

ABSTRACT

Title of Document: Integrating a Dual-Vth Design Flow Using Mixed Vth Cell Libraries in EDA Tool

Chandra Sekhar Nagarajan
Master of Science 2007

Directed By: Associate Professor, Gang Qu
Department of Electrical & Computer
Engineering

Leakage power has become one of the most important components of power dissipation in sub-micron designs. It is critical to include leakage mitigation methods in the low power design flow. Dual-Vth assignment, an effective leakage reduction technique, has been adopted by industrial EDA tools in gate-level design. This thesis presents a dual-Vth based low leakage design flow using a mixed Vth standard cell library, in which transistors within a cell have different Vth values. A mixed threshold standard cell design methodology is implemented and the cells are characterized using HSPICE with 130nm model. We integrated the design flow in a state-of-the-art industrial CAD tool and ran experiments on eleven ISCAS benchmarks and three industrial designs. On an average, our method can achieve 40% leakage saving over designs with nominal threshold voltage cells and 9% saving over the cell-based dual-Vth assignment technique, without any delay penalty.

Integrating Dual-Vth Design Flow Using Mixed Vth Cell Libraries in EDA Tool

By

Chandra Sekhar Nagarajan

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2007

Advisory Committee:
Professor Gang Qu , Chair/Advisor
Professor Bruce Jacob
Professor Martin Peckerar

© Copyright by
Chandra S Nagarajan
2007

Acknowledgement

First of all, I would like to thank my parents for lending their support throughout my stay at University of Maryland.

I am very grateful to my advisor, Dr.Gang Qu for his support and encouragement. It was a great experience to work with him. I am very thankful to Dr. Lin Yuan for his valuable insights into the research problem.

I would like to thank everyone at Atmel Corporation, Chesapeake Design Center, Columbia for giving me an opportunity to work as an intern for over a year. The work done in thesis is essentially the culmination of the work done at Atmel as part of my internship. Special thanks to Dean Uehara, Barbara Stamps, Pushkar Pulastya, Dan Meyer for the brain storming discussions and their support.

Special thanks to my friends – Bhuvan, Amit Apte, Ashish who have always inspired me to work harder and do things better.

I would also want to thank my friends Bargav, Rakesh, Jishnu, Raghuraman, Alankar, Rashi, Shitu, Sravya, Shraadha and Neha for being through my thick and thin and supporting me throughout to do things better.

Table of Contents

Acknowledgement	ii
Table of Contents	iii
List of Tables	v
List of Figures	vi
Chapter 1	1
Introduction and Motivation	1
Sources of Power dissipation	2
<i>Dynamic Power</i>	3
<i>Static Power</i>	3
Leakage Power Reduction Techniques	5
<i>Power Gating</i>	5
<i>Input Vector Control</i>	6
<i>Threshold Voltage Manipulation</i>	7
Chapter 2: Design Flow	10
ASIC Design Flow	10
Power-Driven Design Flow	12
Chapter 3: Related Work on Dual-Vth Design Techniques.....	14
Gate Level Dual-Vth Design	14
Transistor Level Dual-Vth Design Methodology	15
Chapter 4: Design Methodology of Mixed-Vth Cells.....	18
Introduction to Mixed Vth Design	18
Low-Leakage Cell Variant Design	19
Design Methodology Using Mixed Vth Cells	22
<i>Low-Leakage MVT Cell Variant Creation</i>	22
<i>Mixed Vth Cell Library Creation</i>	23
<i>Dual-Vth Design Flow Using Mixed Vth Library</i>	23
Chapter 5: Implementation	26
Mixed Vth Library Characterization.....	26
<i>HSPICE Simulation Setup</i>	26
<i>HSPICE Timing Characterization</i>	27
<i>Leakage Characterization</i>	27
Synthesis Flow	27
Chapter 6: Results.....	30
Leakage Reduction.....	30
MVT Cell Usage Statistics.....	31
Total Negative Slack Statistics	32
Chapter 7: Conclusion and Future Work	34
Appendices.....	37
Bibliography	38

List of Tables

Table 1.1:	The leakage current values for 2-input NAND gate.....	6
Table 4.1:	Timing arcs in a standard cell nd02d1 with 1x drive strength from a 130nm.....	20
Table 4.2:	Timing arcs in a standard cell, with MVT variant nd02d1 with 1x drive strength from a 130nm cell.....	20
Table 6.1:	Leakage Reduction.....	31
Table 6.2:	MVT Cell Usage.....	32
Table 6.3:	Total Negative Statistics.....	33
Table 7.1:	Delay values for a 1X NAND2 cell with transistor sizing.....	35
Table 7.2:	Sizing and Leakage for various transistor configurations.....	35

List of Figures

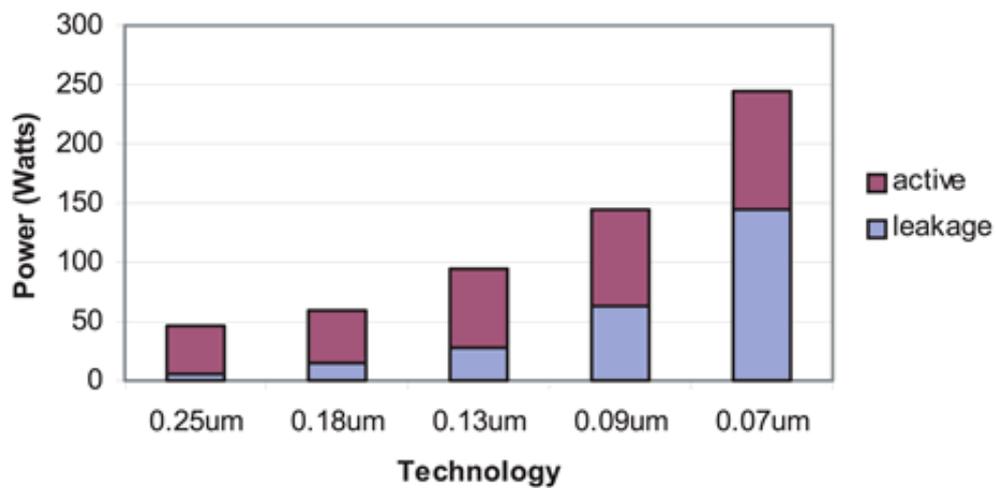
Figure 1.1:	Active/Leakage Power.....	1
Figure 1.2:	Power density Vs leakage and Active power.....	2
Figure 1.3:	Sources of leakage current in NMOS Transistor.....	4
Figure 1.4:	Power Gating Circuit.....	6
Figure 2.1:	ASIC Design Flow.....	10
Figure 2.2:	Power Aware Design Flow.....	12
Figure 4.1:	Schematic View of an NAND2 LVT Cell.....	20
Figure 4.2:	Schematic View of an NAND2 MVT Cell.....	21
Figure 4.3:	Dual Vth Design Flow Using Mixed Vth Cells.....	25
Figure 5.1:	Generic Synthesis flow using Design Compiler.....	29
Figure 7.1:	Schematic View of NAND2 Cell	35

Chapter 1

Introduction and Motivation

With the advent of many wireless and portable applications, power has become one of the most critical design constraints for designers. As the technology scales down, the leakage power contribution to the total power has been increasing [Fig 1.1]. It contributes to about 18% in 130nm and is expected to contribute upto 50% of the total power at 65nm technology [14].

Figure 1.1: Active/Leakage Power

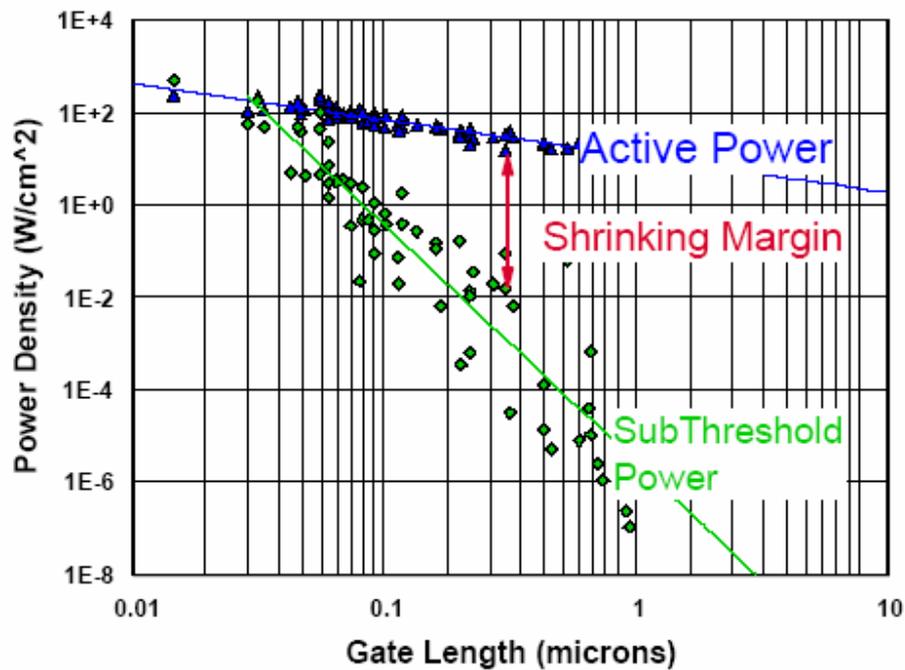


Source: Microprocessor power consumption, Intel

As shown in Figure 1.2, the margin between the active and sub threshold power has been shrinking as gate lengths decrease. The contribution of leakage or the static power calls for new design methodologies with leakage power as an important criterion. As technology scales down, the supply voltage decreases and the threshold voltage needs to be reduced accordingly in order to maintain performance. However, this results in an exponential increase in sub threshold leakage. Hence leakage aware designs are becoming increasingly common in the current sub micron era.

