LUND UNIVERSITY

DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY

MASTER OF SCIENCE THESIS

Clock tree design in sub-Vt circuits - Analysis on standard- and full-custom gates

Author:
Yuqi Liu

Supervisor:
Oskar Andersson
Babak Mohammadi
Joachim Rodrigues

Lund June 19, 2013

The Department of Electrical and Information Technology
Lund University
Box 118, S-221 00 LUND
SWEDEN

This thesis is set in Computer Modern 10pt,
with the LaTeX Documentation System

Abstract

Process variation at the subthreshold (sub-V_{T}) region is a key factor that affects the functionality of a clock tree. Standard-cell clock buffers experience more performance variation among different design corners than that in the nominal supply voltage (V_{DD}). This variation effect should thus be reduced. In this report, some aspects of constructing clock tree for sub-V_{T} operation has been studied and compared. It has been confirmed that performing clock tree synthesis (CTS) at nominal V_{DD} and loading lower V_{DD} libraries is more applicable than applying CTS directly in the sub-V_{T} region. Mixing different V_{T} transistors, also known as dual-V_{T} method, has been employed customizing standard-cell buffer for constructing robust clock networks. It has been approved in the simulation that this customized buffer could increase the performance of delay, skew and slew by factor of 9.8×, 8.9× and 9.8× respectively at 400mV.

Acknowledgments

I would like to thank my supervisor, Oskar Andersson who has been an excellent teacher and always very patient helping me throughout the entire project. I would like to thank my supervisor Babak Mohammadi who has been given great help in analog part of this project. Special thanks to Dr. Joachim Neves Rodrigues for providing valuable advices and guidance.

Yuqi Liu
Lund, June 2013

Contents

List of Tables

List of Figures

List of Acronyms

ULV Ultra-Low Voltage

VT Threshold Voltage

ASIC Application Specific Integrated Circuit

CTS Clock Tree Synthesis

SPICE Simulation Program with Integrated Circuit Emphasis

MOS Metal-Oxide-Semiconductor

DIBL Drain Induced Barrier Lowering

HVT High Threshold Voltage

SVT Standard Threshold Voltage

LVT Low Threshold Voltage

CMOS Complementary Metal-Oxide-Semiconductor

PMOS P-type Metal-Oxide-Semiconductor Field-Effect Transistor

NMOS N-type Metal-Oxide-Semiconductor Field-Effect Transistor

PUN Pull-up Network

PDN Pull-down Network

FIR Finite Impulse Response

P&R place-and-route

LUT Look Up Table

FO4 Fan-out Four delay

Chapter 1

Introduction

Ultra-low power (ULP) circuits are gaining more and more concern in not only research but also commercial field due to the power efficient characteristic [1]. Nowadays, ultra-low voltage (ULV) circuits are often used to achieve ULP. For applications which are concerning ULP, e.g.) biomedical devices, environmental monitoring devices, battery lifetime is usually required at the range of one or few years. Thanks to the relation between supply voltage and power consumption [1], ULV circuits, which are circuits operate at supply voltage near or even lower than threshold voltage, are possible to achieve 10-20 times or more energy savings [2]. Thus, ULV circuit becomes a promising way of fulfilling the requirement of those applications mentioned above.

Due to the fact that transistor on-current has an exponential relation with respect to supply voltage, the circuit speed would thus be lowered compare to nominal voltage [1]. Although, speed is not the key factor concerned in ULV, it is still desired to accomplish higher speed and better performance while maintaining almost the same amount of energy consumption. Therefore, clock network should be treated carefully since it is a critical part of synchronous design which limits the speed of the entire circuit. Though gate quality and logic style could affect the critical path, performance of the clock network is essentially an important part which should be concerned.

Lowering the supply voltage does not only decrease total energy consumption, but also the transistor on- and off-current ratio which further weaken the robustness of the design [2]. Also, ULV circuits may experience more process variation than that in the nominal V_{DD} which again increase the difficulty in constructing robust clock network for ULV system. Furthermore, it has been proven that clock slew has much effect on clock robustness in ULV region because buffer delay, flip-flop timings are functions which are strongly related to the slew [3]. Increasing clock slew would result in increasing in those timings and therefore reduce the clock speed

of the design. In short, designing clock network for ULV circuits is much different than that in super-V_T, and it requires awareness of different parameters. So the goal of this project is to improve the clock tree performance for sub-V_T region.

In this report, several aspects of constructing the clock tree has been studied and the results have been compared as well.

The rest of this report is organized as following. Theoretical background such as transistor operation character at sub-V_T, critical clock tree parameter and dual-V_T concept are shown in section 2. Reference design, digital ASIC clock tree synthesis (CTS) flow, simulation procedure and library characterization are discussed in section 3. Clock tree constructing method comparison, corner simulation, SPICE simulation and customized buffer performance are analyzed in section 4. Conclusions and summary are addressed at the end.

Chapter 2

Theoretical Background

In order to perceive the behavior of transistors at subthreshold (sub-V_{T}) region, a transistor model has been presented. The threshold voltage adjustment technique has also been discussed so as to understand the dual-V_{T} method more clearly. Clock tree structure and the main concerning parameters are reviewed for clarity.

2.1 Sub-V_{T} Transistor Model

Sub-V_{T} usually define as the region where $V_{GS} < V_{\mathrm{T}}$, here V_{T} is the threshold voltage of the transistor. The main source of the drain current in this region is the subthreshold current originates from the carrier diffusion between drain and source terminal [1]. Although other components could be included into the model such as the gate current I_G which is the electron tunneling current through the insulator layer under the gate [1], and junction current I_J which generated from tunneling across the depletion region around the drain and source diffusion area [4], they are neglectable compare to the subthreshold current [1]. Thus, only the subthreshold current will be presented in the model. The sub-V_{T} region also known as the weak inversion region. For a four terminal MOS transistor, the subthreshold current has an expression as 2.1

$$I_{DS} = \frac{W}{L} I_M^{'} e^{(V_{GS}-V_M)/(n\phi_t)}(1 - e^{-V_{DS}/\phi_t}), \tag{2.1}$$

where $\frac{W}{L}$ is the width to length ratio, and $I_M^{'}$ is defined as:

$$I_M^{'} = \mu \frac{\sqrt{2q\epsilon_s N_A}}{2\sqrt{2\phi_F + V_{SB}^{'}}} \phi_t^2. \tag{2.2}$$

