A P-CELL APPROACH TO INTEGER GATE SIZING

A Thesis

by

UDAY DODDANNAGARI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2007

Major Subject: Computer Engineering

A P-CELL APPROACH TO INTEGER GATE SIZING

A Thesis

by

UDAY DODDANNAGARI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Weiping Shi
Committee Members,	Jiang Hu
	Yong Kyu Jung
Head of Department,	Costas Georghiades

December 2007

Major Subject: Computer Engineering

ABSTRACT

A P-cell Approach to Integer Gate Sizing. (December 2007)

Uday Doddannagari, B.E(Hons.), Birla Institute of Technology & Science

Chair of Advisory Committee: Dr. Weiping Shi

Standard-Cell-library-based design flow is widely followed in the Application Specific Integrated Circuit(ASIC) industry. Most of the realistic cell libraries are geometrically spaced introducing significant sparseness in the library. This is because uniformly spaced gate sizes would result in a large number of gate sizes and maintaining the huge volume of data for this number of gate sizes is difficult. This thesis aims to propose a practical approach to implement integer gate sizes. A parameterized cell (p-cell) approach to the generation of layouts of standard gates is presented. The use of constant delay model for gate delay estimation is proposed which eliminates the need for maintaining huge volumes of delay tables in the standard cell library. This approach has tremendous potential since it greatly simplifies the standard-cell-based design methodology and can give significant power and area savings.Power and area savings of up to 28% are possible using this approach.

To My beloved father, late Prof. D. Raja Reddy

ACKNOWLEDGMENTS

Many thanks to my advisor, Prof. Weiping Shi, for giving me such an interesting topic to work on. I would like to thank him for all the support and for his invaluable advice during the course of research. Also thanks to my committee members, Dr. Jiang Hu and Dr. Yong-Kyu-Jung for their encouragement.Many thanks to Karandeep for his help and encouragement when things were not going well in my research. Special thanks to my sister, Lavanya and my mother, Krishnamma for motivating me towards doing thesis work. I thank all my friends without whose support this thesis would not have been possible.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

Standard Cell library based design flow is widely followed in ASIC industry. A standard cell library is a library of cell/gate types such as INVERTER, NAND and NOR gates implemented in multiple drive strengths in a given processing technology. The importance of maintaining multiple drive strengths or sizes of each gate type in the cell library is well known. In [1] Keutzer and Scott demonstrated experimentally that it is advantageous to maintain multiple drive strengths of each cell type in the library. In principle, if we make available all the sizes for each cell type in the library, we get the best performance of the circuit after synthesis. Also, the gate sizing problem then becomes a continuous gate sizing problem for which there are well known techniques one of which is a posynomial programming approach [2]. In reality however most of the cell libraries are sparse. In fact the cell sizes are not even uniformly spaced. This is because uniformly spaced gate sizes would result in a large number of gate sizes and this puts a heavy burden on managing the data volume of the library and on the gate sizing algorithms. Since only a few discrete gate sizes can be maintained in the library, optimal selection of these gate sizes for standard cell libraries has gained importance. In this direction, it is proved in [3] that under certain conditions, the set of optimal gate sizes must satisfy a geometric progression.

A. Continuous/integer gate sizing approach

A practical implementation of continuous or even integer gate sizes has the potential to significantly improve the quality of designs. The best known gate sizing technique

The journal model is *IEEE Transactions on Automatic Control.*

for discrete gate sizes [4] uses a continuous solution guided approach to come up with a discrete gate sizing solution. It can clearly be seen from this work that the final discrete solution using a library of geometrically spaced cell library has a significant area overhead compared to the continuous solution. Also gate sizing algorithms are much more simpler for continuous gate sizing case than for the discrete gate sizing case. Hence there is a need to come up with techniques to practically realize continuous or near-continuous/integer gate sizes. One approach in this direction is the use of discrete gate sizes in parallel to realize an integer gate size. In [5], it has been proposed that the use of just 5 discrete gate sizes can be used to realize any continuous gate size with an accuracy of 1.7%. However using discrete gate sizes in parallel imposes some constraints on the 'place and route' tools. For example the place and route tool must make sure that the discrete gate sizes used to realize a gate are all placed close together since placing them far apart can give rise to short circuit between the vdd and ground supplies. Since the timing information of only a few discrete gate sizes is available in this approach, one more challenge faced by this approach is to predict the timing information of the Integer gate size realized using discrete gate sizes in the library.

The goal of this thesis is to propose a practical approach for realizing integer gate sizes without imposing any additional constraints on the synthesis tools. Our aim is to come up with a methodology that would dramatically reduce the data volume maintained in the cell library while still allowing all integer gate sizes. We now focus on the components of data volume stored in the standard cell library for each cell.

B. Composition of a cell library

The main components of the data volume stored in the cell library for each cell are 1) Layout of the cell that can be used by the automated place and route tools. 2) Timing information of the cell which is stored as a delay table.

1. Layout view

Fig. 1 shows the layout of a typical standard cell. The layout structure that is shown in the figure has been in use for many years now. It was first proposed by Uehara and Vancleemput in 1978 [6]. The standard layout structure consists of horizontal Metal1 lines running at the top and bottom. The top line is the VDD supply whereas the bottom one corresponds to the Ground supply. There is a pmos section at the top and an nmos section at the bottom. For a given gate type, multiple layouts which differ in their drive strengths are maintained in the library. It can be seen from this layout that the diffusion regions of adjacent transistors are shared. Sharing of diffusion regions is a key advantage of this layout structure since the area is reduced compared to separately laid out transistors.

2. Timing information

The timing information of each standard cell is stored in the cell library. The timing information typically comes as a delay table. Once the layout is made, the parasitics of the gate are extracted to get the SPICE netlist of the gate. The delay of a gate depends on the input transition time and the load capacitance that it drives. Hence the input transition time and load capacitance of the gate are varied in steps during simulations to construct the delay table for that particular cell. This cell delay table stores the gate propagation delay and output rise/fall time of the cell as a function of

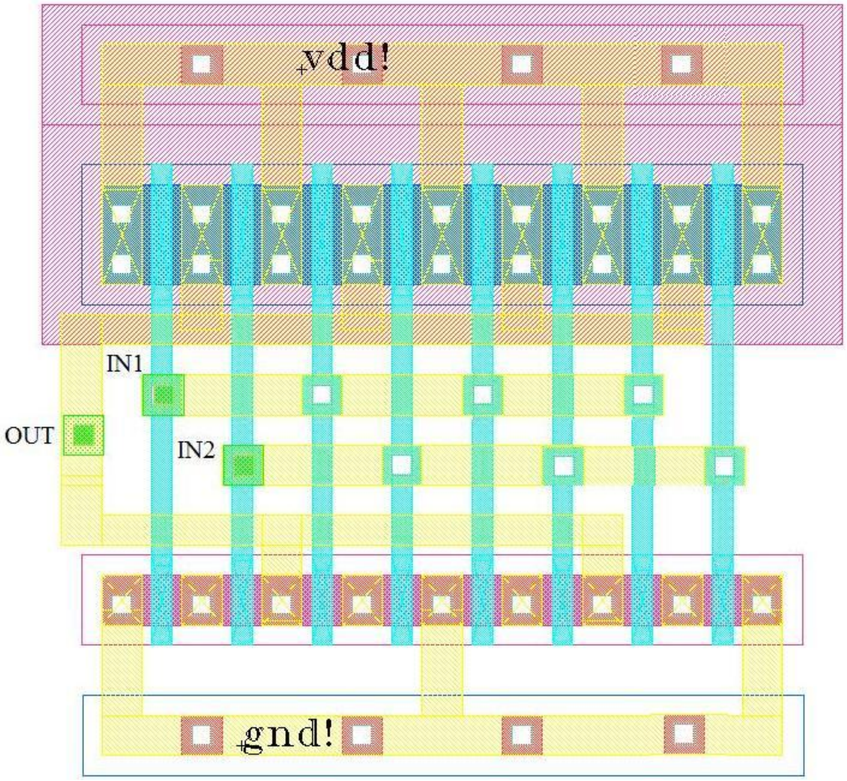


Fig. 1. A typical standard cell layout. Layout of a nand gate

the load capacitance and input transition time. Separate delay tables are maintained for cells of different drive strengths in the cell library even if the cells are of the same type. A typical delay table for a gate in Fig. 2 is shown in the Fig. 3. For intermediate values of input transition time and output capacitance, piecewise linear approximations are used. These tables are equivalent to empirical 'k-factor' formulas for delay and output transition time.

The delay tables of gates are used during timing analysis. In VDSM technologies, we can not assume that the load is purely capacitive. The interconnect resistance is significant and can not be ignored. Hence in reality, the load is an RC load. Hence in order to use the delay tables, we need to determine the effective capacitance of

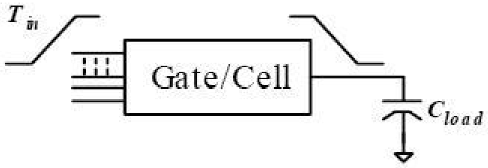


Fig. 2. A gate driving a load capacitance. The gate delay is a function of the input transition time and load capacitance

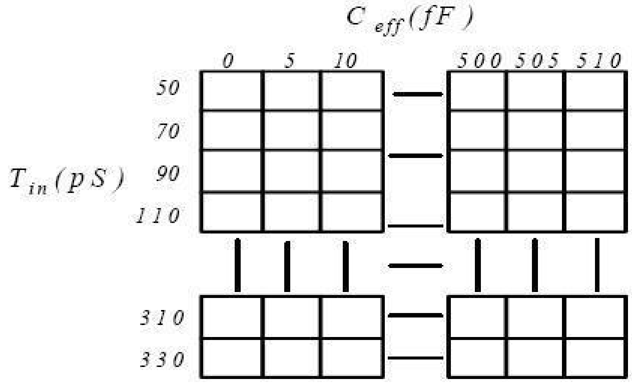


Fig. 3. A typical delay table

this RC load and then use this value to look up the delay tables. The simplest way of computing effective capacitance is to sum up all the capacitances present in the RC load. However this method is too pessimistic. There are techniques to compute the effective capacitance of an RC interconnect line in very deep submicron technologies which allow accurate estimation of the 50% propagation delay and the output transition time using the delay tables [7]. Given an RC load that is driven by a gate, these techniques are used to determine the effective capacitance of the load. This value of capacitance along with the known input transition time are used to look up the delay tables for that particular gate to obtain the 50% propagation delay and