Low power design techniques could be applied to various levels of abstraction, right from the architecture level, all the way down to the device level. To get the best possible design optimized for power, we should resort to a holistic power minimization approach that optimizes all components of power. There are other sources of power dissipation as described in the following section.

Figure 1.2: Power density Vs leakage and Active power [17]



Sources of Power dissipation

There are three main sources of power dissipation in CMOS Circuits and they can be categorized as

1. Switching power ($P_{\text{switching}}$)
2. Internal Power(P_{Internal})
3. Leakage Power (P_{leakage})

$$P_{\text{total}} = P_{\text{switching}} + P_{\text{Internal}} + P_{\text{leakage}}$$

The main components of total power depend on the operating mode of the circuit.

Dynamic Power

Dynamic power ($P_{\text{switching}} + P_{\text{Internal}}$) is the dominant component in active mode. Switching power is due to charging and discharging of load of capacitances. In static CMOS circuits, when

$$V_{Tn} < V_{in} < V_{dd} - |V_{Tp}|$$

where V_{in} is the voltage changing at one input of the gate, while the voltages at other inputs remain steady, the pull up and pull down network conduct simultaneously and a short circuit path exist for direct current flow from VDD to GND . The power dissipated due to the short circuit current flowing through VDD to GND is known as the short circuit power or the internal power dissipated. The sum of switching and short circuit power is the dynamic power dissipated in the circuit given by the following equation.

$$P_{dyn} = \frac{1}{2} (f) \left[(V_{dd}^2) \left(\sum_i (\alpha_i C_{L_i}) \right) + V_{dd} \sum_i \sum_j (\alpha_i C_{ij} V_{ij}) \right]$$

Static Power

The second major component of power is the leakage power which is mainly due to leakage current in active and standby current in the circuit. The major sources of leakage current are

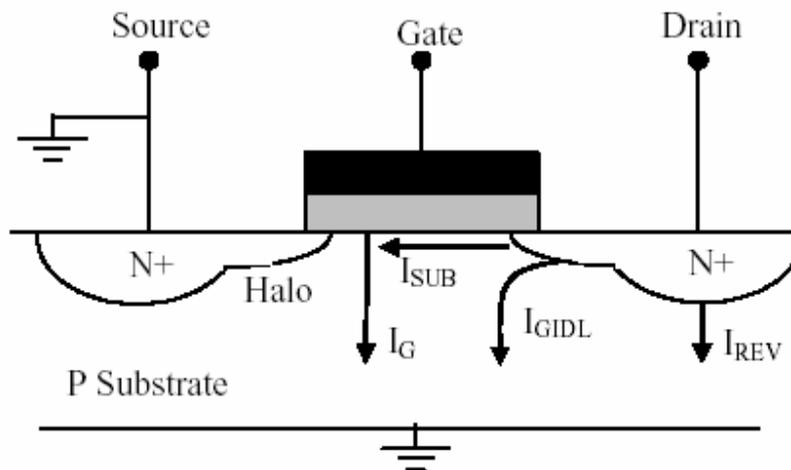
1. Sub threshold Leakage (I_{SUB}): The current which flows from the drain to the source of a transistor operating in the weak inversion region. Leakage power can be expressed mathematically by the equation

$$I_{sub} = K \left(1 - e^{-\frac{V_{DS}}{V_{th}}} \right) e^{\left(V_{GS} - V_{th} + \frac{\eta V_{DS}}{n V_{th}} \right)}$$

As seen from the above equation, sub threshold current has an exponential relationship with the V_{th} -threshold voltage. Decreasing the threshold voltage increases the sub threshold current exponentially and vice versa.

2. Gate Leakage (I_G): The current that flows directly from the gate through the oxide to the substrate due to gate oxide tunneling and hot carrier injection.
3. Gate Induced Drain Leakage (I_{GIDL}): The current which flows from the drain to the substrate induced by a high field effect in the MOSFET drain caused by a high VDG.
4. Reverse Bias Junction Leakage (I_{REV}): The current caused by minority carrier drift and generation of electron/hole pairs in depletion regions.

Figure 1.3 Sources of leakage current in NMOS Transistor



The exponential nature of leakage further calls for efficient techniques for low power design of standard cells with leakage as one of the design criterions.

Leakage Power Reduction Techniques

Leakage reduction techniques can be grouped into two main groups depending whether they reduce the standby leakage or runtime leakage [4]. Standby leakage methods are methods that reduce leakage of devices that are idle. On the other hand, runtime leakage reduction technique minimizes leakage power of devices in active mode. Traditionally, runtime leakage power has been less of a concern than standby leakage since dynamic power dissipation has been the main contributor to power dissipation during the active mode. Some of the frequently used leakage reduction techniques are described below.

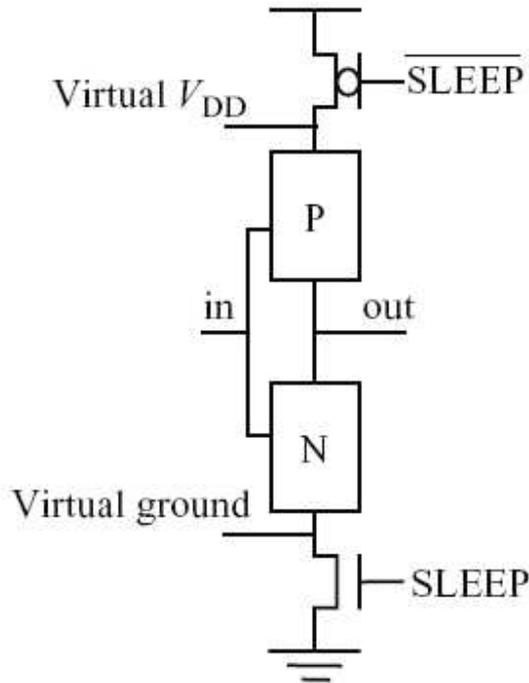
Power Gating

One of the ways to reduce leakage in standby mode is to turn off the voltage such that the idle parts of the circuit don't dissipate power. This is achieved by using one PMOS and one NMOS transistor in series with each logical block as shown in the Figure 1.4

The NMOS and PMOS sleep transistors create a Virtual VDD and Virtual GND. When the circuit is in active mode, the sleep transistors are ON and the circuits function normally. When the circuit is in idle mode, the sleep transistors are turned off and the gate is disconnected from the power and ground. In practice, only one sleep transistor is necessary and NMOS transistors are used as they have a lower on-resistance.

In order to make power gating effecting in reducing the leakage power during standby mode, the threshold voltage of sleep transistor must be high. To guarantee the proper functionality of the circuit, the sleep transistor has to be carefully sized to decrease its voltage drop while it is on as a result of having a high threshold voltage [2].The area overhead and dynamic power consumption increases with size of transistors. Therefore, it is hard to achieve total power dissipation reduction for devices with short idle periods.

Figure 1.4: Power Gating Circuit



Input Vector Control

It is found that leakage current of a gate is a strong function of its input combination [1]. The main reason for this dependency is that the input vector values affect the number of off transistors in both NMOS and PMOS networks of the gates. Table 1.1 shows the difference leakage value for all input combinations of a 2-input NAND gate [30] built in a 0.18 μ m technology with 0.2V threshold voltage and a 1.5V supply voltage.

Table 1.1: The leakage current values for 2-input NAND gate

Input		Leakage Current (nA)
A	B	
0	0	23.06
0	1	51.42
1	0	47.15
1	1	82.94

As it can be seen from the above table, the leakage current strongly depends on the input vector. The maximum leakage is about three times the minimum leakage. Therefore, the goal is to find the input combination that maximizes the number of off transistors or minimizes the leakage current in all stacks across the circuit. Once the input vector with minimum leakage current, MLV, is determined, we can switch the input of the circuit to this minimum input combination when it is idle to reduce the standby power consumption. In [32] it's shown that the MLV problem is NP-complete and the authors also propose a gate replacement technique which can be combined with input vector control to significantly improve the leakage reduction and reduce the run-time complexity.

Threshold Voltage Manipulation

A device with high threshold dissipates less leakage power than a device with low threshold voltage. The relationship of threshold voltage with delay and leakage of a gate could be used to reduce leakage effectively in a circuit. There are many methods that exploit the relationship between leakage, delay and threshold.

Variable Threshold CMOS is one of the methods where in threshold voltage of a device can be controlled by the substrate bias. Hence, we could have a high threshold during the stand-by mode while having a low threshold during the active mode [8]. Thus the off-current is very low due to high- V_{th} devices, the on-current can be changed accordingly in the active mode. A self-substrate bias circuit is used to control the body bias to get different voltages while operating in different modes.

Threshold voltage of devices could also be changed using multiple channel lengths [31]. Longer transistor lengths are used to achieve high-threshold devices but they also tend to increase the gate capacitance which in turn increases the dynamic power dissipation. Compared to multiple threshold voltages, this method has similar or lower process cost and

has lower process complexity as multiple masks aren't required as in case of multiple threshold processes.