2.2 is technology dependent because μ, q, ϵ_s and ϕ_t are physical constants. Furthermore, N_A is the acceptor diffusion concentration and ϕ_F is the corresponding fermi potential related to certain diffusion concentration, both of which are technology dependent. $V_{SB}^{'}$ denotes a fixed source bulk voltage. [5] Thus, $I_M^{'}$ is purely technology dependent which can be seen as constant for certain technology [1]. V_M in 2.3 stands for the upper limit of the weak inversion region [5]

$$V_M = V_{FB} + 2\phi_F + \gamma\sqrt{2\phi_F + V_{SB}^{'}}, \tag{2.3}$$

which could also be view as the gate-source extrapolated threshold voltage described in [5]. 2.3 would be replaced by the symbol of V_{TH0} hereafter. When taking into account of drain induced barrier lowering (DIBL) and body effect, this item should be further calibrated using 2.4 [1]

$$V_{\mathrm{T}} = V_{TH0} - \lambda_{DS}V_{DS} - \lambda_{BS}V_{BS}, \tag{2.4}$$

where λ_{DS} and λ_{BS} are the corresponding coefficients for the two effects. One last thing to mention is that n is another constant called subthreshold factor [1]. It denotes the inverse of the slope of in the figure of surface potential versus gate-bulk voltage [5]. By inserting 2.4 into 2.1 and rearrange the equation, the subthreshold current is expressed as

$$I = \beta e^{V_{GS}/nv_t}[e^{\lambda_{DS}V_{DS}/nv_t}(1 - e^{-V_{DS}/v_t})], \tag{2.5}$$

where β indicates the strength of the transistor shown as [6]

$$\beta = I_M^{'}\frac{W}{L}e^{-(V_{TH0}-\lambda_{BS}V_{BS})/nv_t}. \tag{2.6}$$

It is clear that several factors could be used in tuning the transistor strength according to 2.6. For example the width to length ratio, the V_{TH0} and body biasing [1]. The width to length ratio and the V_{TH0} has been applied during cell customization in section 3.

2.2 Threshold Voltage Adjustment

Threshold voltage adjustment is a common used technique in integrated circuit manufacturing. The idea of this process step is to implant a dose of low concentration low energy beam of impurity atom into the channel region under the gate of the transistor [7]. Due to the correlation between threshold voltage and impurity doping concentration, threshold voltage can be adjusted [8].

Thanks to the process step mentioned above, three kinds of threshold voltage transistors can be constructed, high-V_{T} (HVT), standard-V_{T} (SVT) and low-V_{T} (LVT). As the name implies, HVT would have the highest threshold voltage while

LVT would be the lowest. Due to the higher threshold voltage that HVT cells have, lower leakage current is produced and further lower leakage power. LVT, on the other hand, has largest leakage power. Also, the threshold voltage is a critical factor for the transistor strength. 2.6 shows that the transistor strength has exponential relation with respect to the threshold voltage. It has been studied that 100mV threshold voltage reduction results in 18 times stronger transistor strength [1]. These characters are also the reason of doing cell customization.

2.2.1 Dual-V_{T} Method

Because of the mobility difference of electron and hole, PMOS and NMOS transistor thus have different driving strength when they have the same size. This phenomenon turns out to imbalance of rise and fall time of a cell and further reduces the noise margin. It worsens when lower the V_{DD} to sub-V_{T} levels because subthreshold current has an exponential dependency with V_{DD} as shown in 2.5. Thus, the circuit would be extremely vulnerable to process variation [9]. It is advisable to balance the strength of the pull-up and pull-down network [10]. The traditional method to balance rise and fall time in the sub-V_{T} region is to upsize the PMOS network. However, upsizing transistor at sub-V_{T} is so inefficient that even a 10× upsizing in PMOS would not achieve strength balancing [9].

As mentioned in section 2, transistors have three kinds of threhold voltage options with the same structure and the strength of the three differs widely. So, the dual-V_{T} method is to replace the weaker network's transistors by its lower-V_{T} equivalent. It has been shown in that this method could results in higher performance and reliability. [9]

2.3 Clock Tree

In a typical synchronous design, the clock network can experience the highest fanout which results in highest net capacitance. Inserting buffers is a common way of dealing with such high fanout problem since buffers has the ability to recover input signal so that the following gates would not experience much distortion [10]. Based on this fact, a common structure of the clock tree has been developed and is known as the H-tree, which is shown in Fig. 2.1.

Because of the symmetry of the structure, each of the sink would experienced the same delay introduced from the clock buffer chain. Thus, there would not be any phase difference between any of two sinks. Also, because of the insertion of clock buffers, the output capacitance of the clock net has been dispersed. For the structure without any buffers, the clock root pin would have to drive all the sinks directly which equals to $16 \cdot C_{reg}$, where C_{reg} is the capacitance of a single register. The buffered structure, on the other hand, would be completely different. The root pin only have to drive the input capacitance of the 1st level buffer, and in turn the

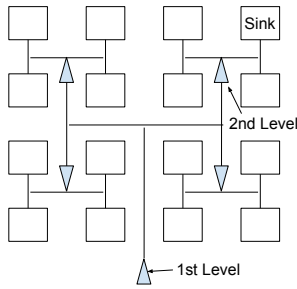


Fig. 2.1: Buffered H-tree structure

1st level buffer drives input capacitance of the 2nd level buffers, and each of the 2nd level buffers drive the sink input capacitances. Although the buffer delay has been introduced in the clock network, the buffered clock tree has a better capability in driving all sinks. Also, it has been stated that absolute delay in the clock tree does not matter as long as the difference at each end is small [10]. The H structure could be repeated so that higher level clock buffer tree could be constructed.

2.3.1 Skew

Clock Skew is commonly referred to as the maximum arrival time difference among all sinks. It is mainly caused by clock path mismatch and varied clock load [10]. Suppose that a simple synchronous circuit datapath illustrated in Fig. 2.2.

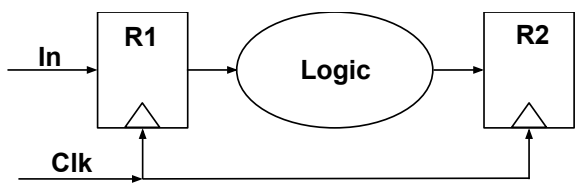


Fig. 2.2: Simple model for synchronous circuit datapath [10]

Due to the path difference between the two registers and input clock pin, the clock signal would be delayed by a small amount of time at the clock pin of R2 [10]. The clock waveform would then look like Fig. 2.3.

In order to ensure correct functionality and avoid race conditions, the following timing requirements need to be fulfilled:

$$T \geq t_{c-q} + t_{logic} + t_{su} - \text{skew}, \tag{2.7}$$

$$\text{skew} < t_{c-q} + t_{logic} - t_{hold}. \tag{2.8}$$

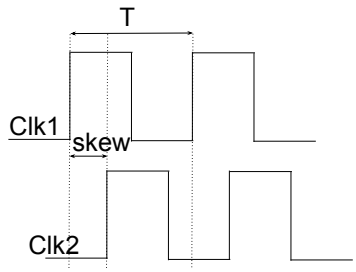


Fig. 2.3: Clock wavefront including skew [10]

2.7 shows that the clock period should be long enough so that new sampled data of R1 is able to be stable and valid a setup time before the second rising edge of Clk2. If it does not, then the circuit would have setup time violation and cause incorrect data sampling. Furthermore, 2.8 indicates that the clock-to-q delay of R1 together with the combinational logic delay should be at least longer than the skew and hold time together. Otherwise, the input of R2 would be unstable before the first rising edge of Clk2 and cause hold time violation and cause incorrect data sampling. Moreover, skew could take either positive or negative value depends on the direction of the routing and it is not possible to have only either positive or negative skew in a design. Thus, minimizing skew is in general desired [10].