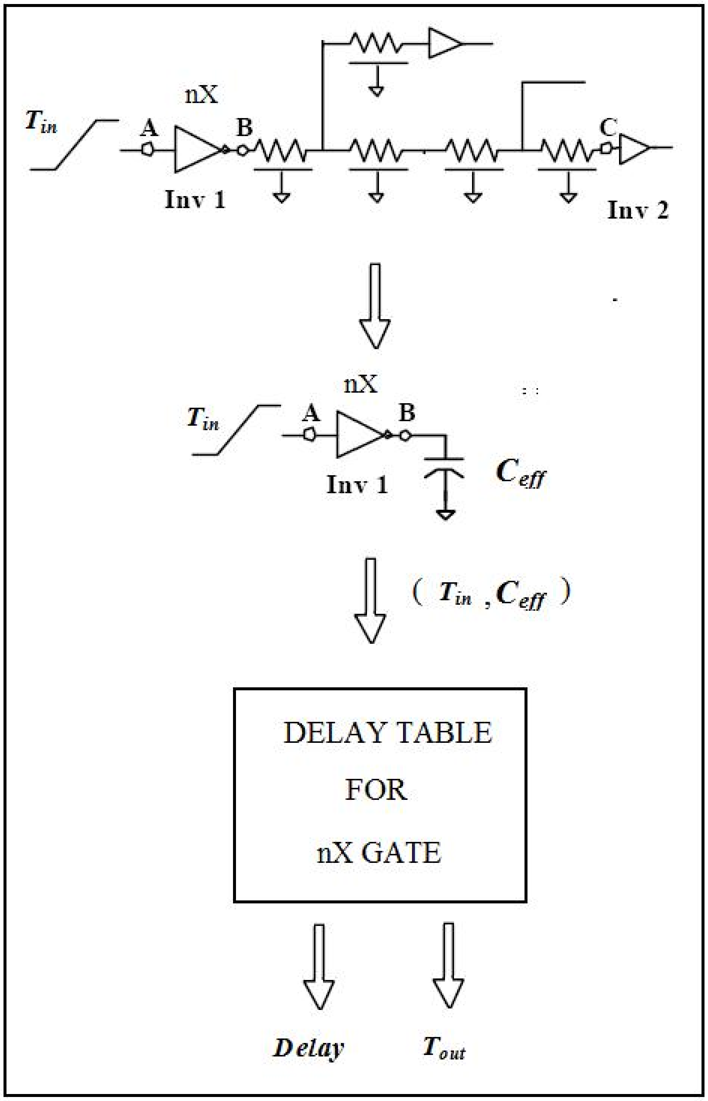


Fig. 4. Figure illustrating the use of delay tables for timing analysis

the output transition time. These steps are illustrated in Fig. 4.

As shown in this figure, if the driving gate is a nX gate then the delay table for nX table is looked up in order to determine the delay. If there are a large number of driving strengths of the same gate type in the cell library, the delay tables of all these gates have to be maintained for timing analysis. The impracticality of maintaining such huge volumes of data has resulted in the use of sparse cell libraries where the gate sizes are geometrically spaced. In the next section, we propose our new delay model that can predict the delay of a gate of any size, given the load capacitance and the input slew rate using the delay table of only a unit cell. This approach eliminates the need of maintaining huge volumes of cell delay tables in the standard cell library.

C. Goal of the thesis

In this thesis, we propose a parameterized cell (p-cell) approach for generating the layouts of cells whose sizes are integer multiples of a *unit cell.* A p-cell of a gate type takes the size of the cell as the input parameter and generates the layout of that particular size. If such an approach is followed, there is no need to maintain multiple layouts for a gate type. A single layout structure can be used to generate the layout/layout information of a cell of any size. Further we propose the use of *unit-cell delay model* to represent the delay information of the cell type and also discuss the constraints imposed on the layout by this delay model to be practical. We then propose p-cell structures for some standard gates. A consequence of this integer gate sizing approach is that it gives us a lot of freedom to choose the drive strengths of gates in our designs without having to worry about the problems of managing huge data volumes.

The idea of automating the layout generation of standard cells is not new. In

[8], the authors propose the use of a Module Generator to automatically generate the layout of a standard cell. However the layouts generated do not follow a p-cell approach and are not compatible with the unit-cell delay model. As a result, every time a module generator is called it has to generate the timing information of the cell in addition to the layout information. Our approach is different. Our p-cell generator generates the layout of a cell of the desired size and does not need to generate the timing information because of the symmetric layout structures. The timing information of a unit size of the cell type alone represents the timing information of any other size of the cell type.

D. Thesis guide

This section gives a brief description of the organization of this thesis. This thesis has been divided into four chapters. In chapter II we discuss two possible approaches of integer gate sizing namely parallel cell approach and a parameterized cell approach. In chapter III we present parameterized cell structures of four basic gates namely INVERTER, NAND, NOR and EXOR gates. In chapter IV we propose the use of constant delay model to represent the delay information of p-cells. In chapter V we discuss the constraints imposed by the unit-cell delay model on the layout structures. Chapter VI presents the conclusions and also future work that can be taken up.

CHAPTER II

POSSIBLE APPROACHES

There are two possible approaches to realize integer gate sizes. The first approach is to use the existing geometrically spaced discrete gate sizes in parallel to realize any integer size. The second approach is the parameterized cell (p-cell) approach which is the main idea of this thesis.

A. Parallel cell realization

Suppose we have a geometrically spaced cell library. We assume that the different gate sizes in the library are created by scaling all the transistor sizes by the same factor. Then we can use these sizes in parallel to realize other integer gate sizes. Fig. 5 [5] shows the connection of two nand gates in parallel to realize a nand gate of a size which is the sum of the sizes of the smaller nand gates. Let the sizes in the library be 1X,2X,4X,8X..2^nX. Then any integer size within this range can be realized by using a combination of the existing sizes. Precisely, the minimum number of discrete gate sizes that can be used to represent a given integer size n is given by the number of 1s in the binary representation of n and the sizes that are to be used are the corresponding bit weights. For example, suppose the size to be realized is 10X, the number of 1s in the binary representation of 10 is 2 (1010). Hence the 10X gate can be realized using 2X and 8X gates in parallel.

Though this approach can realize gates integer sizes, it imposes some constraints on the place and route tools. Suppose we decide to use a 2X cell and a 8X cell in parallel to realize a 10X cell. Then they have to be placed as close as possible to each other. If they are placed far apart, then a difference in the interconnect delays between the input pins of the two cells may result in a short circuit current.

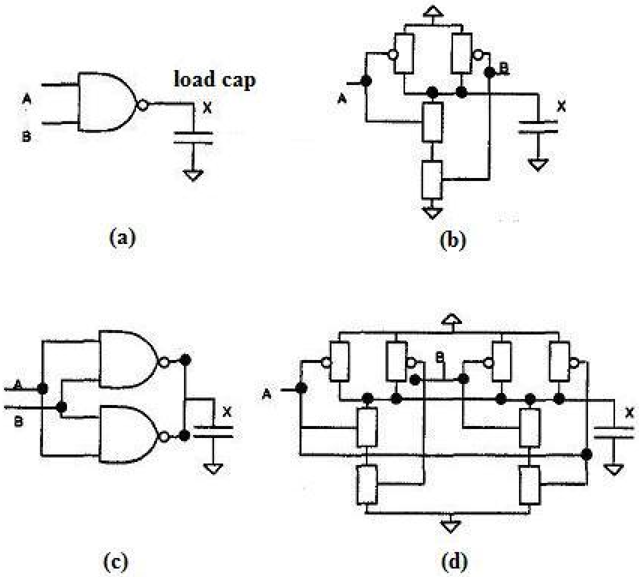


Fig. 5. A parallel cell realization example

Consider the Fig. 6 which illustrates this case. The figure shows an 8X inverter and a 2X inverter connected in parallel. But the input signal arrives early at the 2X gate because of the interconnect between the input lines. Then we come across a scenario where the two inverters try to drive opposite logic levels to the output hence causing short circuit between vdd and ground. To avoid this problem, the cells must always be placed close together.

B. Parameterized cell (p-cell) realization

A parameterized cell (p-cell) is defined as a programmable cell that takes a set of input parameters to quickly generate the layout of a cell. A practical approach to realize integer gate sizes is to use a p-cell approach. P-cell-based approach saves us a lot of disk space since storing only one layout structure enables us to generate

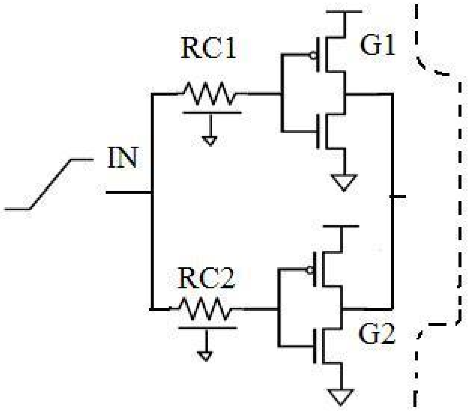


Fig. 6. Figure illustrating the short circuit problem if the cells are placed far apart

many variations of the layout depending on the input parameters. A cell type in the standard cell library can be easily implemented using a p-cell-based approach if it has a fixed layout structure. In other words if each and every detail of a cell type's geometry and position can be expressed in terms of certain input parameters, p-cell approach can be used to realize any cell size of that cell type.

We propose the use of a p-cell whose input parameter is the size of the cell. The inverter in the figure on page 15 shows the p-cell structure of an inverter with input parameter of n. This layout structure consists of n poly fingers, the reason for which is explained later. The p-cells of some other standard cells are presented later. It can be seen from the figure that the layout is symmetric in the horizontal direction. The position and size of every rectangle can be expressed as a linear function of the size of the cell. Because of the simplicity of the structure, every detail of the layout can be dynamically generated and used by the place and route tools. The only information we need to store in the library are a set of linear equations in the size of the cell.

