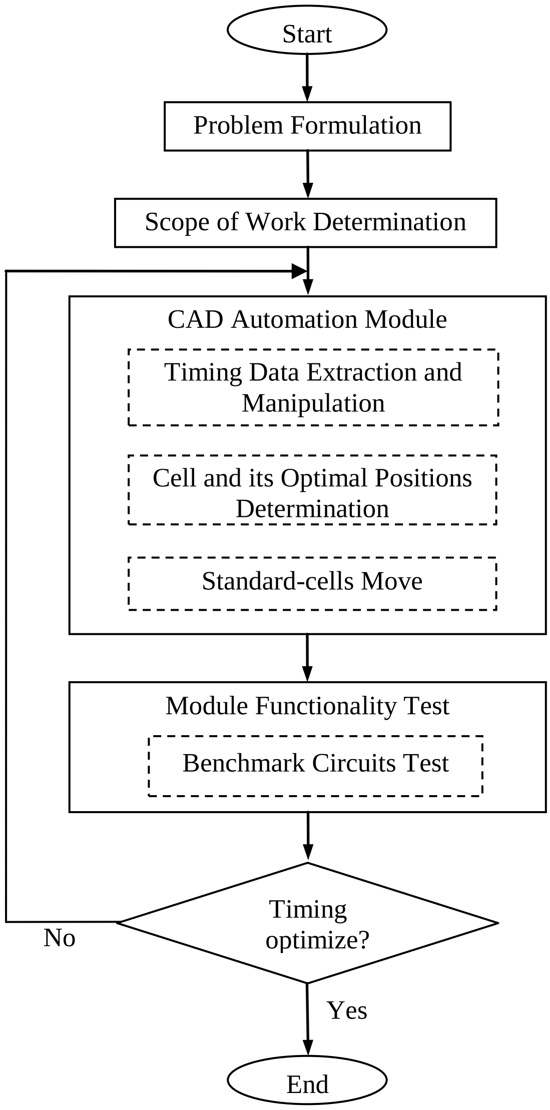


Figure 1.3: Project workflow of CAD automation module development

The following software tools are used in this work.

a) TCL scripting – used to develop the entire core program of proposed CAD automation module.

b) EDA industrial tool – used to re-produce ISCAS85 and ISCAS89 benchmark circuits in layout drawing design and data extraction.

c) Intel in-house Genesys tool – used to demonstrate the effectiveness of timing driven incremental algorithm.

1.6 Research Contribution

This research has a few contributions as shown below:

a) Delivering a timing driven incremental placement for initial placement that consists of negative slack timing.

b) Delivering a CAD automation module for timing driven incremental placement algorithm.

c) Adding cell filtering function as in (Li, 2003b) to improve negative slack path timing optimization

1.7 Thesis Organization

This thesis is organized into six chapters. First chapter is the introduction chapter. It covers the background of the research, problem statement, research objectives, scope of work, project approach and tools, research contribution and thesis organization.

The second chapter reviews the fundamental concept of incremental placement. It also consists of previous timing driven incremental placement research work and background theory.

Chapter Three delivers the proposed timing driven incremental placement algorithms. The algorithm separates to two majors section in Cell and its Optimal Position Determination and Standard-cell Move Technique. In section Cell and its Optimal Position Determination, a filtering function will be introduced to finalize which cell in negative slack timing need to be moved. Also, this section calculates an optimal position for those cells to be moved. Then, section Standard-cell Move Technique provides three algorithms to make sure no overlap between cells after cells moved to its optimal position.

Chapter Four describes implementation work of the proposed CAD automation module. The CAD automation module consists of main program unit and three sub-modules such as Timing Data Extraction and Manipulation Sub-module, Cells and its Optimal Positions Determination Sub-module, and Standard-cell Move Sub-module.

Chapter Five presents on how the test platform is been set-up. This chapter also shows the experimental results of the developed CAD automation module test on fifteen benchmark circuits follow by discussion of the experimental results.

The final chapter provides conclusion remarks, and future work recommendation.