2.3.2 Slew

Clock slew is the transition time that clock signal takes when changing from logic 0 to logic 1 or vice versa [10]. For an ideal clock signal, the transition of the signal happens idealy which means slew equals to zero.

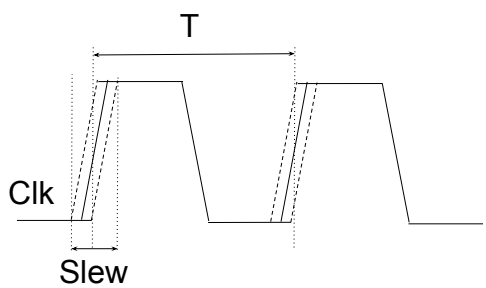


Fig. 2.4: Clock wavefront including rising edge slew [10]

In reality, clock signal needs much more time to accomplish a complete transition because of the existence of output capacitance of each clock network. Thus, the circuit should avoid any functional sampling during the clock signal transition. Fig. 2.4 shows how clock slew would look like.

Due to the clock slew, the setup time requirement of the circuit becomes [10]

$$T \geq t_{c-q} + t_{logic} + t_{su} + 2t_{slew}, \tag{2.9}$$

where the 2 before t_{slew} accounts for the slew of both the first and second rising edge. 2.9 implies that slew has worsened clock timing and should be minimized as well.

Chapter 3

Flow Description

A 16-tap transposed FIR filter is used as reference design in the project and simulations such as clock tree synthesis (CTS), SPICE simulation and library characterization have been employed as well. This section contains details on the steps applied in the simulation procedure.

3.1 Reference Design

The design that has been used throughout the project is a transposed 16-tap FIR filter with 8-bit input and 7-bit coefficient. It contains 313 bits of registers including one 8-bit input register and 19-bit output register. Due to the fact that FIR filter is a common design understood by a wide range of people and the clock tree only cares about registers in a design regardless what functionality the design might have, the FIR filter is thus chosen. For the transposed architecture, due to the fact that registers locates at the input and output branch of adders, more bits of registers are used compare to the direct form. A simple diagram of the design is shown Fig. 3.1.

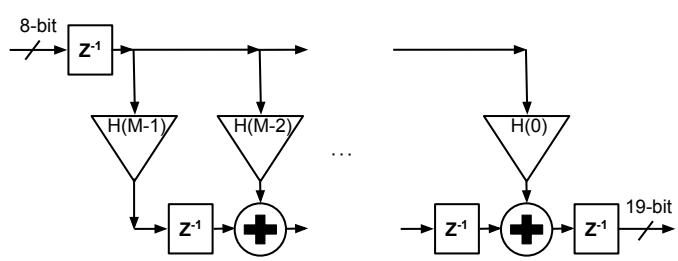


Fig. 3.1: Block diagram of the reference FIR filter

The reference design was synthesized using Design Compiler (DC) and imported into Cadence Encounter for Place and Route (P&R). The technology used in this project was 65nm CMOS.

3.2 Encounter Clock Tree Synthesis

There are two ways of performing clock tree synthesis in Encounter which could be recognized as auto-CTS and manual-CTS. For both of the CTS mode, the flow is the same and shown in Fig. 3.2.

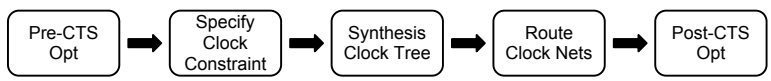


Fig. 3.2: Encounter CTS flow

Setup time, hold time together with the design rule check (DRC) are analyzed in pre-CTS stage. Techniques such as resize gates, move instances, add buffers, etc are used in the optimization.

In the second stage, clock timing constraint file is loaded. The constraint file differs from auto-CTS and manual-CTS. For the manual-CTS, the user constructs the clock tree manually by specifying the topology of the tree. An example constraint file is shown in table 3.1.

Table 3.1: Example code for manual CTS constraint file

ClockNetName	*clockRootNetName*		
LevelNumber	*NumberOfClockTree*		
LevelSpec	*LevelNumber*	*NumberOfBuffer*	*BufferName*
PostOpt	*YES*		
OptAddBuffer	*YES*		
End			

ClockNetName points to the root net name of the clock signal source. LevelNumber specifies the level of clock tree to be constructed. LevelSpec is the command to tell the tool that how many and what kind of buffer should be used in each level of the tree. Each level must be specified. PostOpt means allow the tool to do post-CTS optimization and OptAddBuffer means allow the tool to add buffers while optimizing the synthesized clock tree. Auto-CTS, on the other hand, proceed in another way. The constraint file contains timing requirements such as skew, slew and delay, etc that the clock tree should meet, buffers can be used and options should be followed. The tool will try to find the best combination of the clock tree structure that fulfill the requirements.

The last three stages are performing the clock tree synthesizing, clock network routing and optimization. The clock network routing can be included in the synthesis stage if the user want. It has been set as the default that routing the clock network with the CTS routing guide file to improve the pre-route-post-route correlation. This gives a better match but with a penalty of longer runtime. For the post-CTS optimization, max capacitance violation, max transition violation and max fan-out load violation has been fixed for this particular design. And the timing has been fixed for both setup and hold time.

3.3 SPICE Netlist Simulation

In order to have a more accurate estimation on the performance of the synthesized clock tree, SPICE netlist simulation is adopted. The flow is shown in Fig. 3.3.

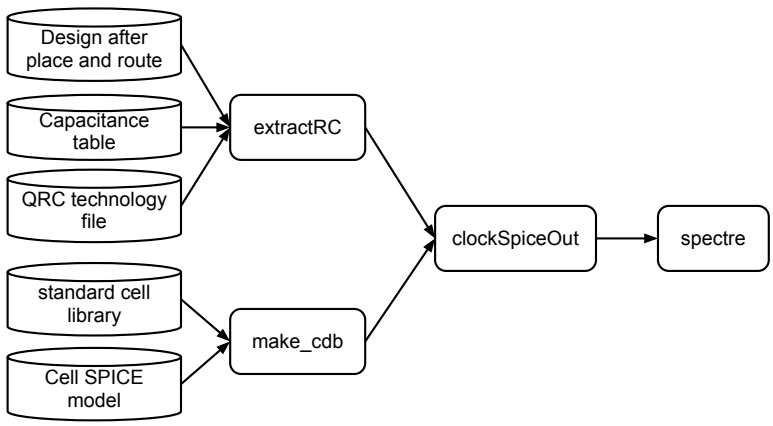


Fig. 3.3: SPICE simulation flow

Before writing out the netlist, two things need to be prepared. First is to extract the parasitics of the routed design which can be done by feeding the capacitance table together with the QRC techfile to the tool and run the command extractRC. The second thing is noise library characterization. Standard cell library together with the SPICE model are adopted in order to perfrom noise calculation. A noise library which has the extension of .cdb is generated. One thing to note is that the tool only accepts a spice model netlist. The command clockSpiceOut generates the clock tree SPICE netlist containing all parasitics. This netlist is then fed into a SPICE simulator e.g.) *Cadence Virtuoso Spectre Circuit Simulator* to perform the simulation.