One main advantage of p-cell approach is that it eliminates the need for main-

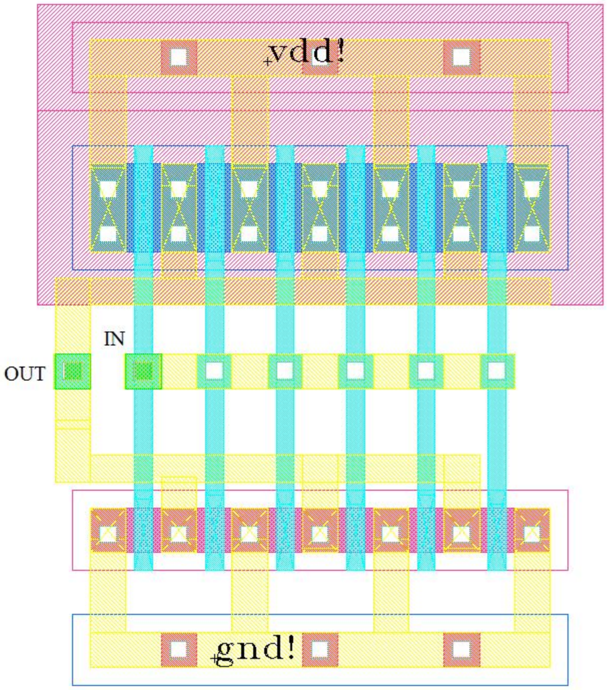


Fig. 7. P-cell of an inverter with input parameter 3

taining multiple layouts of the same cell type in the library. P-cell realization occupies lesser area compared to the parallel cell realization since the layout is realized as a single unit. In parallel cell realization the cells, though placed adjacent to each other have a minimum spacing between them, thus causing an area overhead. Thus the p-cell approach is better compared to the parallel cell approach.

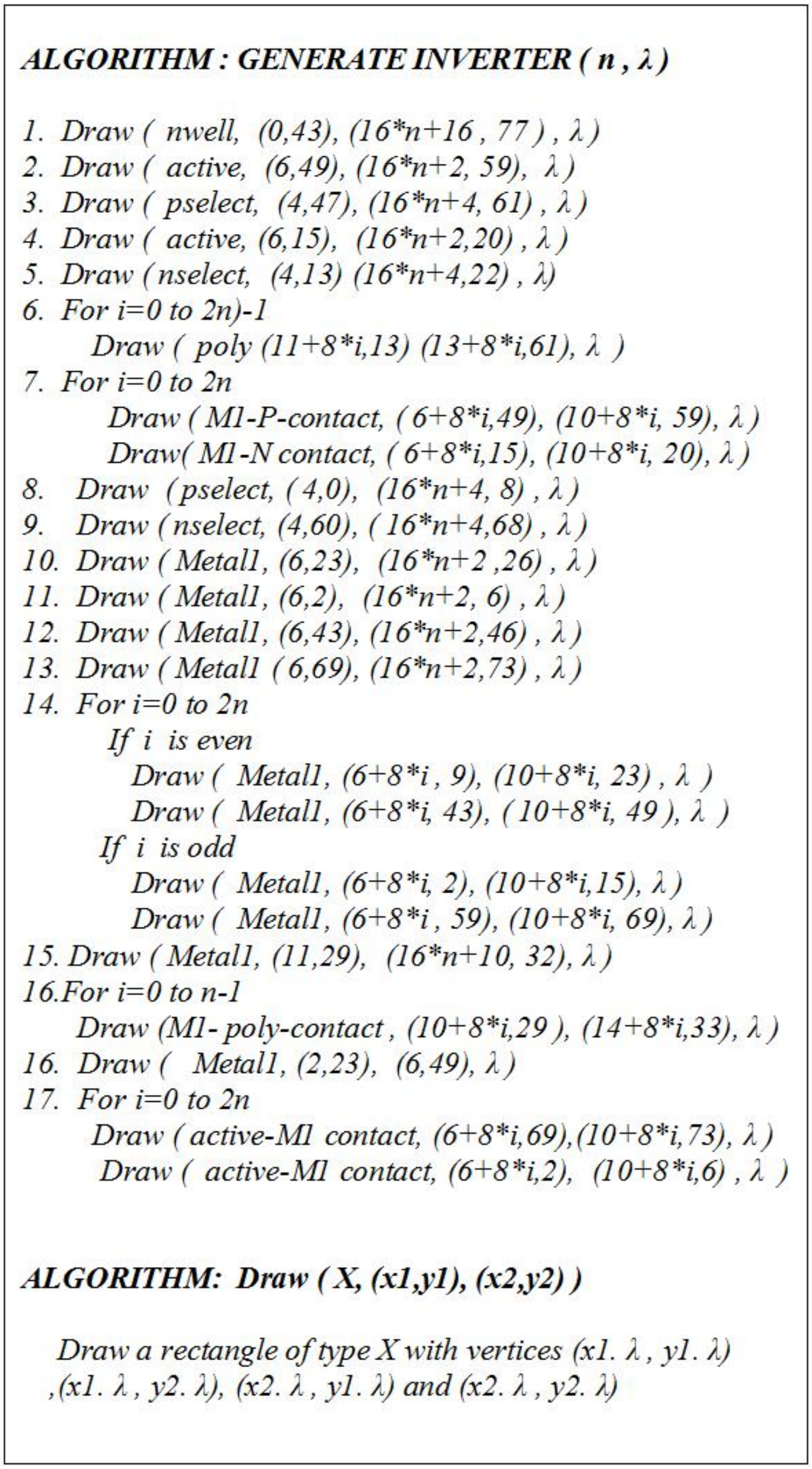
CHAPTER III

P-CELLS OF SOME STANDARD GATES

In this section, we propose p-cell structures for some standard gates. As mentioned earlier, a p-cell can be visualized as a module generator that takes a set of input parameters and generates the layout of a cell. The key requirement of such layout structures is symmetry in the layout. Every detail of every mask in the cell must be able to be represented in terms of the input parameters. The input parameters for the p-cells in our case are the size of the cell and λ, the process technology parameter.

A. Inverter

The unit cell of an inverter consists of two poly fingers.The reason for this will be explained later in the thesis when we talk about our delay model. We use a two-poly-finger inverter in the figure on page 33 as our unit cell. Fig. 8 is a representation of our p-cell.This algorithm takes the size of the inverter, n, as the input parameter and generates the layout in a given processing technology. The algorithm takes the bottom left corner of the cell to be the origin and generates each of the polygons in the layout .The algorithm also takes λ as the input parameter. λ is half the minimum gate length in a given processing technology .As seen in the algorithm, the position and size of every rectangle in the layout of inverter can be expressed in terms of the input parameters n and λ. Fig. 8 shows the layout of an inverter generated by the algorithm with input parameters of 3 and 0.09μm.In this layout structure, the poly connections across the unit cells have been made using horizontal Metal1 lines and the diffusion regions at the extremes have been connected to the respective bulk terminals. The drain regions of the nmos/pmos section are connected using horizontal Metal1 lines and these horizontal Metal1 lines are then connected to each other using



```
ALGORITHM : GENERATE INVERTER ( n , λ )

1.  Draw ( nwell,  (0,43), (16*n+16 , 77 ) , λ )
2.  Draw ( active,  (6,49), (16*n+2, 59),  λ )
3.  Draw ( pselect,  (4,47), (16*n+4, 61) , λ )
4.  Draw ( active, (6,15),  (16*n+2,20) , λ )
5.  Draw (nselect,  (4,13) (16*n+4,22) , λ)
6.  For i=0 to 2n)-1
        Draw ( poly (11+8*i,13) (13+8*i,61), λ  )
7.  For i=0 to 2n
          Draw ( M1-P-contact, ( 6+8*i,49), (10+8*i, 59), λ )
          Draw( M1-N contact, ( 6+8*i,15), (10+8*i, 20), λ )
8.   Draw  (pselect, ( 4,0),  (16*n+4, 8) , λ )
9.   Draw (nselect, (4,60), ( 16*n+4,68) , λ )
10.  Draw ( Metal1, (6,23),  (16*n+2 ,26) , λ )
11.  Draw ( Metal1, (6,2),  (16*n+2, 6) , λ )
12.  Draw ( Metal1, (6,43), (16*n+2,46) , λ )
13.  Draw ( Metal1 ( 6,69), (16*n+2,73) , λ )
14.  For i=0 to 2n
          If  i  is even
            Draw (  Metal1, (6+8*i , 9), (10+8*i, 23) , λ  )
            Draw (  Metal1, (6+8*i, 43), ( 10+8*i, 49 ), λ  )
         If  i  is odd
            Draw (  Metal1, (6+8*i, 2), (10+8*i,15), λ )
            Draw (  Metal1, (6+8*i , 59), (10+8*i, 69), λ )
15. Draw ( Metal1, (11,29),  (16*n+10, 32), λ )
16.For i=0 to n-1
      Draw (M1- poly-contact , (10+8*i,29 ), (14+8*i,33), λ )
16.  Draw (   Metal1, (2,23),  (6,49), λ )
17.  For i=0 to 2n
        Draw ( active-M1 contact, (6+8*i,69),(10+8*i,73), λ )
         Draw (  active-M1 contact, (6+8*i,2),  (10+8*i,6) , λ  )


ALGORITHM:  Draw ( X, (x1,y1), (x2,y2) )

   Draw a rectangle of type X with vertices (x1. λ , y1. λ)
  ,(x1. λ , y2. λ), (x2. λ , y1. λ) and (x2. λ , y2. λ)
```

Fig. 8. Algorithm to generate an nX Inverter

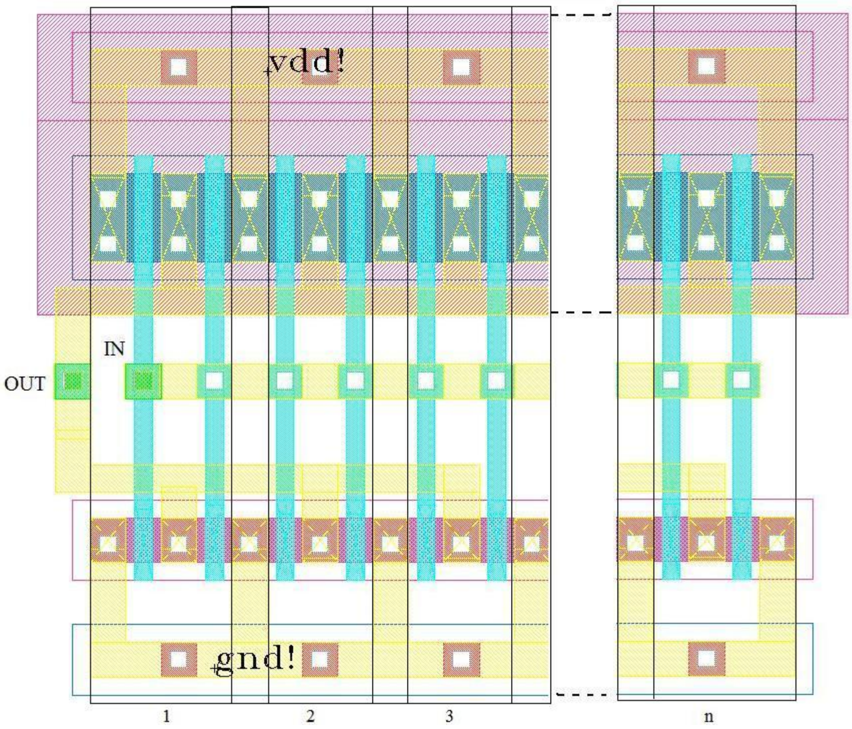


Fig. 9. The layout of a nX inverter laid out using 2n poly fingers

a vertical Metal1 line towards the left of the layout. Vertical Metal1 lines connect the source regions to the bulk. Fig. 9 shows the layout of a nX inverter. This fixed layout structure enables us to generate the layout of inverter using a p-cell based approach.