Dual-Vth technique is one of the alternate ways to reduce leakage where in there are two flavors of every possible gate [7] – a High threshold cell (HVT) and a Low threshold Cell (LVT). Having a design with all LVT cells would make the design faster but leakier. However, at this point in time, we propose a methodology to better optimize leakage. Not all cells in the design are in the critical path. Hence we could make our critical paths faster and meet the timing constraint while recovering leakage from the non-critical paths. The non-critical paths could in theory contain as many low leakage cells as possible. This algorithm is implemented in many of the state-of-the-art EDA tools such as Synopsys Design Compiler. Thus, given two libraries, HVT and LVT standard cell libraries, we could use both the libraries appropriately and save on leakage without violating the timing constraint.

Transistor level Dual-Vth design is another further towards leakage minimization. It has been previously proposed in [10, 20, 24]. The motivation for the methodology comes from the limits of well dual threshold flows. In each of the HVT and LVT libraries, it should be observed that all the transistors belonging to every cell/gate have high threshold or low threshold. For example, a simple NAND2 cell consists of two NMOS and two PMOS transistors. In the HVT version of this cell, both NMOS transistors use NMOS high Vth devices and both PMOS use PMOS high Vth devices. The same analogy holds well in case of the LVT version of this cell. However, the four different transistors could potentially have different threshold configurations (high/low). This would further reduce the leakage power of each cell. The contribution of the work is to implement a Mixed-Vth library under practical industrial environment and develop a methodology to integrate it in the design flow of an ASIC.

The goal of this research work is to build a methodology to reduce static leakage power in sub-micron designs. Leakage power is being reduced by making use of a transistor

level multi-V_{th} approach. First step towards building the methodology is to build a library with multiple thresholds in the transistor level. The second step is to build a design flow so as to integrate it in the existing design flow of an ASIC and reduce leakage effectively.

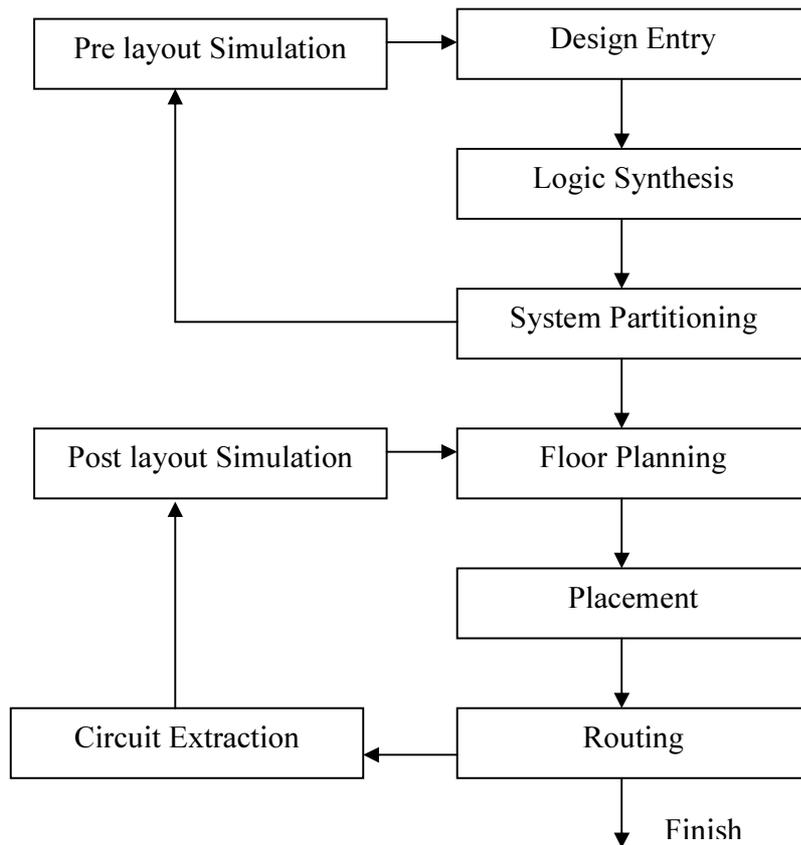
The contribution of the current work comes from the fact of giving designers a new library to play with early in the synthesis phase of the design so as to get the maximum savings in terms of leakage power without any trade-offs in delay. An accurate characterization of each multi-V_{th} cell for various load and slew ranges also makes sure results are accurate enough for analysis. Practical design methodology is proposed for single and multi stage cells considering the leakage and timing characteristics of each multi-V_{th} cell. Seamless integration into the existing design flow makes it very much practical and applicable to real design.

Chapter 2: Design Flow

ASIC Design Flow

Applications Specific Integrated Circuits or ASICs are, as the name indicates, non-standard integrated circuits that have been designed for a specific use or application. Generally an ASIC will be designed for a product that will have a large production run, and it may contain a very large part of the electronics needed on a single integrated circuit. As estimated, the cost of an ASIC design is high, and therefore they tend to be reserved for high volume products.

Figure 2.1: ASIC Design Flow [28]



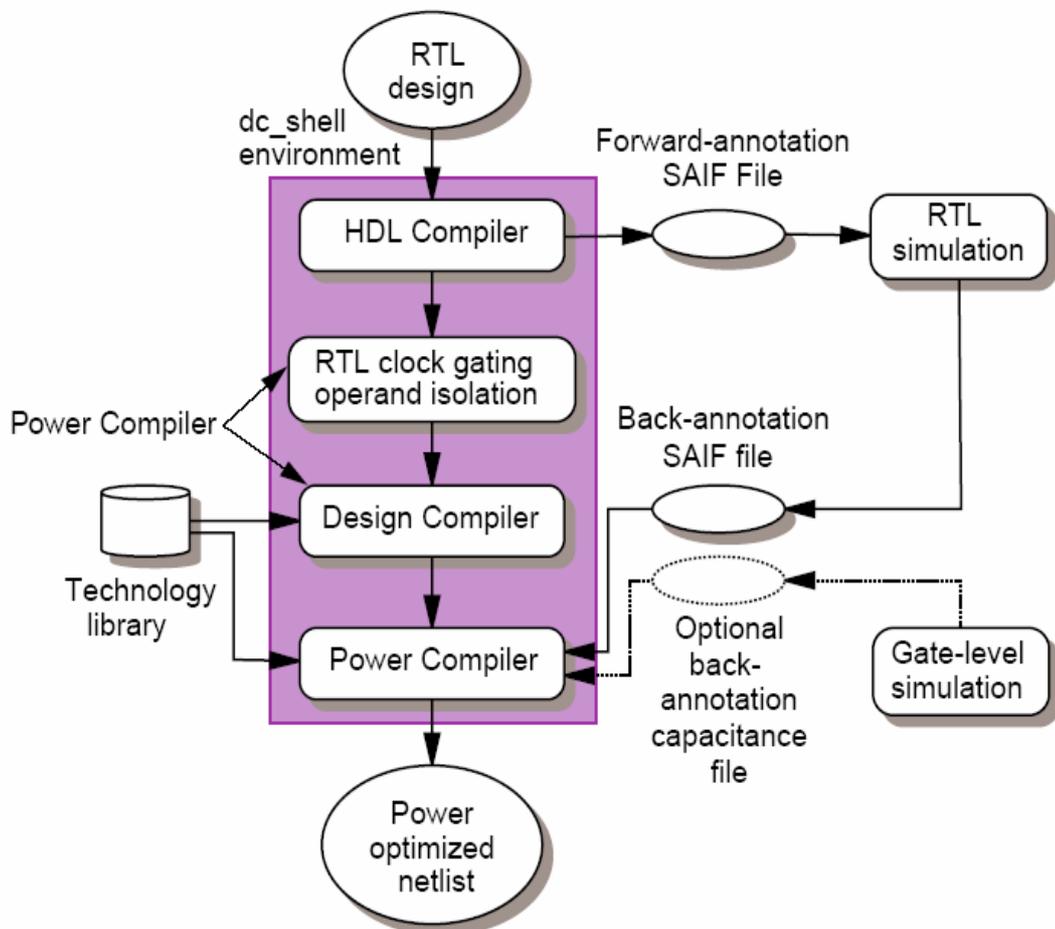
Despite the cost of the design of an ASIC, they can be very cost effective for many applications where volumes are high. It is possible to tailor the design of the ASIC to meet the exact requirement for the product and using an ASIC can mean that much of the overall design can be contained in one integrated circuit and the number of additional components can be significantly reduced. As a result they are widely used in high volume products like cell phones or other similar applications, often for consumer products where volumes are higher, or for business products that are widely used.

Fig. 2.1 describes basic ASIC based design flow. Design entry refers to the description of the intended digital design in a hardware description language. (Verilog/VHDL). Logic Synthesis is a process by which an abstract form of desired circuit behavior (typically register transfer level (RTL) or behavioral) is turned into a design implementation in terms of logic gates. System Partitioning deals with dividing an ASIC into modules. Pre layout simulation is done using the RTL to check if the design meets the functional specification. Floor planning refers to the arrangement of blocks of netlist on the chip. Placement decides the location of cells in a block. Routing make the connections between cells and blocks. Extraction determines the resistance and capacitance of the interconnect. Post layout simulations refer to the design validation after taking into account real wire and interconnect delays. It should be noted that this flow is normally iterated many times to meet the design requirements on timing, area and power.

Power-Driven Design Flow

Power is one of the most critical design constraints of modern SOC design and hence power optimization techniques have been used in every design level. For example, Fig. 2.2 describes the power aware design flow at register transfer level (RTL) that takes a RTL design specification and produces power optimized netlist. Design Compiler and Power Compiler are state-of-the-art synthesis tools from Synopsys Inc, a leading EDA company.