CHAPTER 2

LITERATURE REVIEW

This chapter reviews the background of incremental placement. The chapter begins with general overview of incremental algorithms with different objectives in VLSI design and then discussion on previous related timing driven incremental placement work.

2.1 General Overview of Incremental Placement Algorithms

Depending on the design style and goal, an incremental placement algorithm may optimize different objectives. The objectives include meeting design specification, congestion-driven, power-driven, and timing-driven.

2.1.1 Meeting Design Specification

Sometimes it is necessary to make local modifications on circuit after placement to satisfy design specification. Local modifications are often made to react to local changes in the design and correct local errors. These modifications usually involve removal or addition of logic elements in the placed circuit. General purpose placement algorithms cannot take advantage of these situations because they are designed to generate a complete placement from scratch and thus very time consuming (Li, 2002). Mechanisms are needed to control the portions of the design that need to be changed only. Incremental placement consists of such mechanism to complete the modification with much lower computational time and cost. Moreover, the incremental placement technique should ensure minimize adjustment of initial placement and optimize wire length. The main challenge is to decide which portion of the design that need to be apply the mechanisms and what is the tradeoff between the design metrics such as power, area and speed.

2.1.2 Congestion-driven

Traditional placement generation objectives involve reducing net-cut costs or minimizing wire length (Dunlop, 1985) (Eisenmann, 1988). Because of its constructive nature, min-cut based strategies minimize the number of net crossings but fail to distribute them uniformly (Saab, 1996). For the same reason, traditional placement schemes which are based mainly on wire length minimization could not adequately account for congestion. Reducing net-cut and minimizing wire length might only help reduce the routing demand globally but do not prevent causing local routing congestion. There are history studies on how to estimate and reduce congestion in placement.

Congestion driven placement based on multi-partitioning was proposed in (Mayrhofer, 1990). It uses the actual congestion cost calculated from pre-computed Steiner trees to minimize the congestion of the chip. However, the number of partitions is limited due to the excessive computational load. Meanwhile, (Wang, 1990) proposed a consistent routing model defined by demand and supply relationship. Experimental results show that the congestion objective is very ill behaved. So it adapts a post processing approach after placement to reduce congestion. But the demand and supply congestion model and bounding-box routing estimation is too simple and will affect the final result. Since congestion and wire length are globally consistent, (Cong, 2000) considered improving local congestion with incremental placement.

In 2003, an incremental placement algorithm for improving local congestion is proposed (Li, 2003a). It first estimates the routing congestion through a new route model. A chip is divided into bins. Bin is considered congested if at least one of the bin edges routing possibility is greater than a certain threshold value. Then, cells in congested bins should move outside to decrease routing demand and achieve more routing resource. A cell flow tendency was introduced to determine cells to move out from congested bin so that changes to the initial placement could be minimized. Each cell has top, bottom, right and left gain to decide possibility of cell to move away from current bin so that nets crossing on the edge or routing possibility could be reduced. After that, an integer linear programming (ILP) problem is constructed to describe the moving orientation of cells. The solution of the problem will determine the destination bins of moved cells. Then, an efficient algorithm in (Li, 2002) is used to post processing moved cells into destination bins without overlap. The overall flow of the congestion driven incremental placement is shown in Figure 2.1. This incremental placement algorithm can avoid conflicts between adjacent congestion regions, but still no global consideration of congestion for entire design. Thus, (Luo, 2005) proposed an incremental placement algorithm for improving local congestion with a global view.