3.4 Library Characterization

Since the design needs to be simulated in sub-V_T region, libraries for sub-V_T are needed and they are generated by library characterization. A comercial tool called *Cadence Encounter Library Characterizer (ELC)* is used in this project. The flow is shown in Fig 3.4.

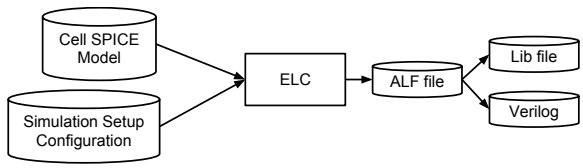


Fig. 3.4: ELC Library Characterization flow

SPICE models and simulation configuration setup files are input to the ELC tool. The output Ambit Library format (alf) file which is the technology library is then converted into timing library file and verilog file.

Although the flow is fully automated, basic setup still needs to be performed. The idea of characterization is shown in Fig. 3.5.

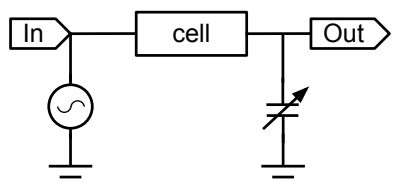


Fig. 3.5: Library Characterization method

In order to perform characterization, suitable simulation input for spice measurements needs to be setup [11]. Under the load, characterization is performed as shown in Fig. 3.5. By measuring all the combinations of the user provide input slew and load, a timing look-up-table (LUT) could be generated and further written into a standard liberty format.

The timing information in the look-up-table thus models the cell timing. When EDA tool looks for timing information of a particular cell, it takes the input slew and output load that the cell experienced in that design and looks for the best match in the look-up-table in the library. Interpolation will occur since there is only small chance that the cell has an operation condition exactly the same as one of the combinations in the table. However, extrapolation may also occur in some cases. Interpolation is usually not a big problem because the value which has been interpolated is within the range of the LUT and it would usually has an acceptable quality. Extrapolation, however, results in timing values outside the LUT boundary

and probably deviates severely from the real value. Due to this fact, the boundary of the input slew and load capacitance should be selected carefully.

Since there exist fully functioning library for super-V_T, temperature slew and load information has been read out from it. Input slew is degraded by a degrading factor while keeping output load unchanged. From this, a setup file is generated for all cells. The degrading factor was found by measuring the timing degrading of some cells in the library manually. Although the resulting library has pretty good quality, the extrapolation problem still exist sometimes. So an idea came up to solve this problem and it can be explained in Fig. 3.6.

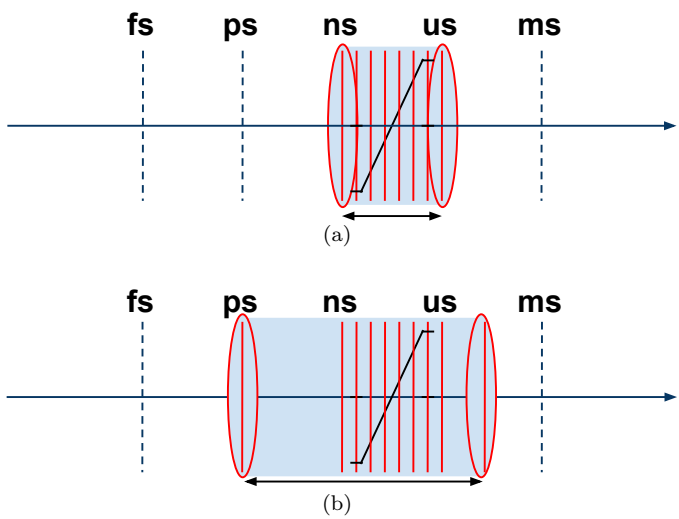


Fig. 3.6: Illustration of library characterization covering range: (a) Common library characterization range. (b)Reduced extrapolation characterization range.

Suppose that the cell timing locates in the range of few μs to few hundreds of μs as shown in Fig. 3.6(a). The conventional input slew range would probably looks like Fig. 3.6(a) and thus covering only the range of micro second. Anything outside this range would need extrapolation and results in inaccuracy. The idea was to add two extension values into the slew range as shown in the lower plot. This will improve the covering range and reduce the probability of extrapolation as shown in Fig.3.6(b). However, the penalty of this method is increased runtime since more combinations need to be simulated. Nevertheless, it is worthy to do if increasing a acceptable amount of the runtime could result in much higher probability of better quality.

3.5 Customizing Clock Buffer

There exists four sets of buffer/inverter in the current technology library which are shown in table 3.2.

Table 3.2: Available buffer/inverter sets

CNBFX10∼CNBFX124
CNIVX3∼CNIVX124
BFX2∼BFX284
IVX2∼IVX284

First of all, the number in the buffer/inverter name indicating the driving strength of the cell. Besides, the name CNBF and CNIV indicating that the cell is particularly designed for clock network. The rest of two groups, on the other hand are ordinary buffer and inverter.

The main difference between clock buffer/inverter and ordinary buffer/inverter is that some extra PMOS transistors have been added in the PUN. These extra PMOSes in the clock buffer/inverter enhance the PUN strength which results in better balanced timing. A comparison of schematic is shown Fig. 3.7. In this project, only clock buffer and no inverter has been used for CTS and customization.

As introduced in section 2, the dual-V_{T} method is employed in buffer customization. The steps for customizing a buffer are listed in table 3.3.

Table 3.3: Customization steps

1. Determine the goal voltage to perform customization.
2. Remove extra PMOS transistor.
3. Sweep PMOS transistor size.
4. Measure rise and fall time and determine transistor size.
5. Layout drawing.
6. Post-layout simulation.

Since the performance of customized buffer is strongly affected by V_{DD}, the target voltage should be determined at the very beginning of customization as shown in step 1).

By replacing the PMOS to lower threshold equivalent, the PUN strength has already been increased which makes the extra PMOS redundant. Thus, extra PMOS are not necessary in the PUN which are removed in step 2).

In step 3), sweeping width or length is the method of adjusting strength for PUN and PDN. Since a buffer contains two inverter stages, the transistor sizes of the two stages are relational in impacting buffer timing. Therefore, transistor size of the PMOS for both stages should be swept simultaneously. Also, different strategy should be applied when dealing with different size of buffers. For a big

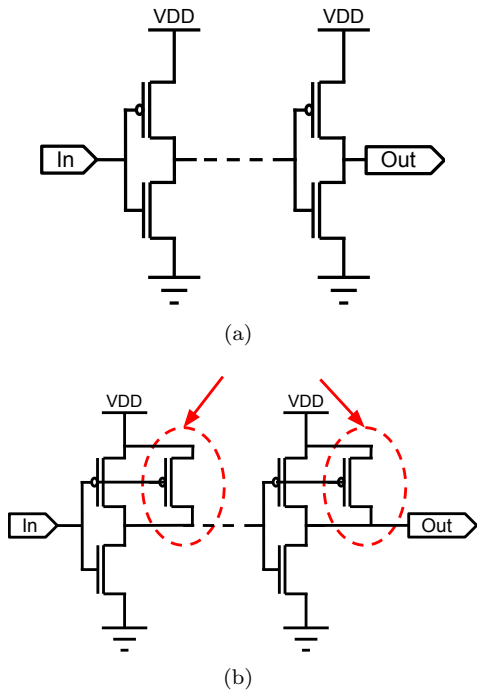


Fig. 3.7: Schematic comparison between ordinary buffer and clock buffer: (a) Ordinary buffer schematic without extra PMOS. (b) Clock buffer with extra PMOS.

buffer as CNBFX103, a cell which has the driving strength of driving 103 loads, sweeping the transistor width for both inverter stages is enough to find out the balance point. However, sweeping the width is not capable of finding the balance point for small buffer such as CNBFX10, a buffer only suitable for driving 10 or less sinks. In order to solve this problem, one can either stack one more PMOS in the PUN and sweep the transistor width or sweep the transistor length instead. It has been found that sweeping transistor length is more effective than stacking. After determining the transistor size, one can switch to layout drawing phase and do the extraction for layout and further verifying the customized buffer performance via post-layout simulation.