Figure 2.2: Power Aware Design Flow [27]



The above figure depicts the power flow used by power compiler. Power optimization can be done at different levels of abstraction. RTL clock gating, operand isolation are architectural

techniques used for power optimization. The accuracy of power calculations depends on the accuracy of the switching activity annotated to the design. Switching activity is calculated using RTL or gate-level simulation and stored in a SAIF file. Forward SAIF files, which are generated by Power Compiler, and contain any information required by HDL simulation to generate the required switching activity. Backward SAIF files, which are generated by HDL simulation, are the actual switching activity information used by Power Compiler to calculate power values.

Technology library is a key part of ASIC design. Most ASIC vendors supply a cell library that conforms to the design rules of the vendor. Every design is synthesized for the given cell library. Different sources of power dissipation can be optimized by various low power techniques. For example, clock gating and operand isolation reduce the dynamic power dissipation. Using multiple threshold cell libraries is one of the most effective leakage optimization techniques. It refers to having two cell libraries for synthesis: one with High threshold voltage (HVT) and the other with Low threshold voltage (LVT). The tradeoff is that HVT cells are slower but have low leakage, while LVT cells are faster but have high leakage. Dual-V_{th} design uses both libraries to minimize leakage under the given timing constraint by using LVT cells on critical path and HVT cells on non critical path.

Although research and practice on cell level dual-V_{th} design abound, there is little work reported on transistor level dual-V_{th} design, especially how to integrate it into existing EDA tools. This work fills this gap by building a mixed threshold library (MVT) where the V_{th} may be different for transistors within the same library cell. Such library will have timing and power characteristics between LVT and HVT libraries. We further propose a power aware design flow to support the design for leakage power optimization using such MVT library.

Chapter 3: Related Work on Dual-Vth Design Techniques

Gate Level Dual-Vth Design

The leakage in a transistor is mainly due to the sub threshold leakage current. The sub threshold current is the weak inversion current between the drain and source when the gate level is less than the threshold voltage. Due to exponential relationship between sub threshold current and V_{th} , an increase in V_{th} can result in orders of magnitude of leakage reduction. However, this also impacts the delay of the gate/cell. Specifically, the gate delay increases with the increase of V_{th} according to the following equation.

$$T_{pd} \propto \frac{CV_{dd}}{(V_{dd} - V_{th})^\alpha}$$

where V_{dd} is the supply voltage, C is capacitance, and α is a technology dependent constant between 1 and 2.

Dual V_{th} essentially means having two sets of NMOS and PMOS thresholds for every gate/cell. A *cell* is the fundamental logic block and is the basic element in the gate level netlist. In a high V_{th} cell, all transistors within a single cell are of high threshold and same analogy holds good for a low V_{th} cell. A high V_{th} cell in general is slow and has very low leakage. However, a low threshold cell has very high leakage but is much faster. Gate level dual- V_{th} refers to having same threshold for all the transistors within one gate/cell.

Dual- V_{th} algorithms have been extensively studied and addressed in [18-25]. In gate level Dual- V_{th} methodology, logic gates on timing critical paths are assigned low- V_{th} value, which means they will be implemented using low- V_{th} library cells; while gates on the non-critical paths are assigned high- V_{th} values, as long as timing is not violated, to save leakage [22].

In [23] authors propose a breadth first search based algorithm for dual- V_{th} assignment. The circuit is modeled as a directed acyclic graph (DAG) where in nodes of the graph represent

the gates and the edges represent the connectivity between different gates. The circuit is broken down in levels and all the gates are assigned low- V_{th} initially. The authors assume that each primary output of the circuit has one fan-in. Further, the algorithm works by first finding the nodes on the non-critical path and flipping some of the nodes in the non-critical path to high- V_{th} without violating the delay constraint. Finally, the optimal high threshold voltage corresponding to the best savings in standby leakage power is searched. This is done by linear search; it performs the assignment for various V_{th} values less than $0.5 V_{dd}$ and compares the leakage savings corresponding to each value of V_{th} . The optimal high V_{th} value is then set to be the V_{th} with the most saving in standby leakage power. An iterative max-cut algorithm is proposed in [22] for dual- V_{th} optimization. A maximal subset of feasible nodes for high- V_{th} assignment is determined while satisfying the circuit delay constraint. In order to determine the feasibility of a node, a threshold value for each gate type, T_{gate_type} is calculated. T_{gate_type} is defined as the difference between the propagation delays of a gate when assigned high and low V_{th} . The slack associated with each gate is calculated as the difference of arrival and required times. If the slack calculated is equal or higher than the T_{gate_type} , the node is called feasible and is assigned a positive weight that is equal to the reduction in standby leakage power. Maximum reduction of power leakage is obtained when the threshold voltage of all nodes in the feasible set is changed from low to high V_{th} . The algorithm finds the largest set of feasible nodes such that changing threshold to high V_{th} does not violate the timing constraint. The work proposed in [18-25], in some form make use of the available slack for leakage minimization thereby not incurring delay penalty.

Transistor Level Dual- V_{th} Design Methodology

In the above gate level Dual- V_{th} approach, all the transistors in a cell are assigned the same threshold voltage, either low V_{th} or high V_{th} . However, as we will show in the next chapter, not all the timing arcs in a cell are critical and this opens opportunity to further reduce

leakage by using high V_{th} values on transistors off the critical arcs in the cell. This technique is referred to as *transistor level dual- V_{th} design*.

Transistor level Dual- V_{th} Design has been previously studied in [4, 10, 20, 24]. Cells with different V_{th} at transistor level form a mixed V_{th} cell library. In [24], the authors propose a methodology to run transistor level static timing analysis. Approximate expressions for delay are used for calculation and which ignore impact of switching time of both NMOS and PMOS transistors on rise and fall transition. The propagation delay of each variant is estimated by an analytical expression. This is impractical for industrial designs where the design libraries normally contain hundreds of standard cells and require accurate delay data.

In [20] a sensitivity based upsizing approach is proposed that begins with assigning nominal- V_{th} to all transistors and then iteratively assigns low V_{th} to transistors based on the sizing and V_{th} assignment information. An enumeration based approach is presented in [10] for transistor level threshold assignment. This approach however grows quickly in space and time for given input size and hence is not practical under industrial circumstances. A practical approach is proposed in [4] to design and characterize the mixed V_{th} cell library. In that approach, leakage efficient mixed V_{th} cells variants are applied on netlist after the gate-level synthesis. An algorithm has been developed to replace low V_{th} cells with mixed V_{th} cells. The algorithm runs for every cell instance in the final netlist. A sensitivity factor is calculated as the ratio of increase in slack and reduction in leakage. Based on the sensitivities of each timing arc, a decision is made as to whether the cell should be replaced by the mixed variant or not. If the cell is replaced with the mixed threshold version, static timing analysis is run again to check if timing constraints are met. Hence the resultant netlist has the same area but with a lower leakage than the original netlist.

In this work, we propose a transistor level dual- V_{th} design flow that can be seamlessly integrated in today's industrial low power design flow. It leverages the incremental leakage power optimization in the EDA tool, but with a mixed V_{th} cell library.

Compared to the existing approaches, our method has the following advantages. First, previous methods are after technology mapping. They lose the opportunity to optimize leakage in the technology mapping phase and have severe limitations when the circuit has tight timing constraints. Our approach includes the mixed Vth cell library at the gate-level synthesis phase, which essentially creates a larger solution space for leakage optimization in the circuit. For example, we are able to have cell variants with 20% timing increase on critical arcs and leave it to the proven EDA tools to map the cells, meet the timing requirements and optimize leakage simultaneously. While in [4], cell variants can only have 1% timing penalty. Second, we propose a divide and conquer heuristic to design the mixed Vth cell library. This results in a mixed cell library of significantly smaller size as compared to previous methods. To demonstrate our approach, we use industry standard simulation tools and models to characterize the cell variant and build a mixed Vth cell library. We have successfully integrated our design methodology using mixed Vth cell library in one of the commercial EDA tools for low power design.

Chapter 4: Design Methodology of Mixed-Vth Cells

Introduction to Mixed Vth Design

A cell is the fundamental digital block for an ASIC design. Cells are the basic blocks of the gate level netlist. Each cell implements a combinational or a sequential function and is realized using NMOS and PMOS transistors. Cell libraries contain the timing and power characteristics of a given cell. Each cell implements a combinational block and is of a given drive strength. Cell libraries are used to map a high level design to a gate level netlist. A cell with high and low threshold at the transistor level is referred to as a Mixed Vth Cell. Generally, cell libraries come in two flavors. One is the HVT cell library and the other is the LVT library. An LVT library contains cells where all the transistors are built with low Vth; a HVT library contains all the cells built with high Vth. Transistor level Dual-Vth refers to building cells with multiple thresholds at the transistor level. For example, a NAND2 cell has two NMOS and two PMOS transistors. The high Vth version of the cell would have NMOS with high threshold and PMOS with high threshold. The low Vth version of the cell would similarly have low thresholds for NMOS and PMOS. We could use a mixture of high and low thresholds at the transistor level, meaning, each of the NMOS and PMOS transistors could potentially have a high or a low threshold. Cells with multiple thresholds are hence known as Mixed-Vth Cell and a library characterized with multiple cells with dual thresholds are known as Mixed-Vth Library or MVT library. MVT libraries are designed to have characteristics that fall in between the HVT and LVT libraries. The HVT cells are slower, but less leaky, whereas LVT cells are faster but have high leakage. MVT cells delay characteristics are better than that of HVT cells and leakage characteristics are better than that of a LVT cell.