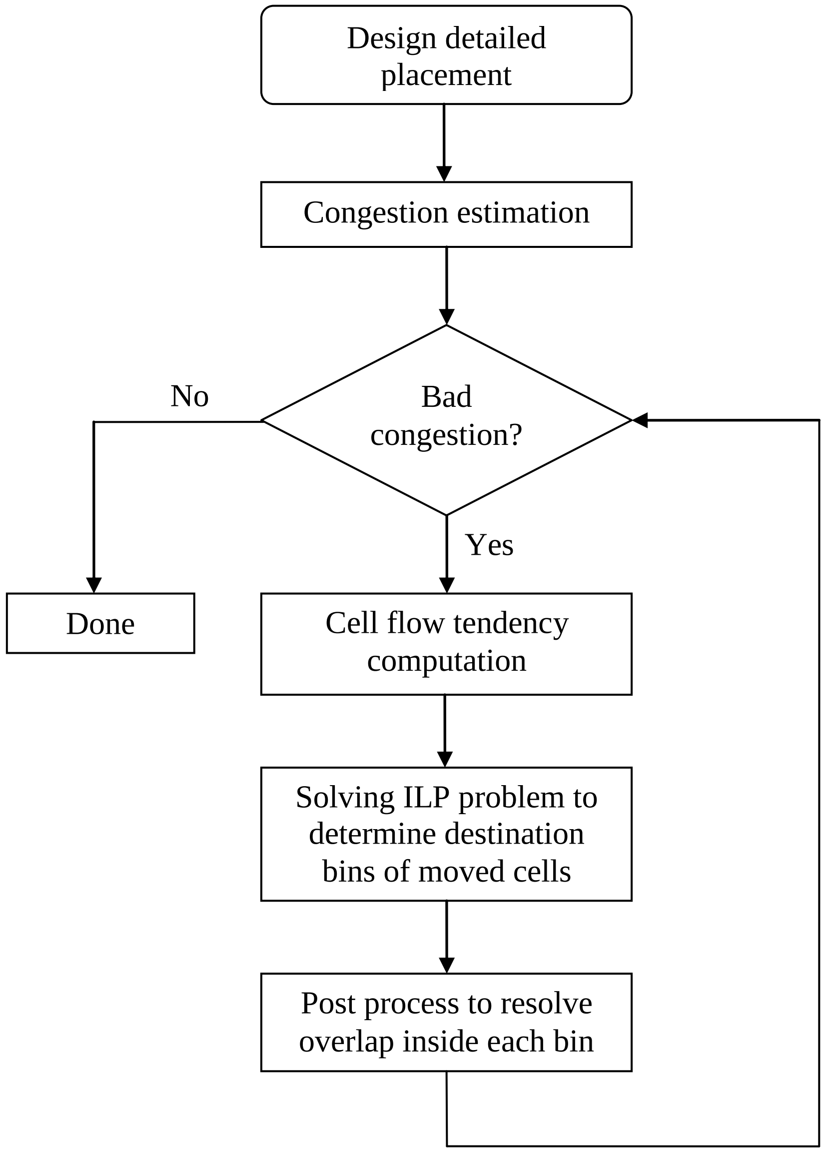
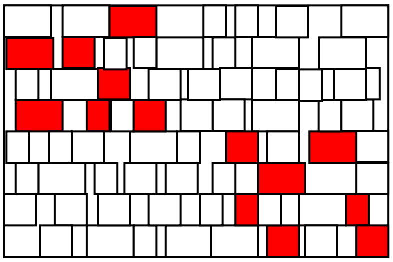


Figure 2.1: Congestion driven incremental placements (Luo, 2005)

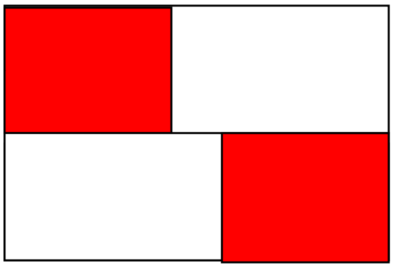
2.1.3 Power-driven

Power management has been considered as one of the important challenge facing the integrated circuit design industry today. The growing world-wide demand for portable electronics presses for low-power designs. The increasing circuit speed and shrinking feature sizes lead to much higher dynamic and leakage power, respectively, making the management of power dissipation more challenging than ever. Multi-Vdd is an effective method to reduce both dynamic and leakage power. It assigns high-Vdd to cells on timing critical paths and low-Vdd to cells on non-critical paths, so that power can be reduced without degrading the overall circuit performance. However, the resulting complex power supply system causes higher design cost, as more routing resource and heavy human intervention are required. Therefore, it is desired that cells of different supply voltages are grouped into a small number of voltage islands where each having a single supply voltage so that the design cost can be limited (Wu, 2006).