B. NAND Gate

The unit cell of a nand gate is shown in Fig. 10. The unit cell consists of four poly fingers. The connections of the diffusion regions to VDD and Ground are made through vertical Metal1 lines. The poly fingers are all connected using Metal1.The reason for this is explained in the following sections. Also it can be seen that the unshared diffusion regions are connected to the respective bulk terminals. The output Metal1 lines run parallel to the nmos and pmos sections and are connected to each other towards the left of the layout through a vertical Metal1 line.The symmetry in the layout enables us to use this as the unit-cell for our p-cell of a nand gate.Fig. 11 shows the layout of a nX nand gate formed using n unit cells of the nand gate in parallel.

C. NOR Gate

The unit cell of a nor gate is shown in the Fig. 12. The unit cell consists of four poly fingers. The connections of the diffusion regions to VDD and Ground are made through vertical Metal1 lines. The poly fingers are all connected using Metal1. The reason for this is explained in the following sections when we discuss the delay model. Also it can be seen that the unshared diffusion regions are connected to the respective bulk terminals. The output Metal1 lines run parallel to the nmos and pmos sections and are connected to each other towards the left of the layout through a vertical Metal1 line.The symmetry in the layout enables us to use this as the unit-cell for our p-cell of a nand gate. Fig. 13 shows the layout of a nX nor gate formed using n unit cells of the nor gate in parallel.

D. EXOR Gate

The unit cell of an EXOR gate is shown in Fig. 14. Unlike the layouts of the other cells proposed in this section, Metal2 has been used to realize the p-cell structure because of the complicated structure of the cell. The corresponding schematic of the cell is also shown in the figure.Metal2 lines running over the diffusion regions are used to make the connections between the inputs IN1 and IN2 across the unit cells.Also Metal2 has been used to connect the gates as well because of the complexity of the gate.The symmetry of the structure is evident from Fig. 15