Chapter 4

Experiments and Results

Experimental results including comparison between super-V_{T} CTS and sub-V_{T} CTS, comparison between evenly and non-evenly distributed design CTS are included. SPICE simulation has been applied in order to verify the accuracy of the results got from Encounter. Furthermore, comparison between customized clock buffer and standard cell buffer is presented. Finally, comparison for buffer grouping and MMMC are shown and discussed.

4.1 Super-V_{T} CTS versus Sub-V_{T} CTS

It has been studied in [12] that performing synthesis at super-V_{T} region results in better timing and energy efficiency for sub-V_{T} systems [12]. However, clock tree performance has not been studied specifically. Thus, a comparison experiment has been carried out for comparing the performance of super-V_{T} CTS and sub-V_{T} CTS.

A few things need to be mentioned here for understanding the result, which are the clock tree constraints and data normalization. Four times of fan-out four (4FO4) delay has been used in constraining the clock tree, i.e. constraining the skew, slew, etc. The reason of choosing such criterion is because that this represents the balance point between clock tree energy consumption and speed [2]. For the normalization, 4FO4 delay is again used. The data acquired at each voltage has been divided by 4FO4 delay at that voltage. As shown in 4.1

$$value/(4FO4(V)), \tag{4.1}$$

where 4FO4(V) representing that this delay is a function of supply voltage. To be clear, data acquired at 1.2V will be divided by 4FO4 delay at 1.2V and data acquired at 0.4V will be divided by 4FO4 delay at 0.4V, etc. Due to the reduction of supply voltage, the 4FO4 delay differs from each other for different voltage.

The reason for using such a mutative normalization factor is because that reducing supply voltage results in reduction of circuit speed, the comparison of the absolute value for the clock tree performance between different voltages is not possible. The simulated result for comparing these two options of performing CTS is shown in Fig. 4.1.

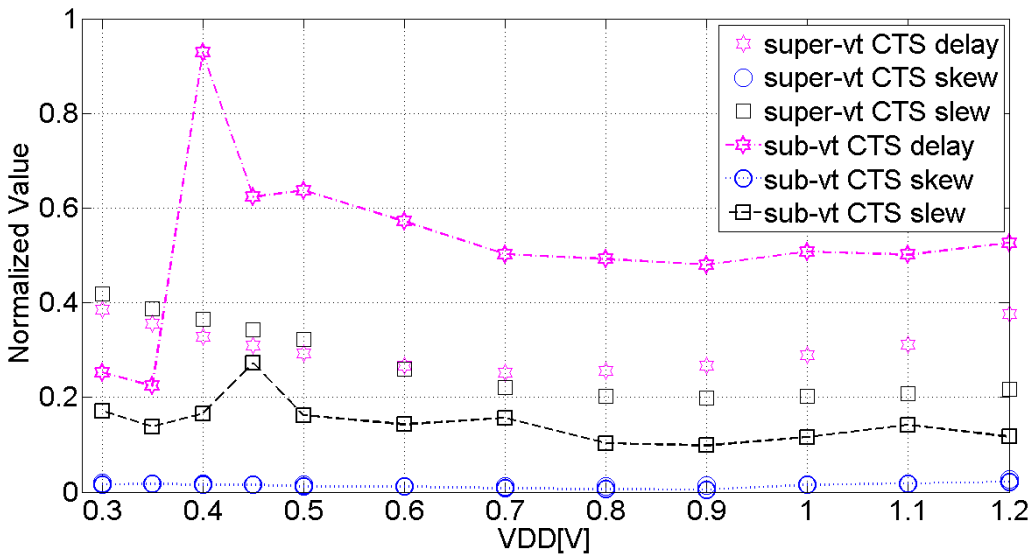


Fig. 4.1: One level clock tree comparison between super-V_T CTS and sub-V_T CTS

Firstly, CTS was performed at 1.2V and the design re-opened with libraries characterized for lower V_{DD}. Clock skew, slew and delay for the performance of the clock tree have been recorded for each voltage. Secondly, acquired skew, slew and delay are used as constraints for performing sub-V_T CTS for each V_{DD}. The results are shown in Fig. 4.1.

The skew, slew and delay curves for super-V_T CTS are smoother and continuous than the sub-V_T CTS. Due to the fact that the structure of the tree does not change, only the operating V_{DD}, which only has a more linear impact. The response for the same clock tree is thus a continuous function to the supply voltage as we discussed in section 2. For performing CTS in the sub-V_T domain, different structures of the clock tree are constructed for different V_{DD}, e.g. performing CTS at 500mV results in a clock tree consisting of 16 CNBFX83 and 58 CNBFX103 buffers. However, performing CTS at 300mV results in 2 CNBFX103 and 22 CNBFX124 buffers.

On the other hand, the skew for both cases are about identical. Slew, however, sees a 63% improvement when performing CTS at a scaled V_{DD} of 0.35V and an average of 30% improvement. For the delay, performing CTS at the same voltage

results in worst case of a degradation of 180% and average of 90% degradation. At 0.3V and 0.35V, however, the delay is improved. Another metric of interest is the total drive strength plot.

A clock tree constructed in the super-V_T CTS flow consisting 4 CNBFX103 buffers and the total drive strength is 412. After CTS of scaled V_{DD}, the structure of the clock tree varies and thus the total strength, which is shown in Fig. 4.2.