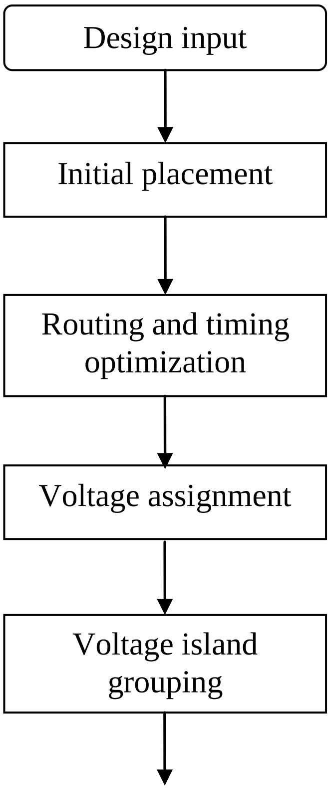
In 2005, (Wu, 2005) proposed an elegant algorithm that given a placement and a voltage assignment at the standard cell level that meets timing such that either power or the number of voltage islands is minimized under a bound on the other, while respecting the timing requirement. Figure 2.2(a) shows cells are automatically group together by voltage assignment and Figure 2.2(b) is a resulting voltage island. The grouping is based on the physical proximity of the high voltage cells. Furthermore, they proposed an efficient algorithm (Wu, 2006) to make the initial voltage assignment at the standard cell level, which not only meets timing, but also forms good proximity of high voltage cells for better voltage island grouping. The flow combining the two works is shown in Figure 2.2(c).