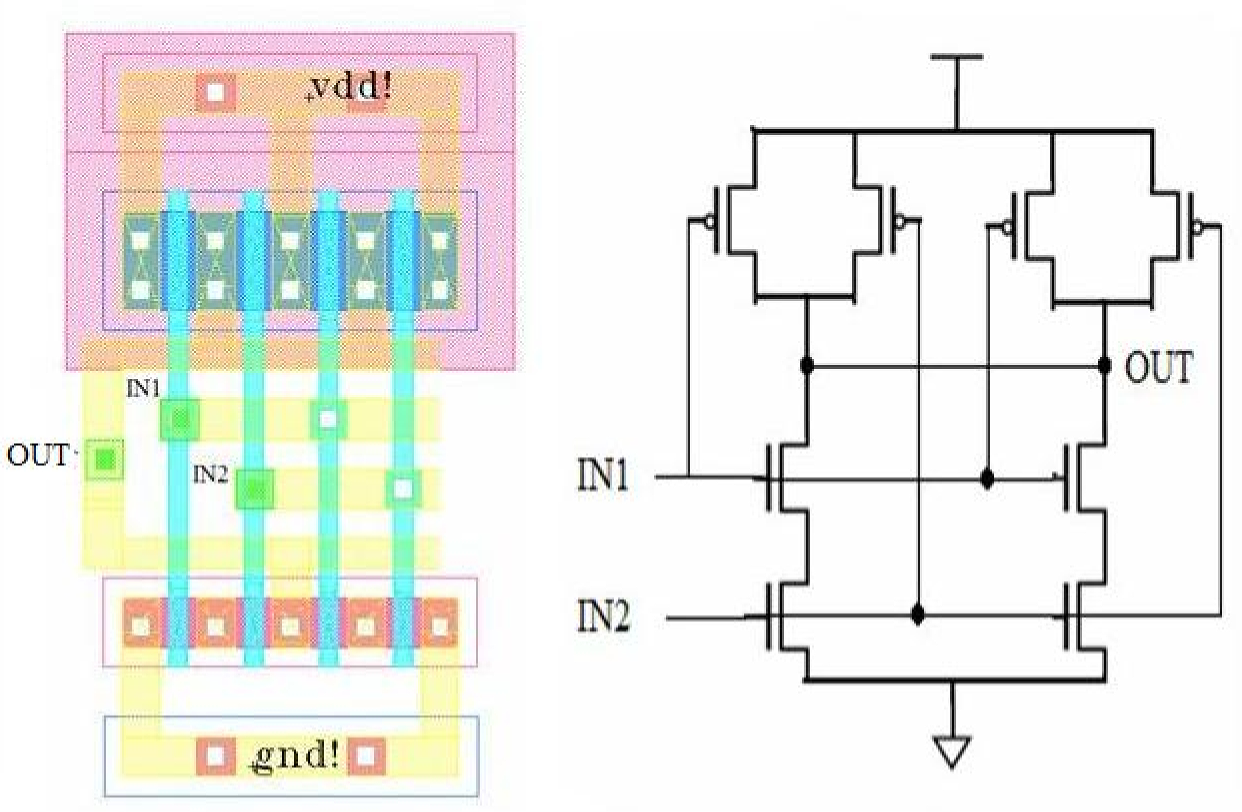


Fig. 10. Unit cell of a 2-input NAND gate

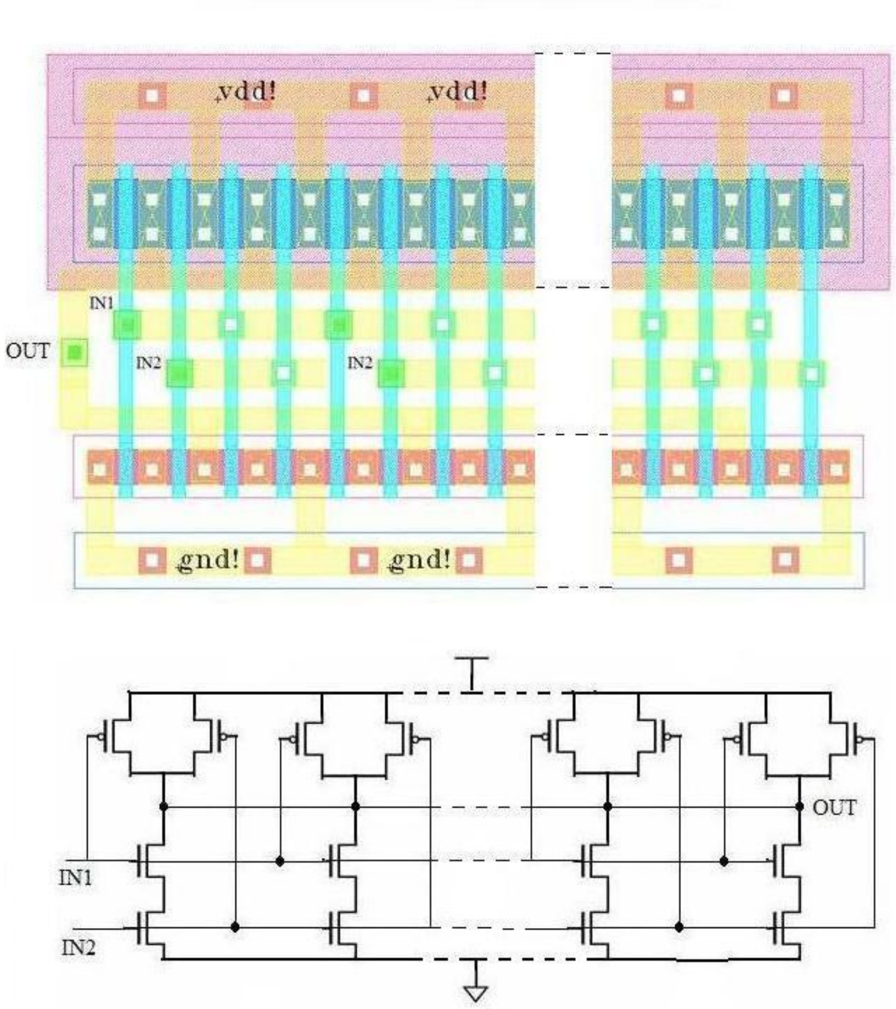


Fig. 11. The layout of a nX 2-input NAND gate

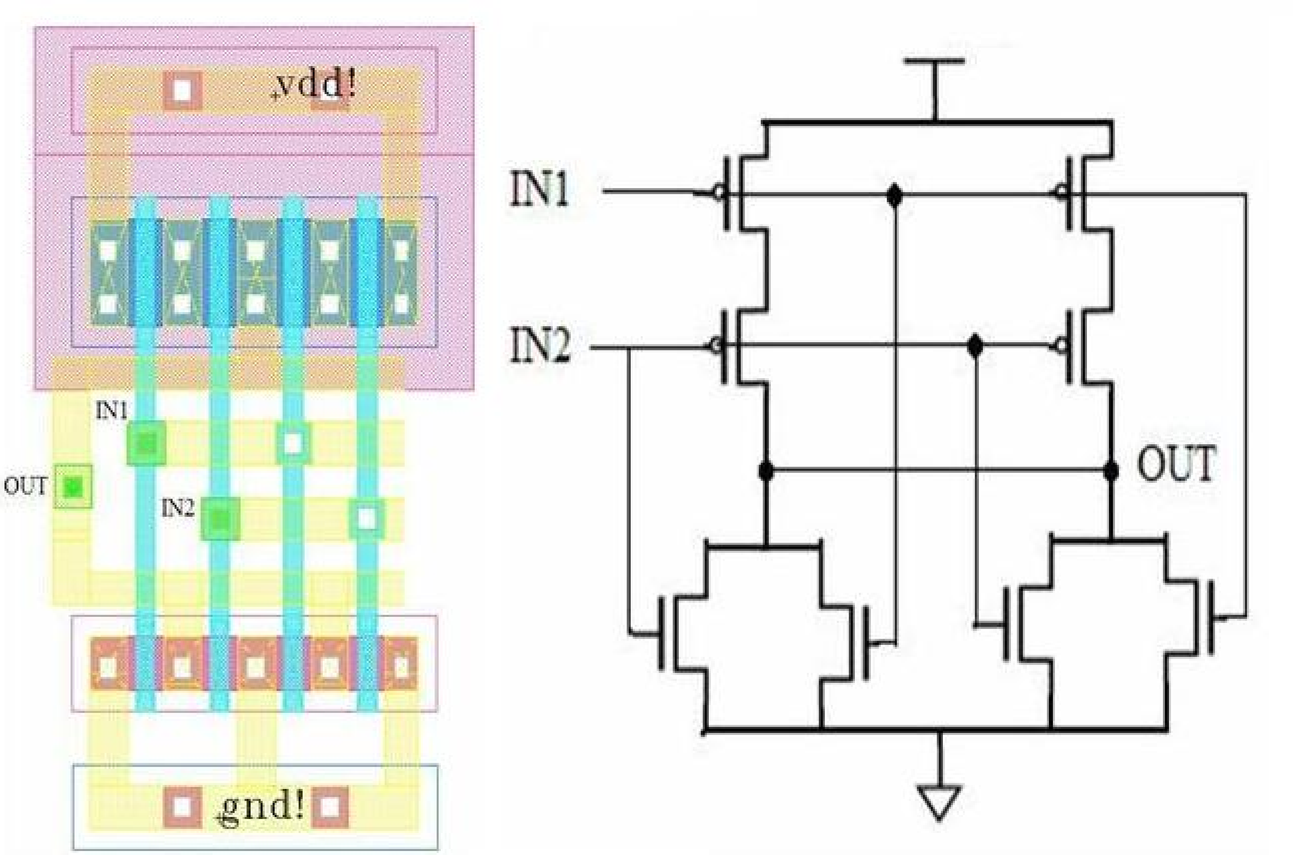


Fig. 12. Unit cell of a 2-input NOR gate

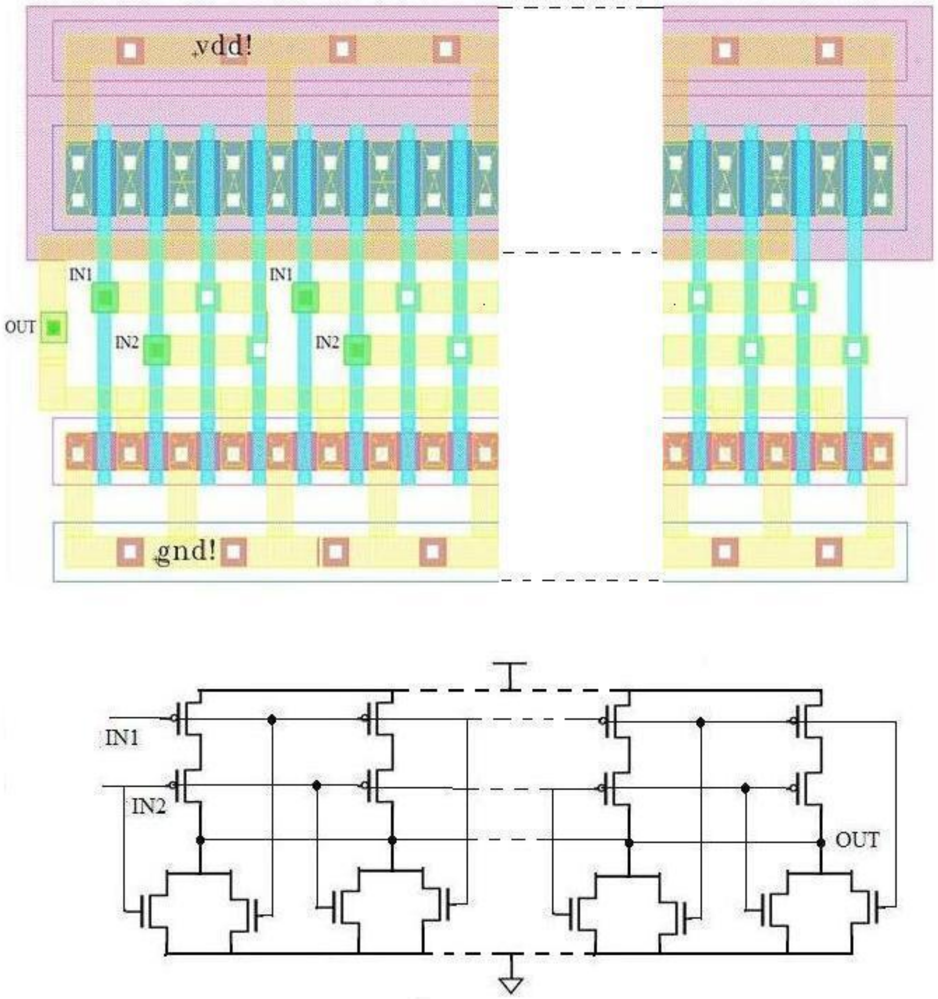


Fig. 13. The layout of a nX 2-input NOR gate

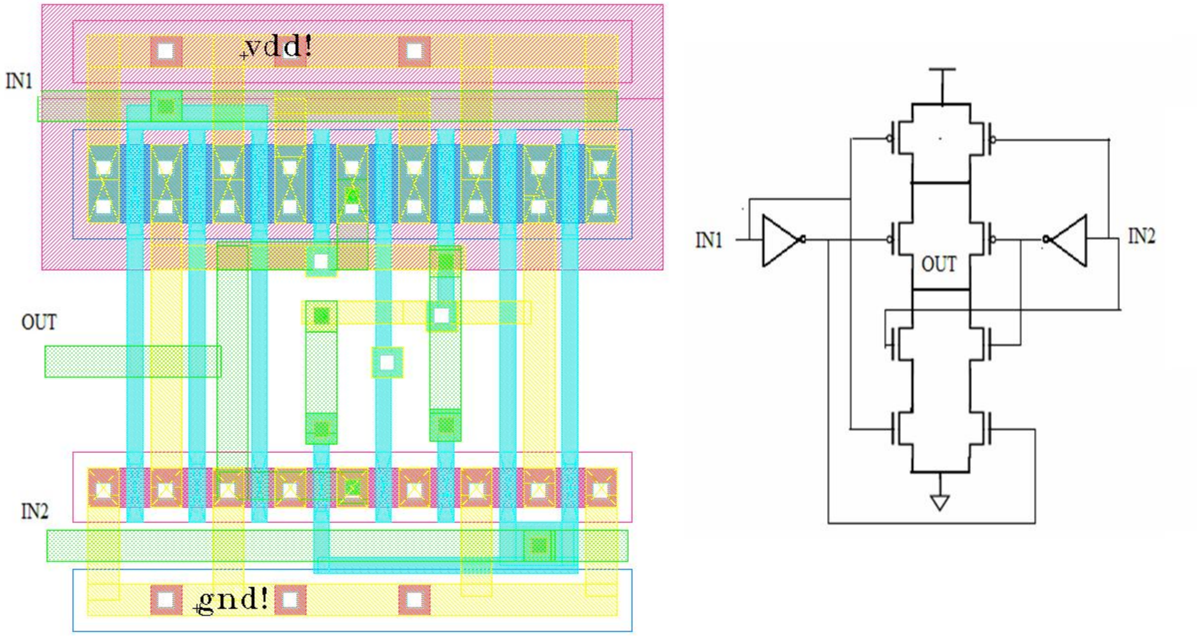


Fig. 14. Unit cell of a 2-input exor gate

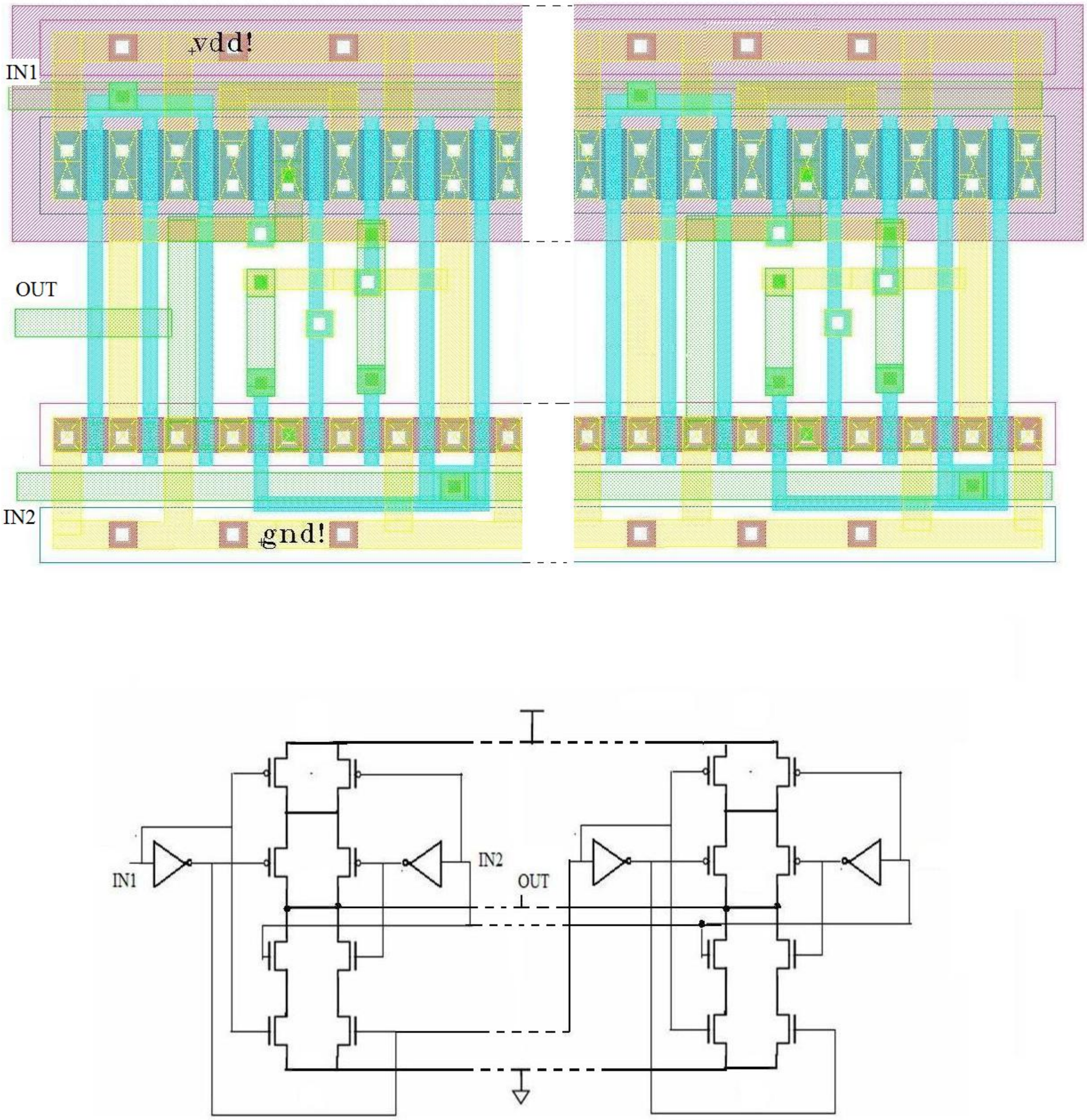


Fig. 15. The layout of a nX 2-input EXOR gate

CHAPTER IV

DELAY EVALUATION

Even though integer gate sizes can be realized using p-cell approach, this alone is not enough for the approach to be practically used. As mentioned in the first chapter, delay evaluation phase requires the delay tables of all the gates in the library. It is practically impossible to maintain the delay tables of all integer gate sizes. Hence there is a need for a very accurate delay model that can predict the delay table of any integer gate size using a single delay table. We propose the use of constant delay model to solve this problem. This delay model is 100% accurate under certain constraints. We first define the terms *unit cell* and *gate of size n* as below.