Low-Leakage Cell Variant Design

In a cell library, the delays of a specific cell are related to timing arcs of the cell. A timing arc defines the timing relationship between one input and one output pin. The propagation delay is defined as the maximum pin-to-pin delay among all the timing arcs in the gate. For example, in a two-input NAND gate shown in Figure 4.1, there are four timing arcs from its input pins $a1$ and $a2$ to its output pin Zn . These timing arcs have different rise and fall delays. The delay on each timing arc is shown in Table 4.1.

Under the same input slew rate and output load capacitance, the delay in the timing arcs can differ by up to 52% (see the first and last rows in Table 4.1). As the propagation delay in the gate is determined by the critical timing arc that has the largest delay, we can leverage this timing asymmetry to reduce leakage by assigning high V_{th} values to transistors that are on non-critical timing arcs.

Consider example of a NAND2 cell. The Low V_{th} version is shown in Figure 4.1 where both NMOS and PMOS transistors are in low thresholds. The timing arcs of this cell are given in Table 4.1. When we assign NMOS transistor $T1$ to high V_{th} , the maximum propagation delay of the gate remains the same, but the leakage reduces from 1.60 nA to 0.86 nA.

Table 4.2 shows the timing arc distribution in case of a Mixed Variant of a NAND2 cell. The timing arc numbers of NAND2 cell for the MVT cell are very intuitive. The $0 \rightarrow 1$ transition takes longer time than $1 \rightarrow 0$ transition for the given Length to Width ratios of NMOS and PMOS.

Thus, output discharging falls in the non-critical path for the given cell and takes the least amount of time. Hence if the transistors on the non-critical path are flipped to high- V_{th} , it might not impact the delay of the overall cell by a huge amount.

Although it is trivial in the case of the NAND2 cell, it becomes hard to apply this on complex cells that have many transistors. A brute-force approach would have to enumerate all the 2^n different V_{th} assignments in a cell with n transistors to find the most leakage effective variant. This is not practical from both technical and economic points of view.

}

Table 4.1: Timing arcs in a standard cell nd02d1 with 1x drive strength from a 130nm cell library.

a1	a2	Zn	Delay (ps)
0→1	1	1→0	78.415
1	0→1	1→0	98.14
1→0	1	0→1	108.95
1	1→0	0→1	119.44

Figure 4.1: Schematic view of an NAND2 LVT

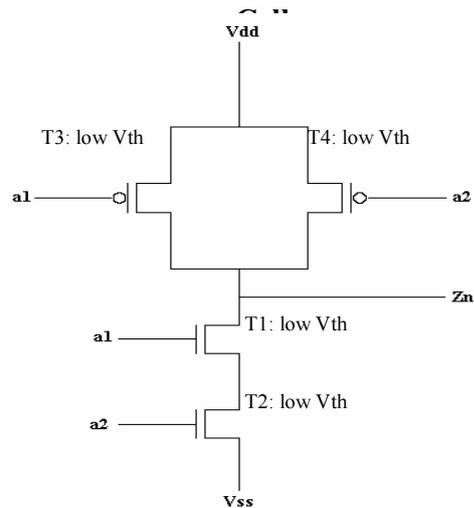
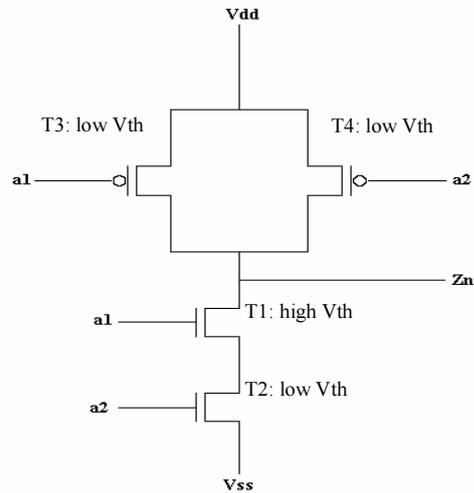


Table 4.2: Timing arcs in a standard cell, with MVT variant nd02d1 with 1x drive strength from a 130nm cell library.

a1	a2	Zn	Delay (ps)
0→1	1	1→0	94.4
1	0→1	1→0	112.18
1→0	1	0→1	109.4
1	1→0	0→1	119.4

Figure 4.2: Schematic view of an NAND2 MVT



We use a divide-and-conquer heuristic to generate low leakage cell variant for complex gates. First we partition a complex cell into multiple simple cells. For each simple cell, we pick the most leakage effective variant and generate a new cell for the original complex cell. Then we measure the propagation delay in this new complex cell. If the delay can meet our timing constraint, we save the new complex cell as a variant of the original complex cell; otherwise, we replace one of the cell variant of the simple cell by a nominal cell and measure the delay again for the complex cell. This process repeats until no new complex cell can be generated.

Consider a two-input AND2 cell whose schematic consists of an NAND2 cell followed by an inverter cell INV, a total of 6 transistors. Instead of trying $2^6=64$ different V_{th} assignments in this cell, we apply the low-leakage variant of NAND2 and low-leakage variant of INV together. It produces a low-leakage cell variant for the AND2 cell with a 13% timing penalty and a 50% leakage reduction. By this method, we can greatly reduce the

complexity of creating cell variants for complex cells. This method could essentially be extended for multi stage cells and sequential cells.

Design Methodology Using Mixed Vth Cells

In this section, we describe our methodology of designing low leakage power digital circuits using mixed Vth cell library in the commercial EDA tools. We first describe a method to design leakage-optimized cell variant for a standard cell by assigning high Vth to some of its transistors. Second, we explain how to build a mixed Vth cell library including the new low-leakage cell variants. Finally, we propose a method to integrate this design methodology in the existing EDA tool.

A. Low-Leakage MVT Cell Variant Creation

The asymmetry in the timing arcs of a given cell is used in designing the MVT variant of the cell. As explained in the previous section, critical timing arcs of every cell are determined.

1. Determine for each cell under consideration, a set of critical arcs. Critical arcs are the timing arcs that take maximum time for a given input transition that results in an output transition.
2. Create variant netlist for each type of gate by systematically changing thresholds of various transistors.
3. Calculate the timing arcs of the variant cell for a given slew rate and load capacitance based on the drive strength of the cell.
4. From the timing arc numbers of variant cell, determine the delay penalties on critical arcs and calculate the leakage.
5. If the delay penalties on critical arcs are lower than 20% then save the configuration of the particular mixed threshold variant for full library characterization.
6. Choose 1 to 3 best possible implementations, out of all the exhaustive possible mixed threshold implementations of a given cell, for complete library characterization

7. Repeat Steps 1-5 for all the most frequently used cells by simulating using HSPICE with netlist changed accordingly

B. Mixed Vth Cell Library Creation

In an industry standard cell library currently used by the low power design EDA tool today, there are often two libraries: a LVT library that contains cells where all the transistors are built with low Vth; an HVT library that contains all the cells built with high Vth. The dual-Vth assignment algorithm in the logic synthesis of the EDA tool will map gates to cells in the LVT and HVT libraries under timing constraint. HVT cells are preferred as they have low leakage.

We create a new library MVT that contains cells which have mixed low Vth and high Vth in different transistors of the same cell. We characterize the delay and leakage of cells in the LVT and HVT. We pick about the top 3% of mostly frequent used cells. More cells can be selected for aggressive low leakage design. For each cell, we create 1 to 3 variants, as described in the previous subsection. The maximum delay penalty in the cell variant is set to be 20%. We add the cell variant to the MVT library and finally, we characterize the delay and leakage of each cell in MVT.