a) Voltage assignment



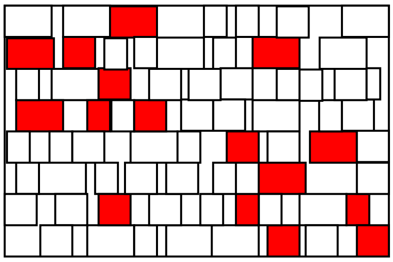
b) Voltage island grouping



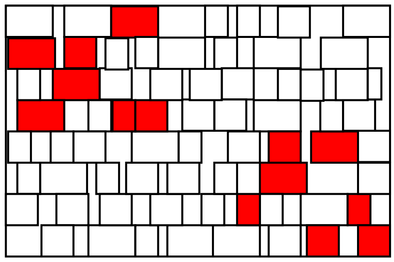
c) A multi-Vdd physical design flow

Figure 2.2: A low power design approach by multi-Vdd

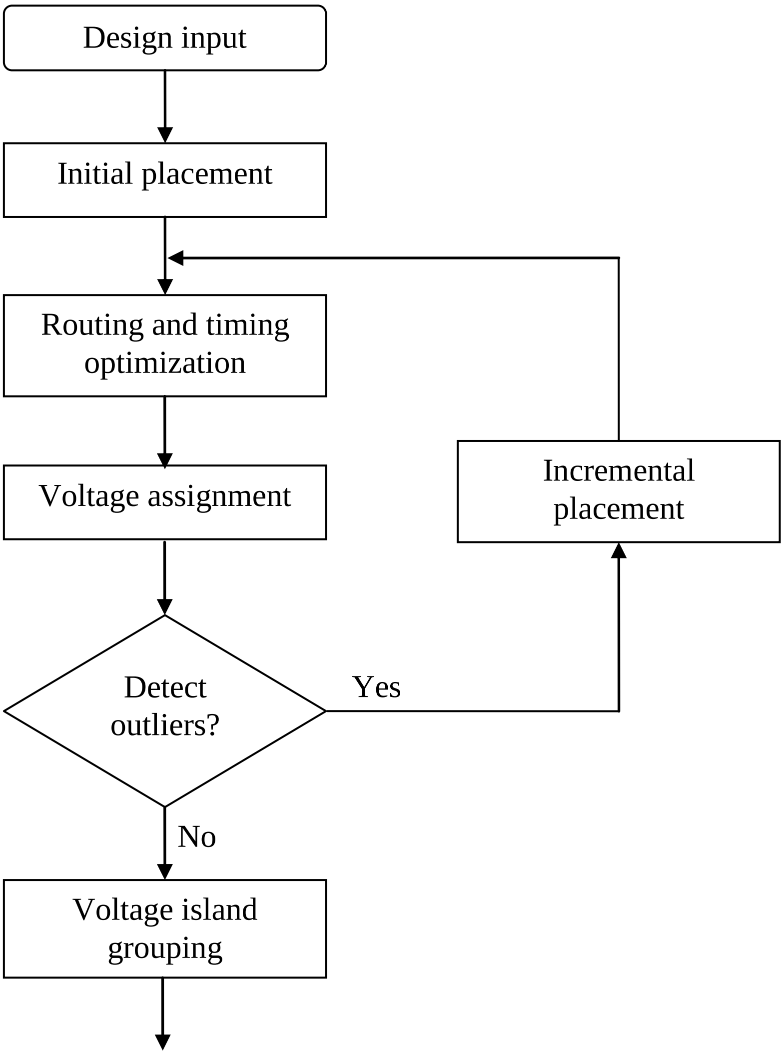
However, there is a certain limitation in this solution. Although the voltage assignment algorithm in (Wu, 2006) tries to assign high voltage cells close to each other and form large continuous areas of pure low voltage cells by allocating slacks according to the distribution of the already assigned high voltage cells which called *sites*, its freedom in doing so is limited by the amount of available slack on each timing path (Wu, 2007). Sometimes, even if some cells are located in a low voltage region and far away from other sites, they do not get voltage reduction due to insufficient slack on the path. They called this few distant site *outliers* (Wu, 2007). Compared to other sites, these outliers will cause disproportionately expensive penalty to the final voltage island grouping. Therefore, it is desired to eliminate them. The natural way for doing this is by modifying the placement. A first thought is to move the outlier cells from the low-Vdd region to a high-Vdd region. However, such a long distance movement is very likely to make some of the nets of the paths passing through the moved cells longer, and violate the timing. A feasible and effective way is to contract the nets on the critical paths passing through the outliers, thereby improve the timing on the paths and generating enough slack to reduce the voltage on the outliers. Thus, (Wu, 2007) presents a novel approach to improve the voltage assignment by eliminating outliers. Their approach consists of the following two steps. First, it automatically detects the outliers according to the site distribution in the current voltage assignment. Second, it performs an incremental placement to improve timing on the paths which have kept these outliers from voltage reduction, so that there will be more slacks for reducing voltage on these outliers in the next iteration. A flow including these two steps is shown in Figure 2.3(c). The improved voltage assignment with outliers being removed is shown in Figure 2.3(b).



a) Voltage assignment with outliers



b) New voltage assignment
after outliers removal



c) An iterative multi-Vdd physical design

Figure 2.3: Improve voltage assignments flow

2.1.4 Timing-driven

In high-speed circuits, multiple iterations might be needed in placement stage for timing closure. Given the complexities of design and tight time-to-market constraints, it is highly desirable to have a placement algorithm that needs fewer iterations of the whole optimization cycle and minimizing a given metric such as delay by minimally disturbs the current placement so that other metrics such as power consumption of the design do not change dramatically. For these reasons, incremental placement algorithms that focus on the most critical paths in the design is very helpful in design convergence.

Compare to timing driven placement from scratch, a timing driven incremental placement can focus on reducing delays of the most critical paths on an initial placement. This will greatly reduce the number of paths that need to be considered. Also, more timing information can be derived from an initial placement such as delay and slack estimates, which highly accurate as extraction done in physical stage. Timing driven incremental placement also finds applications in ECO scenarios where changes in the physical design stages generally required changes in the placement and routing stages. In such applications, timing driven incremental placement would make the required placement changes, while minimizing placement changes in the unaffected portion of the circuit. Thus, any deterioration in critical path delays is kept to minimum. Timing driven incremental placement can also be invoked in ECOs for the purpose of reducing delays of paths that already violate targeted clock speed constraint by appropriate placement changes in cells on these paths (Dutt, 2006).