A. Definitions

Definition 1: A *unit cell* of a gate type G is defined as the gate of minimum size that can be practically used in the design and whose delay table can be used as the reference for the constant delay model.

Definition 2: A *gate of size n* or alternatively an *nX gate* or an *nX cell* of a gate type G is a gate that is implemented using n parallel unit cells of gate type G where n is a positive integer.

B. Constant delay model

In this section, we propose the constant delay model which can be used to represent the delay information of the p-cells. In general, under the constant delay model, the width of the gate increases linearly with the external load capacitance if the delay of the gate is to be kept constant as shown in the Fig. 16. In order to use this model as a practical representation of the gate delays, the width of the gate is not the only thing that should scale linearly to keep the delay constant. The parasitics at each and every node in the gate must scale linearly too. The simplest way of implementing such a configuration is to use the unit cells in parallel to realize a larger gate. In such a case, the constant delay model can be used as a practical representation of gate delays.

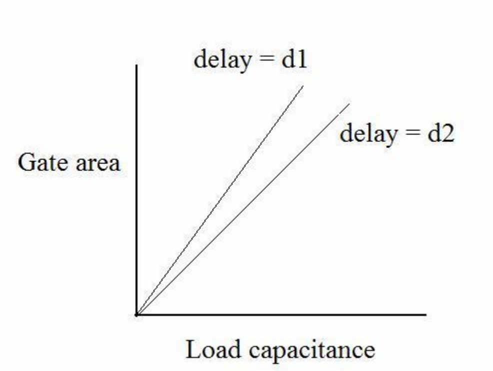


Fig. 16. Illustration of constant delay model

We first give the intuition behind the constant delay model. Consider an nX cell driving an effective load capacitance of C. Let the delay and output transition time of this gate for a given input transition time T_{in} be D and T_{out}. The nX cell is made of n identical unit cells in parallel and the capacitance C can be thought of n capacitances each of value C/n in parallel. By symmetry, we can conclude that the the delay of a unit cell driving a capacitance C/n is exactly equal to D. Similarly the output transition time of a unit cell driving a capacitance of C/n is equal to T_{out}. This fact is stated formally as Theorem 1.

Theorem 1:

Consider an nX gate of gate type G driving a load capacitance equal to C. For a

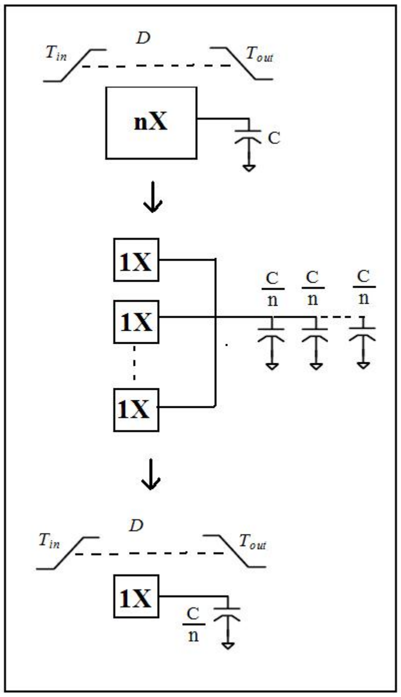


Fig. 17. Illustration of theorem 1.

given input transition time T_{in}, let the 50% propagation delay of the gate be $D(G, n, C, T_{in})$ and the output transition time be $T_{out}(G, n, C, T_{in})$. Then

$$D(G, n, C, T_{in}) = D(G, 1, C/n, T_{in}) \tag{4.1}$$

$$T_{out}(G, n, C, T_{in}) = T_{out}(G, 1, C/n, T_{in}) \tag{4.2}$$

proof:

The proof of this theorem for an inverter will be provided here for a step input case. The rise/fall delay is a function of C_{load}/k_n when the power supply and threshold voltages are unchanged [10] .Here C_{load} is the total load driven by the gate including its own parasitics. Thus C_{load} is a sum of C_{self_load} and C_{ext}. k_n is a parameter which is proportional to the width. When the size of the gate is increased by n times because of the definition of an nX gate, $C_{self_{load}}$ increases by *exactly* by n times. From the definition of k_n, this parameter too increases exactly by n times. Hence when C_{ext} is increased by n times, the delay remains the same.

The use of the constant-delay model to represent a cell library has been proposed in [9]. The authors state with experimental results that the gate area varies *almost exactly* linearly as the load changes so that the delay is kept constant. The possible reason for even the slight deviations from linearity in this work might be that only scaling of the gate areas are considered and not the details about the perimeters of diffusion regions. In our approach, because of the way we defined an nX gate, all the diffusion region areas and perimeters scale with the size of the gate and hence the relationship between the size of the gate (integers) and the load is *exactly* linear for a constant delay as shown in Fig. 18. The idea of invariance of C_{load}/k_n for *good* layout structures has also been suggested in this work. The p-cell layout structures proposed in this thesis satisfy such requirements.

Table I. Results for varying values of n and T_{in} for the example circuit

Input Slew ps	n	Rise Delay ps	Fall Delay ps	Rise Tout ps	Fall Tout ps
40	1	35.2	31.8	68.7	48.8
	5	35.2	31.8	68.7	48.8
	10	35.2	31.8	68.7	48.8
	20	35.2	31.8	68.7	48.8
	50	35.2	31.8	68.7	48.8
	100	35.2	31.8	68.7	48.8
80	1	42.6	39.9	75.4	54.7
	5	42.6	39.9	75.4	54.7
	10	42.6	39.9	75.4	54.7
	20	42.6	39.9	75.4	54.7
	50	42.6	39.9	75.4	54.7
	100	42.6	39.9	75.4	54.7

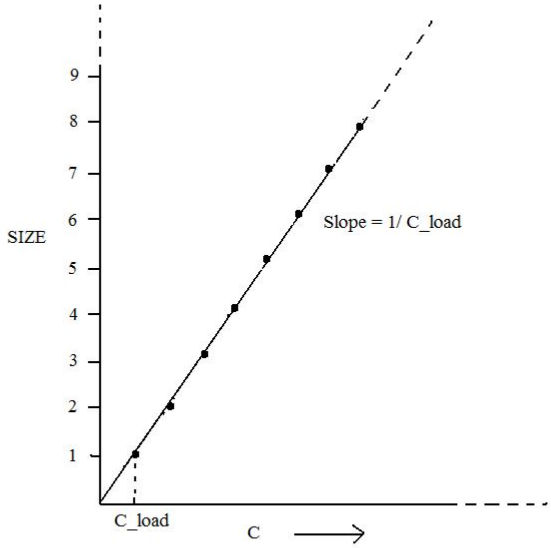


Fig. 18. A plot of size vs load cap is an exact straight line when plotted for integer values of size

As an example to illustrate the Theorem 1, let us consider the circuit in Fig. 19. Here in Fig. 19(a), a unit cell of an inverter drives another unit cell of an inverter through an interconnect load which has been represented as a pi-model. Fig. 19(b) shows an nX inverter driving another nX inverter through an interconnect whose width (interconnect width) is n times that of the interconnect in Fig. 19(a). It can be easily seen that the effective load capacitance seen by the driver in the case of Fig. 19(b) is n times the effective load capacitance seen by INV1 in Fig. 19(a). Thus, for a fixed input transition time, the output delay and output transition time have to be equal in both these cases. Table I shows these results for four different input slew rates and for varying values of n. The simulations were done using HSPICE using the netlists extracted from the layouts of the p-cell structures. However we have ignored the resistance of Metal1 during simulations. This is acceptable since the resistivity

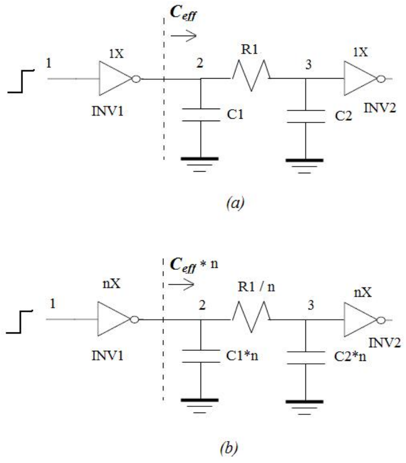


Fig. 19. An example circuit used to illustrate theorem 1.

of Metal1 is very low. The results confirm that for a fixed input transition time, the delay values and output transition times remain unchanged as n varies. Thus the delay tables of p-cell structures can be represented using the constant delay model. The constraints that the delay model imposed on the layouts of p-cell structures in order for them to be compatible with the model are discussed in the next chapter.

C. New approach for gate delay estimation

The consequence of the constant delay model is that, for timing analysis, the only data that we need to maintain for a given gate type is the delay table of a unit-cell of that gate type. From this data, the delay of a gate of any other size which is made of the unit cells can be accurately estimated. The method of gate delay estimation using the new delay model is illustrated in Fig. 20. After modeling the RC load as an effective capacitance, the delay table for 1X gate (unit cell) is looked up to compute the delay since the delay of an nX gate driving a load of C_{eff} is exactly equal to that of 1X gate driving a load capacitance of C_{eff}/n. Thus the constant delay model significantly reduces the burden of maintaining huge volumes of delay tables in the standard cell library.