C. Dual-Vth Design Flow Using Mixed Vth Library

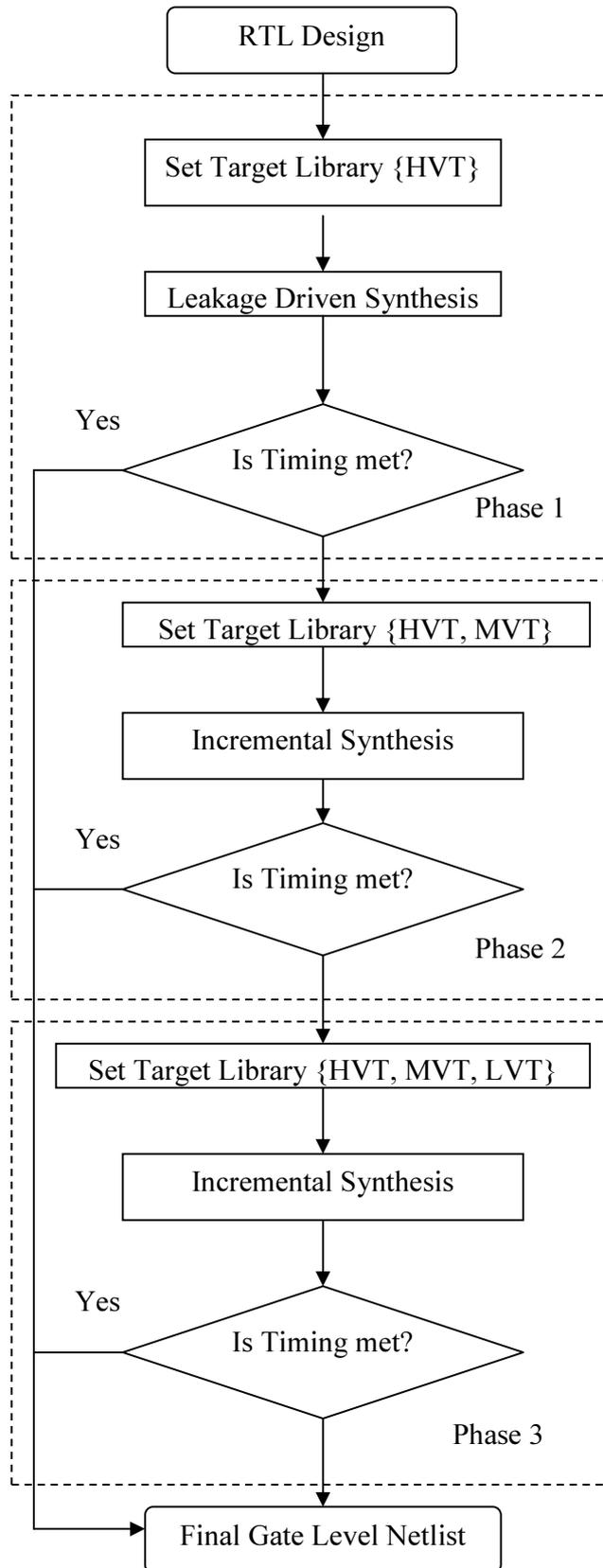
We introduce the MVT library in low leakage dual-Vth design flow. To leverage the efficiency and effectiveness of the dual-Vth assignment algorithm in the existing EDA tool, we integrate the mixed Vth assignment process in the logic synthesis stage. Our new design flow is shown in Figure 4.3. The flow consists of three phases. In Phase 1, the RTL description of a design is read in and the library is set to HVT in which each cell generates small leakage, but has large delays. The consequent leakage-driven logic synthesis procedure will map the design. If timing is satisfactory, we exit the flow with a leakage-optimized

design. Otherwise, in Phase 2, we set the target library to be the combination of HVT and MVT. An incremental synthesis is performed which only re-map the gates that have negative timing slack to a library cell with smaller delay. Since cells in MVT have smaller delays than their correspondents in HVT, the delay of the circuit will potentially be reduced. If the timing still does not meet the requirement, we enter Phase 3, where the target library is set to the combination of all HVT, MVT, and LVT libraries. Since the LVT contains all the fastest cells, the results after incremental synthesis should guarantee that the timing of the design be met.

We note that it is crucial to start the target library with the HVT library instead of the combination of HVT, MVT and LVT. Since the synthesis algorithm makes certain heuristic decisions, it does not always pick the most leakage-optimized solution. By running synthesis in three phases, we guide the synthesis tool to use the leakage optimized cells to the maximum extent. Besides, the incremental synthesis procedure is fast, which does not increase much of design-time overhead.

Finally, we mention that there are two major advantages in our method as compared to that in [4]: First, [4] applies mixed Vth cells only after the logic synthesis is complete. The design space of optimizing leakage using mixed Vth cells is restricted, because all the gates in the design are mapped and their approach can only replace the cell with the same type of mixed Vth cell. For this reason, in their mixed Vth cell creation, they can only allow 1% delay penalty, while in our approach we can allow up to 20%. The incremental synthesis procedure will pick the right cells that meet timing while optimizing leakage. Second, our method can be seamlessly integrated in the existing EDA low power flow. As a matter of fact, when the circuit has very tight timing constraints, we observe that [4] fails to replace a single mixed Vth cell variant.

Figure 4.3: Dual Vth Design Flow using Mixed Vth Cells



Chapter 5: Implementation

Mixed Vth Library Characterization

The original LVT and HVT libraries contain 590 cells respectively that are implemented using 130nm technology. We first identify 15 most frequently used cells from the LVT library and create 1 to 3 variants for each of them.

The cells that were considered for the variants based on synthesis reports of large set of real designs and ISCAS benchmarks - inv0d1,inv0d2 , inv0d4 , nd02d1 , nd02d2 , nd02d4 ,buffd1, buffd2, an02d1, an02d2 , an02d4 , or02d1 , or02d2 , or02d4 , nr0d1

We use Synopsys HSPICE and Atmel tool to characterize the leakage current and delay in the cell variants and add them to the MVT library as described in Section II. The final number of cells in MVT is 40. The following are the implementation details for a given cell.

HSPICE Simulation Setup

HSPICE was used for the timing characterization of the cell variants. Variants are cells that have mixed threshold for transistors (both high and low) in their cell topology. The first step in the whole process of Mixed Threshold CMOS design is to take RC extracted netlist of HVT and LVT transistor netlists and compare them. When the extracted netlist of nd02d1 and nd02d1_hvt cells where compared, it was found that the only difference was the transistor models.

HSPICE Timing Characterization

Based on the drive strength of the cell being characterized, a set of load and slew ranges are determined. HSPICE simulation is run to characterize the timing on each pin for each possible transition.

Leakage Characterization

Leakage of every variant is calculated as the average leakage power for different inputs to the gate. All possible inputs are applied to a given cell and the sub threshold current is measured at steady state for every combination. The leakage power is given by the average of all possible states.

Library characterization is run on all the 40 MVT cells with 130nm model files. The final library is obtained in either .db or .lib format which can later be directly used by Synopsys Design Compiler.

Synthesis Flow

Figure 5.1 describes the generic synthesis flow for the designs using Synopsys Design Compiler. For every design/benchmark the following synthesis flow was used with different design specific constraints.

Develop HDL Files: In our case, the HDL files are given either in Verilog/VHDL format for synthesis.

Specify Libraries: HVT, MVT, LVT libraries are the target libraries for various benchmarks.

Read Design: This step refers to reading the design and checking for syntax and other linking errors and is done using the command `read_verilog`

Define Design Environment: External and internal operation conditions to model the environment are specified in this step. For our synthesis, we use the worst case corner files

for defining the design environment. This includes worst case temperature, process and voltage for 130nm node. Auto wire load models are used for estimating the wire load and inverters with 1X drive strength are used as drive cells

Set Design Constraints: This step refers to setting constraints that are design specific. Timing, area and power constraints are specified in this step. For sequential benchmarks, clock frequency is also specified with other clock related parameters in this step. The design constraints are set such that the design meets timing when synthesized with target library as LVT.

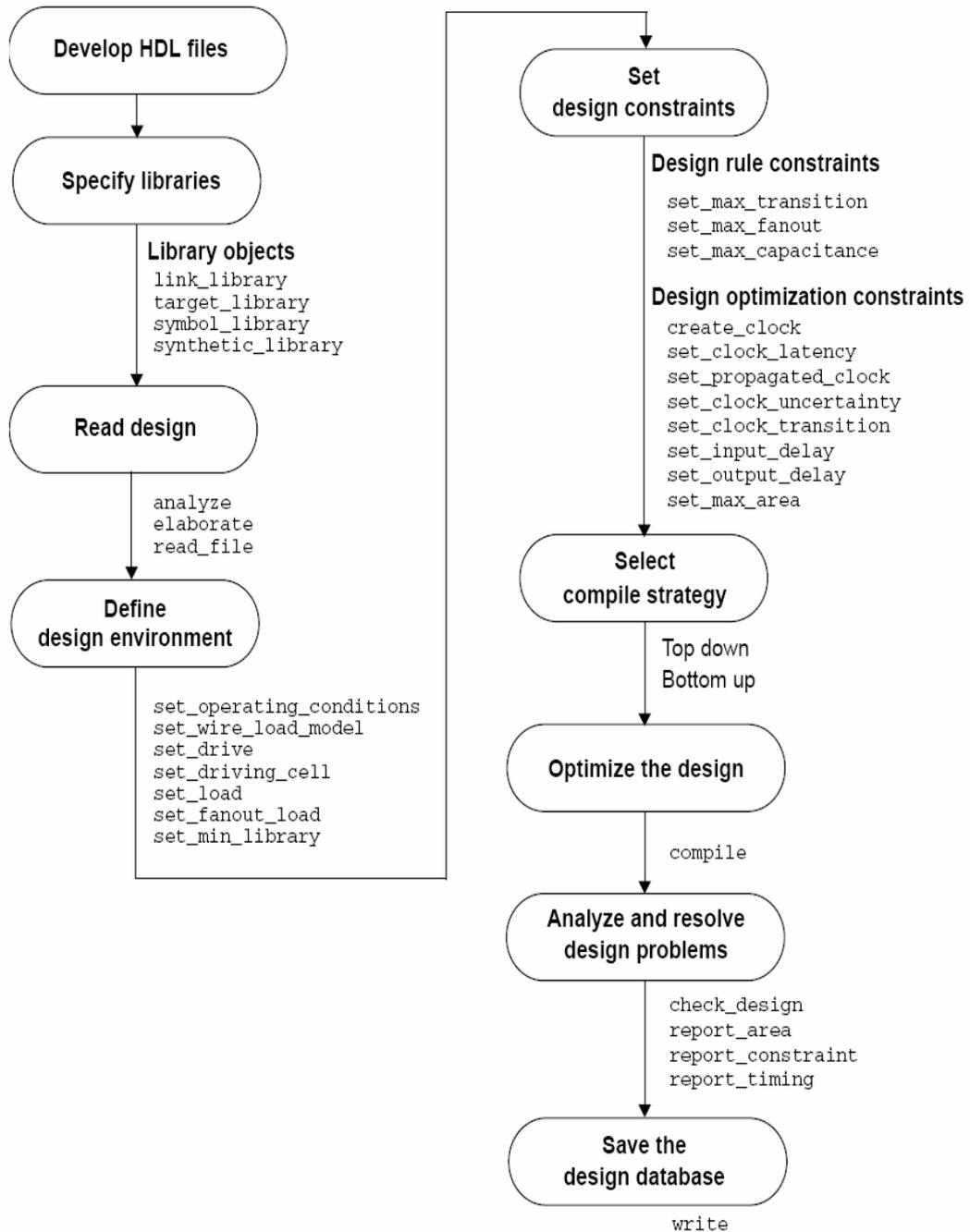
Compile Strategy: This step refers to the compile mode used for synthesizing the design. Design compiler supports many modes and a combination of the modes could be used to synthesize the design. Compile `-inc` is often used in our case to perform incremental synthesis to improve the quality of results and meet the design constraints.