2.2 Timing Driven Incremental Placement Techniques

There are few techniques to perform timing driven incremental placement algorithms. The techniques are gate sizing and buffering, technology remapping and standard-cell move.

2.2.1 Gate Sizing and Buffering

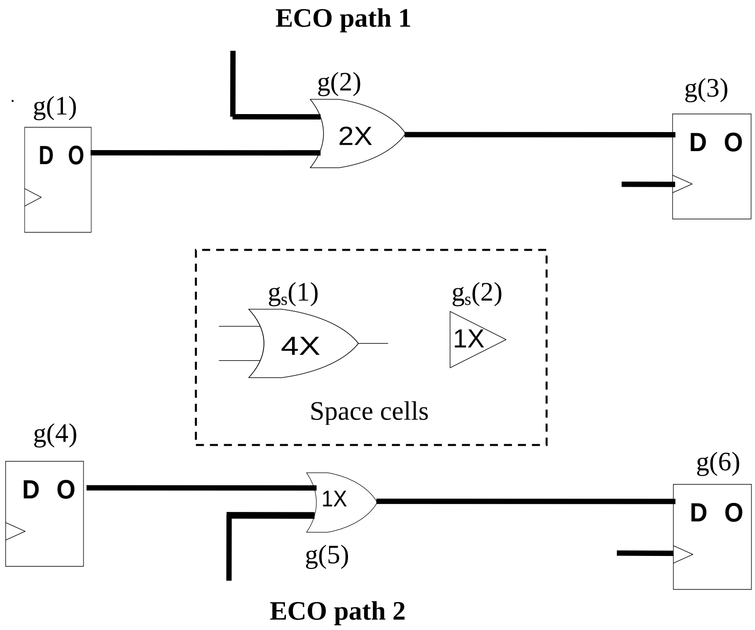
Gate sizing and buffering operations are often used for ECO timing optimization. The goal of gate sizing is to determine optimal sizes for the gates so that the circuit meets the delay constraints with least area and power cost. A larger gate will have higher drive strength and hence will be able to charge and discharge output capacitances faster. However, it also has a higher input capacitance. This results in the preceding gate seeing a larger capacitive load and thus suffering an increasing delay. Thus, sizing requires a careful balancing of these two conflicting effects. The optimal solution will thus require the coordination of the correct sizes of all the gates along and off critical paths.

Buffering, like gate sizing, is an electrical optimization and not a logical optimization in the sense that it does not change the logical structure of the netlist. Buffering can be used to increase the drive strength for a node that is driving a large load. A chain of one or more buffers can be used to drive a large load. Buffering also been used to restore signal levels and shield signals on a critical paths from high-load off-critical-path signals by driving the off-critical path load. Given that buffering a given net can change the constraints on the pins of another net, the final solution is sensitive to the order in which the nets are visited. In addition, once a net is buffered, the gates may no longer be optimally sized. Resizing gates before the next net is buffered can modify the buffering problem. Researchers have considered combining sizing and buffering into a single step (Jiang, 1998), but again this problem is very complex and far from being considered as solved.