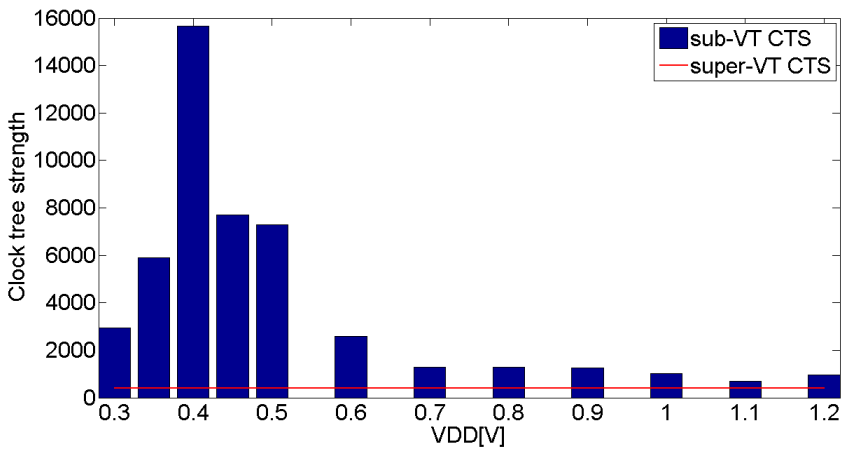


Fig. 4.2: Total buffer strength comparison for the ways of performing CTS

The line in the plot indicates 412 and the bars represent the total buffer drive strength for each V_{DD}. As seeing in the figure, the tool tends to use a lot more buffers in sub-V_T region. The maximum occurs at 0.4V where the tool uses 25 CNBFX83 and 132 CNBFX103 buffers which is about 40 times more buffers than that in the super-V_T CTS flow. The increased amount of buffers thus explain the improvement of slew, as increased clock network strength results in increased total charging current that the tree can deliver, and thus increases the charging speed at the sink node. However, using this many buffers is not desired because of the increase in energy disipation and increase area. So, performing CTS flow at super-V_T region is preferred.

The clock tree that has been discussed so far is only one level. Fig. 4.3 4.4 4.5 4.6 show the corresponding plots for clock tree with 2 to 5 levels.

For all cases, clock skew has similar values, on the other hand, the clock tree delay increases dramatically when increased the clock tree level. It is explained as the path from clock root pin to the sink is deeper when increases the level. For the clock slew, the super-V_T CTS experience worse clock slew than performing CTS at scaled V_{DD} directly. This relates to the number of buffer that has been used in the clock tree. As similar behavior is seen for both multi and single level clock tree, only single level is studied thenceforth.

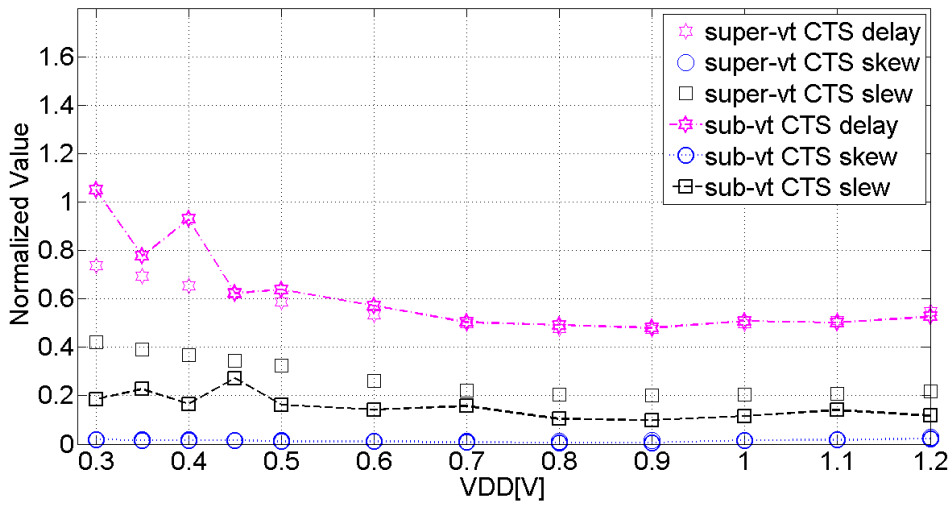


Fig. 4.3: Two level clock tree comparison between super-V_T CTS and sub-V_T CTS

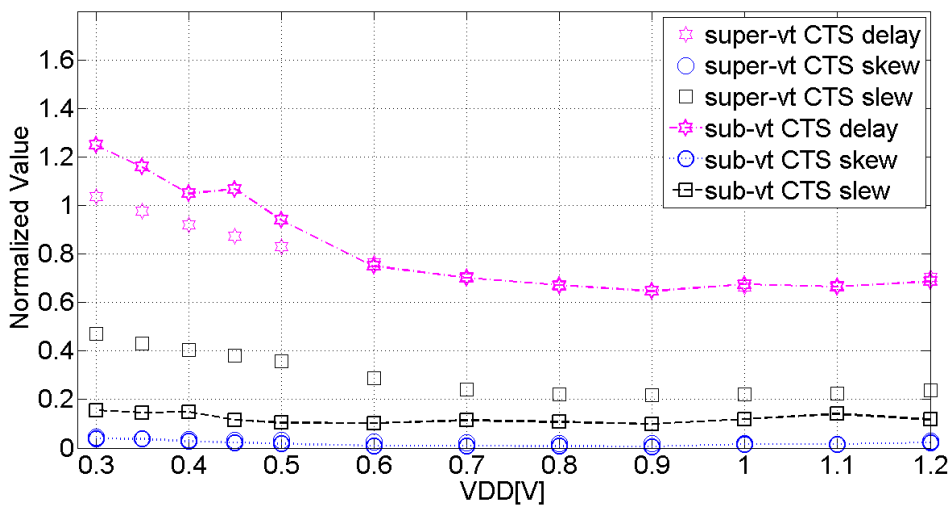


Fig. 4.4: Three level clock tree comparison between super-V_T CTS and sub-V_T CTS