Optimize Design: This step refers to setting appropriate optimization flag to guide the design compiler to perform specific optimization. `Set_max_leakage_power` flag is used in our case to optimize designs for leakage power.

Analyze and Resolve Design Problems: This step refers to checking the synthesis reports for errors and changing the flow accordingly to meet the goals. Often synthesis is done multiple times so as to meet the design constraints and is usually iterative.

Save the design database: This is the last step in synthesis and refers to saving the current synthesized design netlist.

Figure 5.1: Generic Synthesis flow using Design Compiler [9]



Chapter 6: Results

Leakage Reduction

After synthesis, the Power Compiler reports the total leakage power in each design. The results are shown in Table 6.1. The first eleven designs are from ISCAS benchmarks and the last three are real designs from ATMEL Corp. [29]. The number of cells in the designs ranges from 1269 to 70966. The circuits are first mapped to LVT library and the delays are shown in the third column. We use this delay as the timing constraint in the dual-Vth assignment. The synthesis design flow explained before is used to target a design to a combination of different libraries.

The fourth column shows the total leakage power in each circuit. The fifth column shows the leakage reduction when we apply the conventional dual-Vth assignment using HVT and LVT libraries. 34% leakage power reduction is achieved on average. The sixth column shows the leakage power reduction when our mixed Vth cell design methodology is applied. We can observe an average of 40% leakage reduction, which is 9% improvement over the dual-Vth assignment with only HVT and LVT libraries.

Considering ISCAS benchmarks alone, the average leakage reduction and improvement are 42% and 10% respectively. It should be noted that design3 has small improvement (2%) over conventional dual-Vth flow. This is because the design3 has many cells with positive timing slack, which makes it possible for Design Compiler to use many cells from the HVT library, leaving little room for MVT cells.

Table 6.1: Leakage reduction.

Design	# Cell	Delay (ns)	Leakage (uW) LVT	Leakage Red. (%) HVT+LVT	Leakage Red. (%) HVT+LVT +MVT	improv. %
C2670	1269	1.6	1.03	16.67	33.28	19.94
C3540	1669	3.2	1.42	22.84	26.06	4.18
C5315	2307	2.5	1.61	26.25	39.85	18.44
C6288	2416	8.5	2.57	13.88	16.61	3.18
C7552	3513	2.25	2.08	26.45	33.70	9.86
s9234	5808	2.3	0.98	18.32	22.93	5.64
s35932	12204	2	7.12	41.89	44.23	4.02
s15850	10306	3.2	3.26	46.98	60.68	25.84
s38417	23815	3.2	10.54	45.82	49.56	6.89
s13207	8589	2.75	2.84	64.45	65.38	2.61
s38584	20679	3.2	8.12	61.02	64.37	8.58
Design1	16838	4	30.2	19.45	24.3	6.02
Design1	30644	4.2	43.08	14.62	21.01	7.48
Design3	70966	13	85.57	63.58	64.2	1.94
Average				34.4	40.4	8.9

MVT Cell Usage Statistics

The statistics of LVT, HVT and MVT cell usage in the synthesized Netlist are reported in Table 6.2. The second column shows the total count of MVT cells. The third and fourth columns show only the count of the 15 most frequently used type of LVT and HVT cells; only these cells may have variants in MVT. We observe that on average 16% of cells in the design belong to the 15 most frequently used cells, as shown in the fifth column. It indicates the percentage of the design space that our approach is optimizing. The last column shows the

percentage of cells that have been mapped to MVT library among the frequently used cells.

This represents the percentage of cell from which leakage has been optimized using MVT.

Table 6.2: MVT cell usage.

Design	#MVT Cells	#HVT Cells*	# LVT Cells*	% Frequently Used Cells	%MVT Used
C2670	50	270	15	26	15
C3540	54	133	27	13	25
C5315	84	231	30	15	24
C6288h	265	388	62	30	37
C7552	80	304	46	12	19
s9234	58	137	23	4	27
s35932	119	1027	32	10	10
s15850	102	656	36	8	13
s38417	355	1246	142	7	20
s13207	61	521	21	7	10
s38584	229	2173	55	12	9
Design1	587	3128	322	24	15
Design2	2748	5265	957	29	31
Design3	937	19708	371	30	4
Average				16	19

Total Negative Slack Statistics

Total negative slack (TNS) refers to the sum of the negative slack present in the design when the design does not meet timing. To observe the TNS statistics, designs are over constrained.

The Table 6.3 shows the TNS for the design C7552 when the timing constraint is 1 ns. It is

observed that when the design is over constrained, the total negative slack distribution in case the LVT+MVT+HVT is the lowest and is 14% lower than the LVT case. Also observe that the slack in the critical path is lowest for the LVT+HVT+MVT. This result indicates that having MVT library also gives another advantage of improving the timing results. This result is intuitive in the sense that a better slack utilization is observed in case of HVT+LVT+MVT case.

Table 6.3: Total Negative Slack Statistics

Design Name : C7552 , Timing Constraint = 1 ns		
Libraries	Worst Slack	TNS
LVT	-1.16	-48.98
LVT+HVT	-1.07	-46.2
LVT+MVT+HVT	-0.95	-42.37
HVT	-1.89	-85.34

Hence, having MVT gives us advantage in the sense that it could at times improve timing and would help in better slack utilization.

Chapter 7: Conclusion and Future Work

In this thesis we present a methodology to use mixed Vth cells in the dual-Vth design flow and integrate it in the existing EDA tool. Our methodology leverages the synthesis procedure in the EDA tool and creates more chances for the EDA tool to optimize leakage with mixed Vth cells. We applied this methodology on both ISCAS benchmarks and real-life designs using a state-of-the-art industry EDA tool. We achieve on average 40% leakage power reduction which is 9% improvement over the conventional dual-Vth design flow. Furthermore, the potential to reduce/recover leakage also appears to be huge. On average only 16% of the design space has been selected for leakage optimization and 19% of the MVT cells are used.

As an extension to this work, similar design flow approach could be applied to designing standard cells with gate length biasing as an alternative way to improve leakage. The second alternative to designing standard cells is to efficiently use sizing on HVT cells so as to improve their timing without increasing the area with minimal increase in leakage. This method is explained with an example of a NAND2 cell with 1X drive strength. When standard cells are designed for a given library, often, the height of the standard cell is fixed also know as the *pitch*. The length of every transistor was also fixed in Atmel's Library and is equal to 0.13 microns. The height of a standard cell is fixed such that a 2X drive strength of NAND2 cell can be fitted in. This is purely a design metric used for simplicity for designing cells.

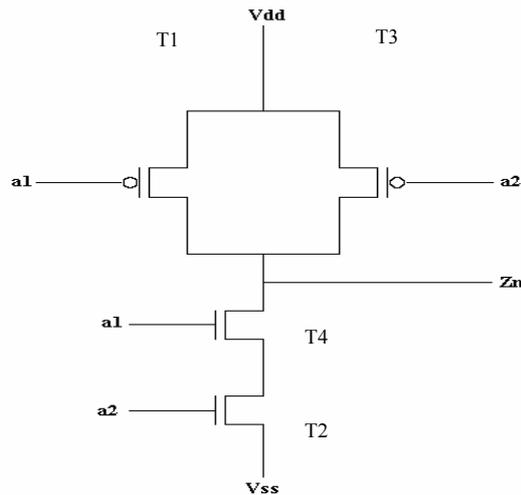
Table 7.1: Delay values for a 1X NAND2 cell with transistor sizing

a1	a2	Zn	Delay (ps) HVT cell	Delay (ps) Sizing_T1	Delay (ps) Sizing_T1_T3	Delay (ps) LVT Cell
0→1	1	1→0	103	104	104	78
1	0→1	1→0	131	132	132	98
1→0	1	0→1	148	148	129	109
1	1→0	0→1	161	139	140	119

Table 7.2: Width Sizing and Leakage for various transistor configuration

Sizing	PMOS_T1 (microns)	PMOS_T3 (microns)	NMOS_T2 (microns)	NMOS_T4 (microns)	Leakage (nW)
NAND2_LVT	0.65	0.65	0.58	0.58	1.73
NAND2_HVT	0.65	0.65	0.58	0.58	0.20
NAND2_HVT_T1	0.72	0.65	0.58	0.58	0.21
NAND2_HVT_T1_T3	0.72	0.72	0.58	0.58	0.21

Figure 7.1: NAND2 Cell Schematic



This essentially means that all the cells with 1X drive strength would have an unused area left. This could potentially be exploited by sizing the transistors on critical path of the cell to gain performance while keeping the leakage low.