In this chapter, we have shown using post layout simulations that the delay information of p-cells can be represented using the constant delay model. In the next chapter, we discuss the constraints that the constant delay model imposed on the layout structures. Our p-cell structures were made in such a way that they satisfied these constraints.

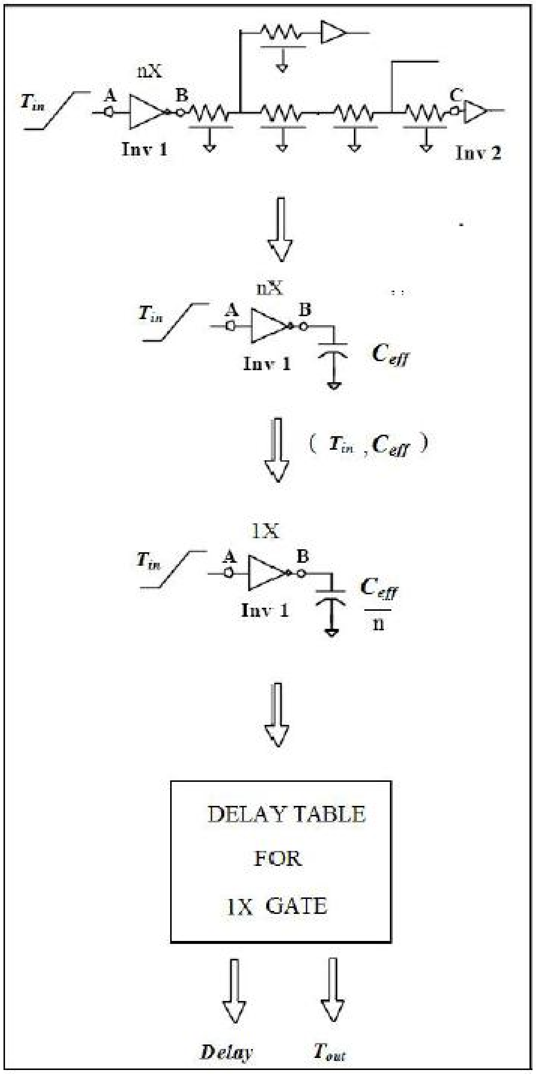


Fig. 20. Illustration of the method of determining the delay of a gate driving an arbitrary RC load using the unit-cell delay model.

CHAPTER V

CONSTRAINTS ON THE LAYOUTS OF P-CELLS

To layout the gates, we use the layout structure which contains a pmos section at the top and an nmos section at the bottom, a common layout style which was first proposed by Uehera and Van cleemput [6]. The metal1 VDD line runs on the top of the pmos section while the metal1 GND line runs at the bottom of the nmos section.

Our goal is to make sure the constant delay model gives a very accurate delay estimation at the layout level too in the presence of all parasitics. In other words, our goal is to make sure that our layouts are made such a way that even after extracting the netlist from the layout, the cell delays must obey the unit-cell delay model accurately enough for practical purposes. The present standard cell libraries were tested for compatibility with our delay model and have been found to give large deviations from the delay values predicted using the unit-cell delay model. The main reason for these deviations are that these layouts do not conform to the fundamental assumptions of our delay model. The constant delay model strongly relies on symmetry. It assumes that an nX cell is composed of n parallel unit cells. The layout structures must make sure that this basic requirement is met. In particular, for the layout structures to be compatible with our delay model, they have to satisfy two constraints. They are 1) The layout of an nX gate must comprise of n unit cells. The connections between poly fingers across the unit-cells must be made using Metal1 and 2) The diffusion regions of the transistors at the extremes (left and right) in the layout must be connected to the respective bulk terminals (VDD for pmos and GND for nmos).

The first constraint imposed by our delay model on the layout structure is that all the poly finger connections across unit-cells have to be made using metal1. This is because our delay model assumes that the input pins of the unit cells that make up a bigger cell are shorted. Using poly to make connections across the unit cells introduces significant parasitic resistance between the poly fingers. This causes the delay to be more than what is predicted by our delay model. Hence it is important to use Metal1 to connect the poly fingers.

The second constraint imposed on the layout structure by our delay model is that all the transistors must be laid out in such a way that the diffusion regions at the extremes are connected to the respective bulk terminals. To illustrate why this has to be so, consider Fig. 21(a) which is the layout of a minimum-sized inverter made in 0.18μm technology laid out using one poly finger. This layout is DRC-clean. Let us consider the possibility of using this cell as our unit cell for inverter. Also shown is the layout of a 2X inverter in Fig. 21(b) which is formed by two poly fingers. Following similar layout style, an nX inverter can be laid out using n poly fingers. This p-cell structure of an inverter is explained in detail later in the next chapter. The SPICE netlists of these two inverters were extracted and were used to drive loads of 1 fF and 2 fF respectively and the delays were measured. Following similar p-cell structure, the SPICE netlists of various sizes of inverters were extracted. Each of these inverters was used to drive a capacitance which is equal in femto farads to the size of the inverter. In other words an nX inverter drives an n femto farad capacitance in our experiment. The delay values (Rise delays are considered here. Fall delays and output transition times have been observed to follow the same trend) are shown in the table. II. If our choice of unit cell is correct , the delays have to be exactly equal in all the cases. However from the table we observe that for layouts made in even number of poly fingers, the delays are exactly equal. For the other layouts, they are significantly different.

Each of the nmos/pmos transistors formed by the poly fingers in these layouts had the same width and length as that of the nmos and pmos transistors of the unit

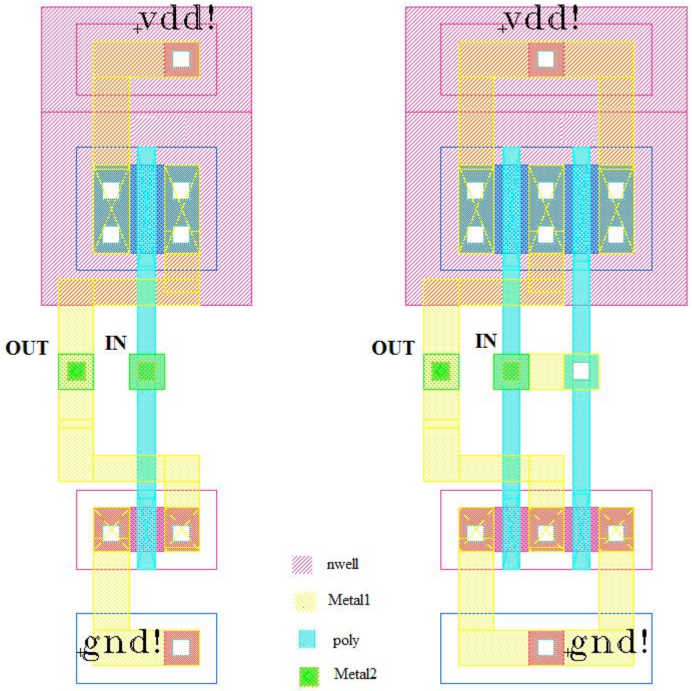


Fig. 21. Layout of 1X and 2X inverters assuming the unit cell(1X cell) is the inverter laid out with only one poly finger

cell. Hence we might expect all of the delays in the table II to be equal. But it is not so. This is because the diffusion areas and perimeters of the transistors at the extremes(left and right) are different from those of other transistors in the same section when the transistors are laid out in odd number of poly fingers. Consider Fig. 22 which shows the nmos section of a 5X inverter. The five nmos transistors are named M1, M2 , M3, M4 and M5 in the figure. The source and drain areas of each of these transistors have been listed in the table III. It can be clearly seen that the drain area and drain perimeter of the transistor M5 are different from those of M2, M3 and M4. The key assumption of our delay model is symmetry. Constant delay model assumes that all the transistors M1,M2,M3,M4 and M5 are exactly identical not just in their widths and lengths but also in their drain and source perimeters and

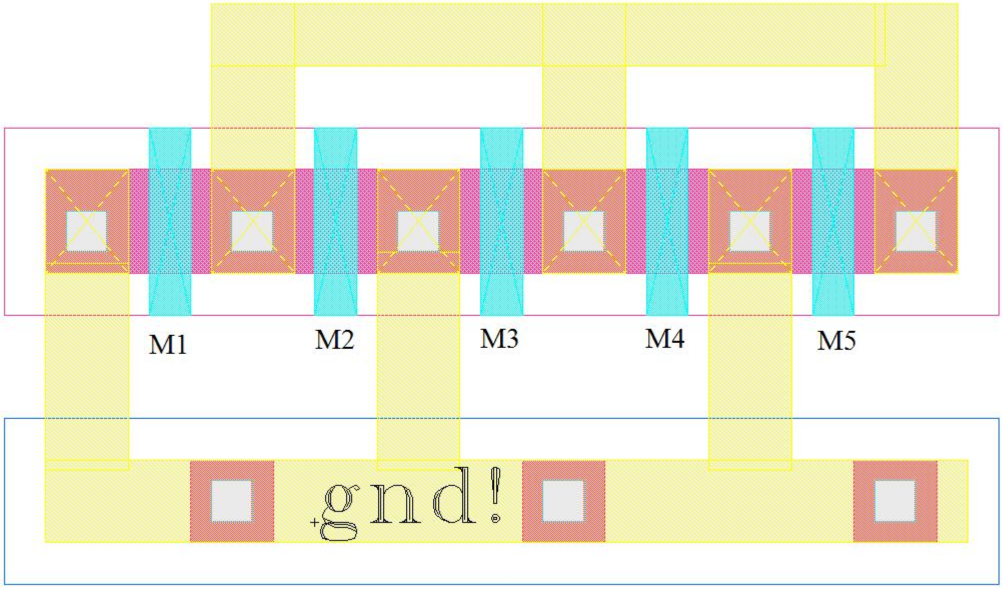


Fig. 22. Layout of 1X and 2X inverters assuming the unit cell(1X cell) is the inverter laid out with only one poly finger

areas. The layout clearly shows that M5 is not identical to the remaining transistors. Also M1 can be considered to be identical to M2,M3 and M4 though the source area and perimeter of M1 are different from the values for M2,M3 and M4 since this source terminal is connected to the bulk and hence there is no parasitic capacitance associated with this diffusion region.

The asymmetry of transistors is caused only in layout configurations in which the diffusion regions at the extremes are not connected to the bulk. Thus for the layouts to be compatible with our delay model, all the transistors must be laid out such that the diffusion regions at the extremes are connected to the bulk. Thus the unit cell of our inverter consists of two poly fingers as shown in Fig. 21(b). The possible inverter sizes that can be used are integer multiples of the size of this unit cell.