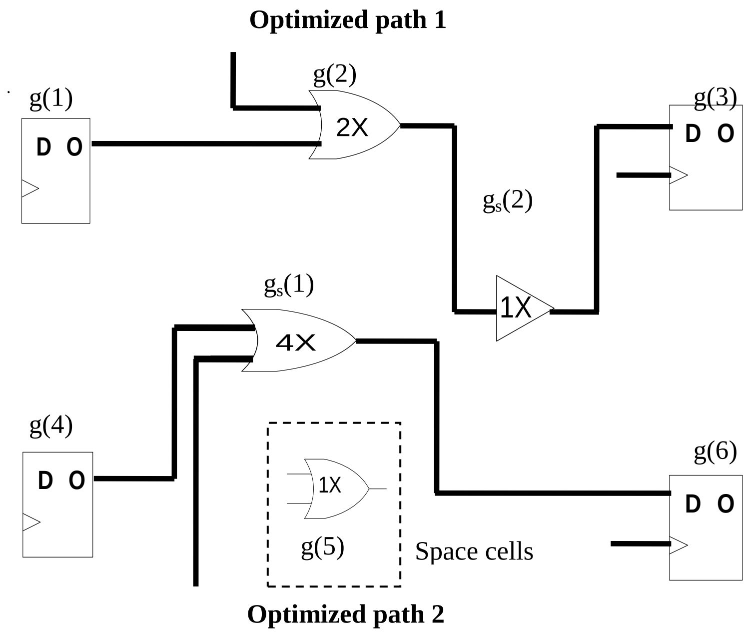
Spare-cell rewiring (Chen, 2007) is a way to perform buffering and gate sizing. Spare cells are standard cells not connected to any circuit but are designed to facilitate circuit debugging, and also to reduce mask cost. They are often evenly placed on the chip layout. The type and number of spare cells vary from different chip designs and are usually determined by designers empirically. By changing net connections, selected spare cells to perform buffering and gate sizing can be merged into a netlist to form a new netlist. The new netlist does not need to be placed again. In this way, the time for placement after the design change is saved. No major changes on layout structure. If the design has ever been taped out, only the masks of metal layers need to be re-produced. Consequently, they can substantially save the production cost since masks are very expensive in nanometer designs. Although spare-cell rewiring is a very good ECO technique, the selection of spare cells and the competition for using a spare cell among multiple paths make the rewiring problem a challenging problem.

Figure 2.4(a) shows an instance of timing optimization by rewiring spare cells. The OR gate $g_S(1)$ and the buffer $g_S(2)$ are spare cells and are initially not connected to any path. Gates $g(1)$, $g(3)$, $g(4)$, and $g(6)$ are D-type flip-flops. The gates $g(1)$ and $g(4)$ is the source of path 1 and path 2 respectively, and gate $g(3)$ and

$g(6)$ is the sink of path 1 and path 2 respectively. Suppose that the delays of paths 1 and 2 violate the timing constraints, the timing of path 1 can be improved by inserting the adjacent buffer $g_S(2)$ into the path to drive the load. To fix the timing violation of path 2, we can use the OR gate $g_S(1)$ instead of the OR gate $g(5)$ on path 2 because $g_S(1)$ has a larger driving capability. After the rewiring, both paths satisfy the timing constraints, and $g(5)$ is released from the netlist and becomes a spare cell.



(a) ECO paths before rewiring



(b) Paths after rewiring

Figure 2.4: Timing optimization by rewiring space cells

2.2.2 Technology Remapping

Technology remapping is an operation to reconstruct the circuit to fix timing violations. Technology remapping attempts to find the best selection of cells from a given cell library to meets a given delay constraint with least area or power. Post-physical design mapping is a remapping step that attempts to find better mapping by using the existing physical information for determining interconnect delay. The challenge is to work on small sections so that the given placement is not significantly disturbed, yet at the same time be effective enough to improve the design. Mapping has been well studied (Hachtel, 1996) and the mapping algorithms are well known but knowing where to apply them is the challenge. One aspect of the problem is determining where to place the new cells created during the remapping phase. Typically some simple solutions based on the fixed boundary locations are used during mapping itself, with a clean up step to make the placement legal (Pedram, 1991).

Traditional layout driven technology mapping typically places standard cells first and then performs technology mapping with the known information about the physical positions of the mapped standard cells. Once a standard cell is placed, its physical position is fixed and so is the wiring cost by using this cell. In other words, each standard cell is tagged with a fixed cost on its area or power. Recently, spare cell-aware technology remapping is developed (Ho, 2010), This spare cell-aware technology remapping, in contrast, typically has multiple choices of spare cells of the same type for mapping the target logic function. By selecting different or even same spare cells during the technology remapping incurs different wiring costs. Therefore,