For example, in case on a NAND2 1X cell, the timing arc, leakage and sizing details are given in Table 7.1 and Table 7.2 . We note that from timing arcs, PMOS transistor fall in the critical path for the given sizing ratios. Hence when the PMOS transistors of the HVT cells are sized, we get appreciable delay improvements. In the NAND2_HVT_T1_T3 case, the delay on the critical arcs get improved by 12% with little increase in leakage when compared to a HVT cell and is around 12% slower than the LVT cell. This would mean that the cell would have characteristics that are exactly between the HVT and the LVT cell, while the leakage is reduced by 8X. The penalty paid when sizing is done is to do with the increase in the input capacitance and increases linearly with sizing.

Hence, transistor sizing could be combined with designing Mixed Vth Cells to reduce leakage without too much penalty. This method presents a huge potential in terms of gaining on leakage and not compromising on speed without increase in area of the cell.

Appendices

Nomenclature:

The number after the alphabet “d” in any cell name indicates the drive strength of the cell. For example, inv0d1, would mean it’s an inverter with 1 input, 1 output with 1X drive strength. “Inv” indicates its inverter. “nd” stands for NAND, “nr” stands for NOR , “an” stands for AND gate , “or” stands for an OR gate ,“df” would stand for a D- flip flop and “buff” stands for a BUFFER. The digit following the type of gate usually gives information about the number of inputs to the gate. For example, an02d1 would mean its two inputs and gate, with 1 X drive strength.

List of top 15 cells are given by

Inv0d1, Inv0d2 , inv0d4 , nd02d1 , nd02d2 , nd02d4 ,buffd1, buffd2, an02d1, an02d2 , an02d4 , or02d1 , or02d2 , or02d4 , nr0d1 , dfcrq1

Timing, leakage values of each of the gates haven't been disclosed in the thesis for the reasons of confidentiality. For more details readers could contact

Barbara G Stamps

Design Manager, Atmel Corporation

Chesapeake Design Centre

Email: bstamps@atmel.com

Bibliography

- [1] A. Abdollahi, F. Fallah, and M. Pedram, "Leakage Current Reduction in CMOS VLSI Circuits by Input Vector Control", *IEEE Trans. VLSI*, vol. 12, pp. 140-154, Feb. 2004.
- [2] A. Agarwal, C. H. Kim, S. Mukhopadhyay and K. Roy, "Leakage in Nano-Scale Technologies: Mechanisms, Impact and Design Considerations," in Proc. ACM/IEEE Design Automation Conference, 2004, pp. 6–11.
- [3] F. Beftink, P. Kudva, D. Kung and L. Stok, "Combinatorial Cell Design for CMOS Libraries," *Integration, the VLSI Journal*, vol. 29, no. 4, pp. 67–93, 2000.
- [4] P. Gupta, A. B. Kahng, P. Sharma, "A Practical Transistor-Level Dual Threshold Voltage Assignment Methodology", in *Proc. IEEE Intl. Symp. on Quality Electronic Design*, pp. 421-426, 2005.
- [5] P. Gupta, A. B. Kahng, P. Sharma and D. Sylvester, "Selective Gate-Length Biasing for Cost-Effective Runtime Leakage Control," in Proc. ACM/IEEE Design Automation Conference, 2004, pp. 327–330.
- [6] J. Halter and F. Najm, "A Gate-level Leakage Power Reduction Method for Ultra Low Power CMOS Circuits," in *IEEE Custom Integrated Circuits Conference*, 1997, pp. 475–478.
- [7] M. Horiguchi, T. Sakata and K. Itoh, "Switched-Source-Impedance CMOS Circuit for Low Standby Sub-Threshold Current Giga-Scale LSI's," *IEEE Journal of Solid-State Circuits*, vol. 28, no. 11, pp. 1131–1135, 1993.
- [8] I. Hyunsik, T. Inukai, H. Gomyo, T. Hiramoto and T. Sakurai, "VTCMOS Characteristics and its Optimum Conditions Predicted by a Compact Analytical Model," in *International Symposium on Low Power Electronics and Design*, 2001, pp 123–128.
- [9] J. Kao, S. Narendra and A. Chandrakasan, "MTCMOS Hierarchical Sizing Based on Mutual Exclusive Discharge Patterns," in Proc. ACM/IEEE Design Automation Conference, 1998, pp. 495–500.
- [10] M. Ketkar and S. Saptnekar, "Standby Power Optimization via Transistor Sizing and Dual Threshold Voltage Assignment," in Proc. IEEE International Conference on Computer Aided Design, 2002, pp. 375–378.
- [11] D. Lee and D. Blaauw, "Static Leakage Reduction Through Simultaneous Threshold Voltage and State Assignment," in Proc. ACM/IEEE Design Automation Conference, 2003, pp. 192–194.

- [12] S. Mutoh, T. Douseki, Y. Matsuya, T. Aoki, S. Shigematsu and J. Yamada, "1-V Power Supply High-Speed Digital Circuit Technology with Multithreshold-Voltage CMOS," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 8, pp. 847–854, 1995.
- [13] S. Mutoh, S. Shigematsu, Y. Matsuya, H. Fukada, T. Kaneko and J. Yamada, "1V Multithreshold-Voltage CMOS Digital Signal Processor for Mobile Phone Application," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 11, pp. 1795–1802, 1996.
- [14] S. Narendra, D. Blaauw, A. Devgan and F. Najm, "Leakage Issues in IC Design: Trends, Estimation and Avoidance," in *Proc. IEEE International Conference on Computer Aided Design*, 2003, tutorial.
- [15] D. Nguyen, A. Davare, M. Orshansky, D. Chinnery, B. Thompson and K. Keutzer, "Minimization of Dynamic and Static Power Through Joint Assignment of Threshold Voltages and Sizing Optimization," in *International Symposium on Low Power Electronics and Design*, 2003, pp. 158–163.
- [16] K. Nose, M. Hirabayashi, H. Kawaguchi, S. Lee and T. Sakurai, "Vth Hopping Scheme to Reduce Sub threshold Leakage for Low-Power Processors," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 3, pp. 413–419, 2002.
- [17] R. Puri et al, "Keeping hot chips cool", In *Proc. ACM/IEEE Design Automation Conference*, 2005, pp. 285-288.
- [18] S. Shigematsu, S. Mutoh, Y. Matsuya, Y. Tabae and J. Yamada, "A 1-V High-Speed MTCMOS Circuit Scheme for Power-Down Application Circuits," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 6, pp. 861–869, 1997.
- [19] S. Sirichotiyakul, T. Edwards, C. Oh, R. Panda and D. Blaauw, "Duet: An Accurate Leakage Estimation and Optimization Tool for Dual-Vth Circuits," *IEEE Transactions on Very Large Scale Integrated Systems*, vol. 10, no. 2, pp. 79–90, 2002.
- [20] S. Sirichotiyakul, T. Edwards, C. Oh, J. Zuo, A. Dharchoudhury, R. Panda and D. Blaauw, "Stand-by Power Minimization through Simultaneous Threshold Voltage Selection and Circuit Sizing," in *Proc. ACM/IEEE Design Automation Conference*, 1999, pp. 436–441.
- [21] A. Srivastava, D. Sylvester and D. Blaauw, "Power Minimization using Simultaneous Gate Sizing, Dual-Vdd and Dual-Vth Assignment," in *Proc. ACM/IEEE Design Automation Conference*, 2004, pp. 783–787.

- [22] Q. Wang and S. B. K. Vradhula, "Static Power Optimization of Deep Submicron CMOS Circuits for Dual VT Technology," in Proc. IEEE International Conference on Computer Aided Design, 1998, pp. 490–495.
- [23] L. Wei, Z. Chen, M. Johnson, K. Roy and V. De, "Design and Optimization of Low Voltage High Performance Dual Threshold CMOS Circuits," in Proc. ACM/IEEE Design Automation Conference, 1998, pp. 489–494.
- [24] L. Wei, Z. Chen, K. Roy, Y. Ye and V. De, "Mixed-Vth CMOS Circuit Design Methodology for Low Power Applications," in Proc. ACM/IEEE Design Automation Conference, 1999, pp. 430–435.
- [25] L. Wei, K. Roy and C. K. Koh, "Power Minimization by Simultaneous Dual-Vth Assignment and Gate-Sizing," in Proc. ACM/IEEE Design Automation Conference, 2000, pp. 413–416.
- [26] Y. Ye, S. Borkar and V. De, "A New Technique for Standby Leakage Reduction in High-Performance Circuits," in Proc. Symposium on VLSI Circuits, 1998, pp. 40–41.
- [27] Synopsys Design/Power Compiler User Guide, Version Z-2006.06, June 2006
- [28] Application Specific Integrated Circuits, Michael John Sebastian Smith, Addison-Wesley
- [29] Atmel Corporation, Chesapeake Design Centre, Columbia.
- [30] F. Fallah, M. Pedram, "Standby and Active Leakage Current Control and Minimization in CMOS VLSI Circuits," Special Low-Power LSI Issue of IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences, 2005
- [31] Wei et.al 'Low voltage low power CMOS design techniques for deep submicron ICs', Proceedings of the thirteenth international conference on *VLSI design*, 2000, pp. 24–29
- [32] L. Yuan and G. Qu. "A Combined Gate Replacement and Input Vector Control Approach for Leakage Current Reduction," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 14, No. 2, pp. 173-182, February 2006.