All the p-cell structures proposed in chapter III satisfied the constraints men-

Table II. Table showing the rise delays when the unit cell was chosen to be the minimum size possible in 0.18 μm technology

Driver	Load fF	Delay ps	Driver	Load fF	Delay ps
1X	1	38.275	2X	2	26.979
3X	3	30.756	4X	4	26.979
9X	9	29.912	10X	10	26.979
49X	49	28.153	50X	50	26.979

Table III. The source and drain parameters for the five transistors in fig. 22. M1,M2,M3 and M4 are identical to each other but M5 is not

Transistor	Source Area	Drain Area	Source Perimeter	Drain Perimeter
M1	0.2025	0.1215	1.35	0.54
M2	0.1215	0.1215	0.54	0.54
M3	0.1215	0.1215	0.54	0.54
M4	0.1215	0.1215	0.54	0.54
M5	0.1215	0.2025	0.54	1.35

tioned in this section. The poly finger connections across the unit cells are made using metal1. The diffusion regions at the extremes are connected to the bulk terminals. Hence our p-cell structures are compatible with the constant delay model.

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

In conclusion, this thesis proposes a practical approach to integer gate sizing. We propose two ways of implementing the integer gate sizing approach. The first approach is to use a combination of the geometrically spaced gate sizes to generate integer gate sizes. The second approach is to use a parameterized cell approach to dynamically generate the layouts of gates. We have shown that the p-cell approach is advantageous than the parallel cell approach in terms of area. One more advantage of using a p-cell approach is that only one layout structure needs to be maintained in the database. We also proposed the use of constant delay model to estimate the delay of a gate of any size from the delay of just a 1X gate. This delay model imposes some constraints on the layout structures. These constraints have been discussed and layouts of INVERTER, NAND, NOR and EXOR gates which satisfy these constraints have been shown.

A. Potential of integer gate sizing approach

A practical approach to integer gate sizing can significantly improve the quality of designs. In [4], the authors use a continuous solution guided approach to come up with a discrete gate sizing solution. From this work it can be seen that the continuous gate sizing solution gives an area savings of around 15% on an average for 10 different ISCAS benchmarks as shown in the table IV. Since the average device size is 15% smaller the total capacitance which is the sum of all gate and diffusion capacitances is 15% smaller for the case of continuous gate sizing. Since the dynamic power dissipation is directly proportional to the capacitance, the power savings is around 15%. Hence even for integer gate sizing we can expect the area and power savings to be around this percentage. The exact savings can be computed by implementing the algorithm proposed in [4] by giving integer gate sizes as the input. We also compare the p-cell approach in terms of area with the parallel cell approach which uses geometrically spaced sizes. This is shown in the plot in Fig. 23

Table IV. Table illustrating potential area savings of Integer gate sizing

Circuit	Timing constraint	Exponential Solution		Continuous solution		Area savings (%)
		Slack	Area	Slack	Area	
C432	1000	15	0.18	2	0.16	11.11
C499	1500	18	0.11	1	0.087	20.91
C880	1100	2	0.21	1	0.17	19.04
C1355	2200	8	0.5	1	0.45	10.00
C1908	1900	2	0.21	0	0.15	28.57
C2670	1800	3	0.23	2	0.2	13.04
C3540	3000	9	0.29	0	0.28	3.45
C5315	2250	21	1.7	2	1.49	12.35
C6288	5700	5	1.22	2	1.02	16.39
C7552	2100	10	0.82	0	0.65	20.73
						Avg = 14.98

B. Future work

A good representation of the power/area savings of integer gate sizing approach will be obtained if a simple circuit like a ring oscillator is implemented using the integer gate sizes as well as a standard commercial library so that the area and power of these two implementations can be compared. This can be taken up as the future work. In this thesis, we have focused on the layout information and timing information of standard cells. One more component of data that is associated with standard cells is power dissipation. We have actually observed that the power value scales linearly as the device size and the load capacitance scale linearly. For example the power dissipation of a nX gate driving a load of C is exactly equal to the power dissipation of a 1X gate driving a capacitance of C/n. In addition to power data, noise data too

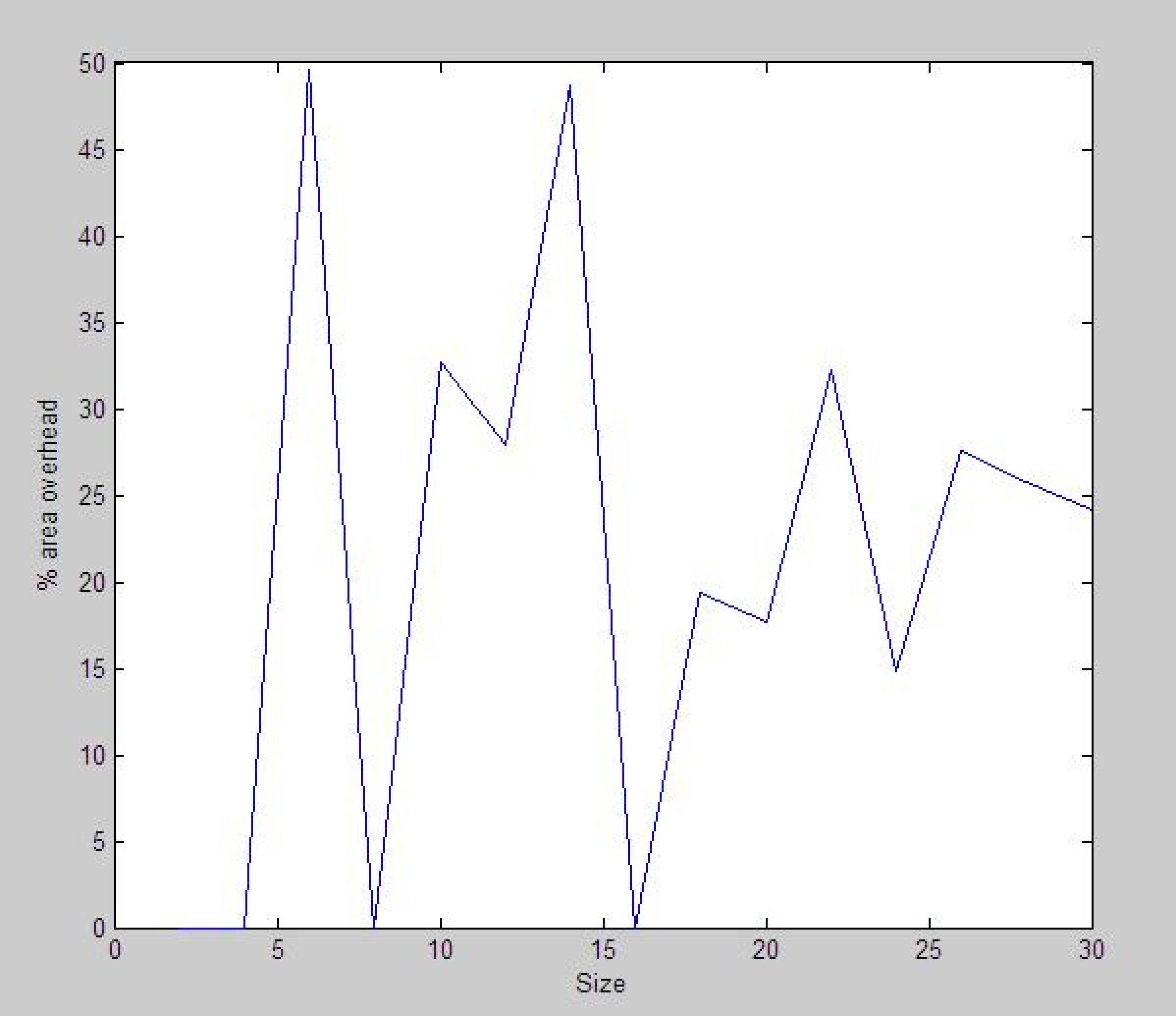


Fig. 23. Area overhead due to parallel cell approach

is stored in the cell library so that it can be used during noise analysis. The future work on this approach can be to come up with ways to represent the noise information of p-cells in addition to the delay/area information. Since the layout structures that have been proposed are very symmetric structures, we may expect the noise data too to follow a simple pattern. This is one aspect that has to be looked into. To construct p-cell structures for gates which are more complex than the ones considered in the thesis can be an interesting work for future.

REFERENCES

[1] K. Keutzer and K. Scott, “Improving cell libraires for synthesis,” in *Proc. International Workshop on Logic Synthesis*, 1993.

[2] J. Fishburn. and A. Dunlop, “TILOS: A polynomial programming approach to transistor sizing,” in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 1985, pp.326–328.

[3] F. Beeftink, P. Kudwa, D. Kung and L. Scott, “Gate-size selection for standard cell libraries,” in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 1987, pp. 545–550.

[4] S. Hu, M. Ketkar, and J. Hu, “Gate sizing for cell-library based designs,” in *Proc. ACM/IEEE Design Automation Conference*, 2007, pp.847-852.

[5] R. Haddad, L. van Ginneken, and N. Shenoy, “Drive selection for library design,” in *Proc. International Workshop on Logic Synthesis*, 1997.

[6] T. Uehara and W. M. Vancleemput, “Optimal layout of CMOS functional arrays,” in *Proc. ACM/IEEE Design Automation Conference*, 1978, pp 287–289.

[7] S. Abbaspour and M. Pedram “Calculating the effective capacitance for the RC interconnect in VDSM technology,” in *Proc. of the IEEE Asia and South Pacific Design Automation Conference*, 2003, pp. 43–48.

[8] Y. You, B. Roetcisoender, A. Cheng, R. Meghee and S. Sugiyama, “Peformance-driven layout through device sizing,” in *Proc. IEEE Custom Integrated Circuits Conference*, 1993, pp 9.3.1–9.3.4.

[9] J. Grodstein, E. Lehman, H. Harkness, B. Grundmann and Y. Watanabe, “A delay model for logic synthesis of continuously-sized networks,” in *Proc. IEEE/ACM International Conference on Computer-Aided Design*, 1995, pp. 458–462.

[10] S.M. Kang and Y. Leblebici, *CMOS Digital Integrated Circuits Analysis and Design.* 3rd ed., Boston : McGraw-Hill, 2003, pp.220–240.

VITA

Name : Uday Doddannagari.

Education : B.E.(Hons.) Electrical and Electronics Engineering,
Birla Institute of Technology and Science, Pilani, India.

Address : Department of Electrical and Computer Engineering,
c/o Prof. Weiping Shi,
Texas A&M University, M.S. 3128,
College Station, TX 77843.

The typist for this thesis was Uday Doddannagari.