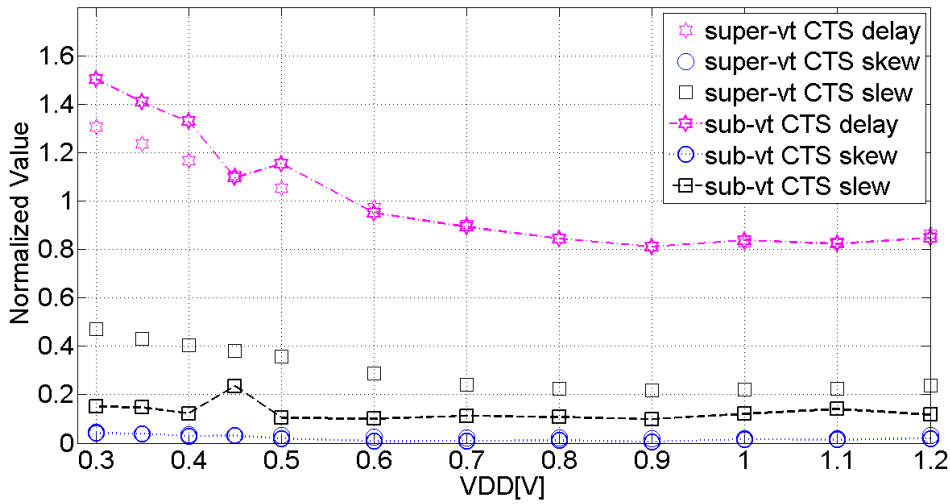


Fig. 4.5: Four level clock tree comparison between super-V_T CTS and sub-V_T CTS

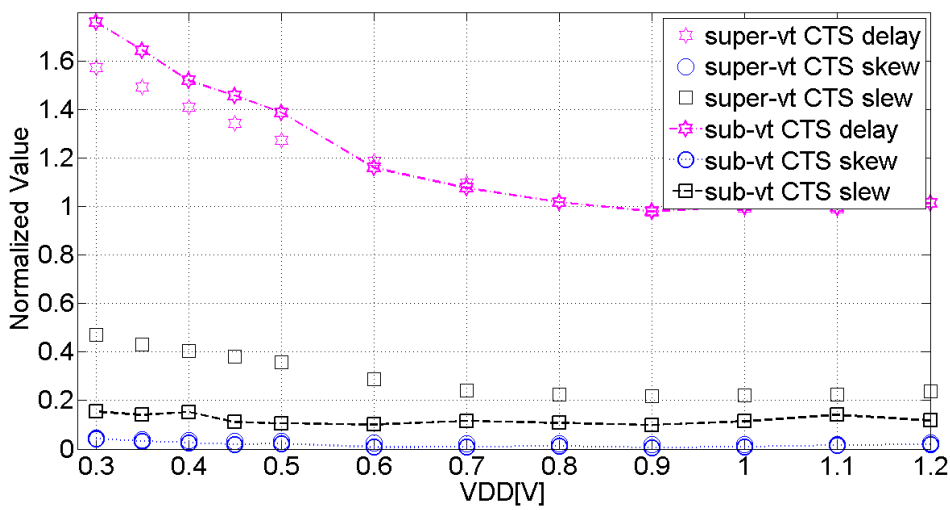


Fig. 4.6: Five level clock tree comparison between super-V_T CTS and sub-V_T CTS

4.2 Evenly Distribution of Sinks

To evenly distribute all the sinks is another topic that has been investigated in the project. The idea is to force the tool to distribute all the registers evenly over the physical core area and investigate if the symmetry of the distribution makes any difference to the clock tree performance, especially clock skew. The corresponding result is shown in Fig. 4.7, as seen, the evenly distributed design does not show

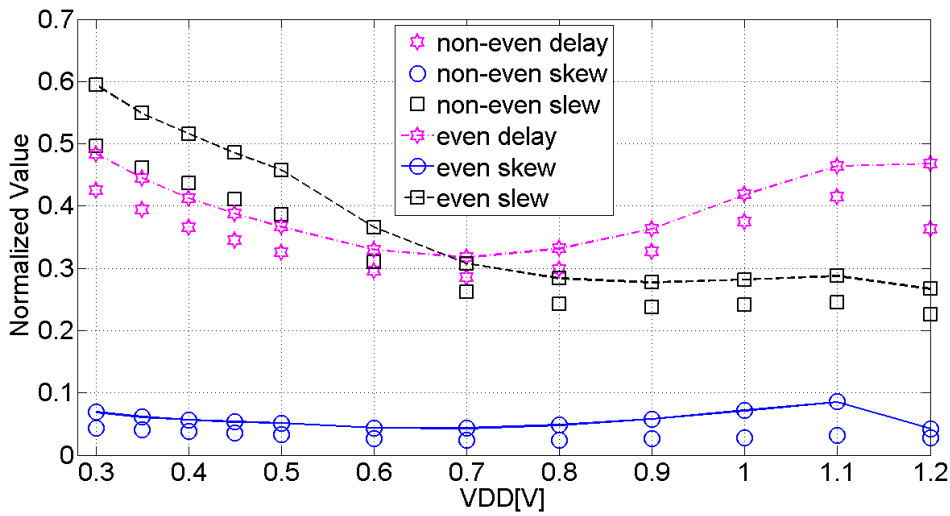


Fig. 4.7: Clock tree performance comparison between evenly distributed design and non-evenly distributed

improve performance. All the three parameters have worsened. Thus, it can be concluded that the evenly distribution of sinks does not improve performance.

4.3 Encounter Simulation, SPICE Simulation and SPICE Netlist Simulation

In order to verify the accuracy of the model in the timing library, SPICE simulation has been carried out. Additionally a comparison is presented between Encounter simulation, SPICE simulation on schematic level and SPICE netlist simulation on layout level with parasitics.

For the schematic simulation, the clock tree generated from Encounter was constructed manually in schematic format in Cadence virtuoso. As no wire parasitic will be included the tree performance is ideal. SPICE netlist on layout level with

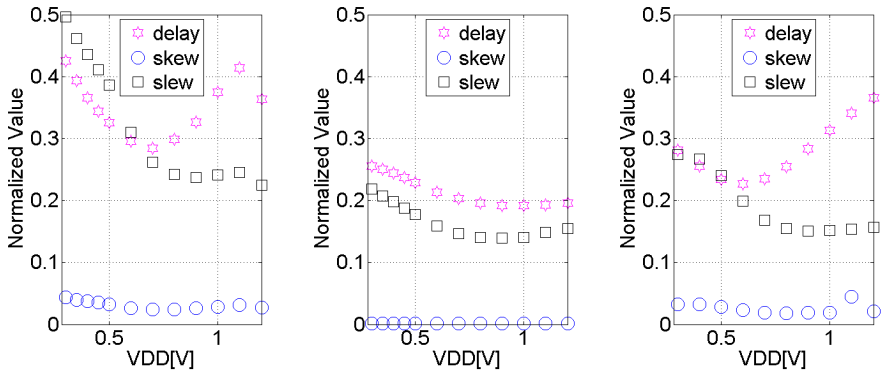


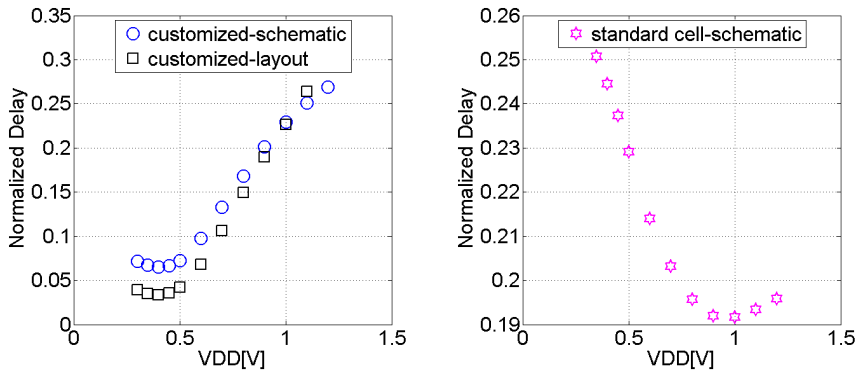
Fig. 4.8: Cross comparison for: (a), Encounter simulation, (b), SPICE simulation and (c), SPICE netlist simulation

parasitics, on the other hand, contains all the parasitic components additionally and shows a more realistic behaviour. Results are shown in Fig. 4.8.

Comparing between Encounter and SPICE layout level gives maximum of 34% relative difference in delay, 45% of relative difference in slew and 26% of relative difference in skew. Although there are such differences, the curve shapes are similar with each other. Also, the continuity of curves in SPICE layout simulation is better than the one in Encounter which implies a better SPICE model is than the timing model in Encounter.

For the schematic simulation in Fig. 4.8(b), skew differs with the other two plots. This is due to the non-existence of wire parasitic. Because the schematic does not include any wire resistance and capacitance, the only clock path difference is the mismatch of transistors. The majority of the skew originates from clock path parasitic, i.e. wire parasitic [6], so the skew in this simulation are almost zero. For the slew curve, it has almost the same shape as the one in SPICE plot. This is because that slew is the charging time needed at the sink node and relates much to the node capacitance. Although parasitics add a few more node capacitance into the total node capacitance, the majority of the sink node capacitance is contributed from D-flipflops (DFF), i.e. when $C_{\text{wire}} \ll C_{\text{sink}}$. Thus, there exists only a small difference between schematic simulation and SPICE simulation.

By comparing the delay curve, it can be found that without parasitics results in the lowest delay as seen. Adding parasitics to the circuit degrades performance. Also, there is a minimum point in the delay curve at around 1V. Since the standard cell that has been used so far has been optimized for 1V, the cell would thus have best balance timing at 1V. When increasing or decreasing the supply voltage, timing



(a) (b)

Fig. 4.9: Normalized delay for: (a), customized clock buffer post-layout simulation for customized clock buffer (b), standard cell buffer

imbalance will occur and results in degraded performance.

4.3.1 Delay Curve Min-point Analysis

Another comparison has been carried out in order to explain this issue. Fig. 4.9 shows the delay for customized clock buffer and standard cell buffer. The customized clock buffer has employed dual-V_T method and has been optimized to have balance timing at 400mV. Both the schematic and post-layout simulation has almost the same behavior as can be seen in Fig. 4.9(a). The reason that layout performs better than schematic will be discussed in the following section. Here only focus on the minimum point of the curve. As seen, the customized clock buffer has a minimum point around 400mV which is the voltage that the cell has been optimized. Although the parasitic will shift this minimum point somehow, this phenomenon is a character of the cell.

One last thing to be mentioned is that appendix A shows the SPICE simulation for the clock tree of evenly distributed sinks. The plot results in the same conclusion that has been drawn in the previous section.

4.4 Corner Simulations

Up until now, the simulations are done for the typical typical (TT) case. However, process variation is another important factor to consider as the effects of process variations are larger in the sub-V_T region. Thus, corner simulations are necessory to see how the circuit performance varies with respect to different corner.

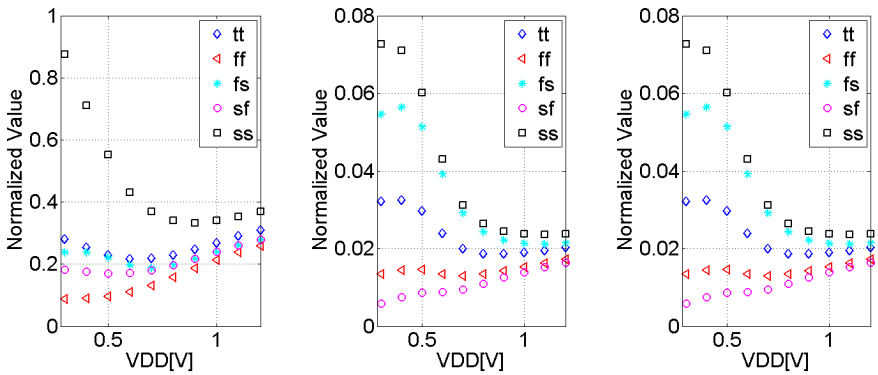


Fig. 4.10: Corner simulation for: (a), Delay (b), Skew (c), Slew

By using SPICE models for different corner into the make cdb step mentioned in section 3.3, clock tree SPICE netlist for different corners can be generated. The generated netlist is fed to *Spectre Simulator*. In Fig. 4.10 the results are shown, i.e., delay, skew, slew. Five corners has been simulated, fast fast (FF), fast slow (FS), slow fast (SF), slow slow (SS) and TT. The first letter in the abbreviation indicates the speed of NMOS transistor and the latter corresponds to PMOS. So a cell in FS corner contains fast NMOS in the PDN and slow PMOS in the PUN.

It is seen that process variation are much larger in sub-V_T. The variation of the three parameters at 400mV are 5.6×, 7.8× and 8.6× compared to 1.2V. These values are calculated by subtracting the max-data by the min-data of 400mV and divide by the one got from 1.2V. This increasing variation makes the super-V_T CTS method unreliable since the circuit would have less probability to meet the timing constraint when reducing supply for different corners. Therefore, the process variation should be reduced to increase the reliability of the circuit.

4.5 Customized Clock buffer

A customized clock buffer has been developed to improve the reliability of the clock tree. Since the one level tree only consists of 4 CNBFX103 buffer, e.g.) a customized buffer with identical drive strength is created. The buffer is aimed to have balanced timing at 400mV.

4.5.1 Layout

Fig. 4.11 shows the layout of the buffer. Because of the selection of SVT transistors in the PUN, the width of the PMOS transistor is 4× shorter than NMOS which reduce the area of the cell and relevant parasitic. Not only because of the reduction of area but also due to the fingering in the layout which reduces parasitics even more. Consider that the schematic use separate transistor model instead of fingered transistor, the schematic model thus have more parasitics than the layout. This is the main reason that layout has better performance than the schematic shown in Fig. 4.9.

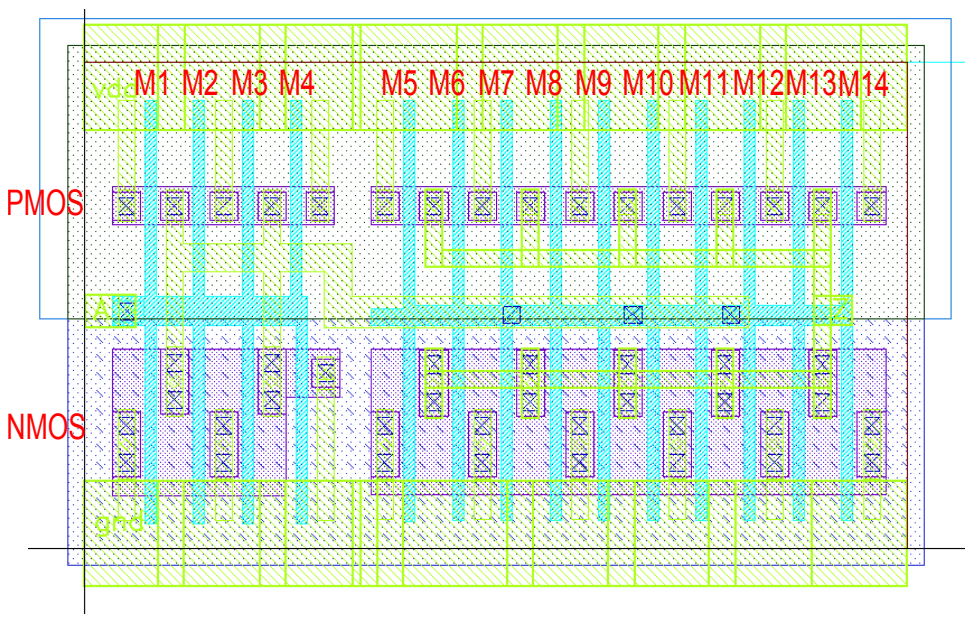
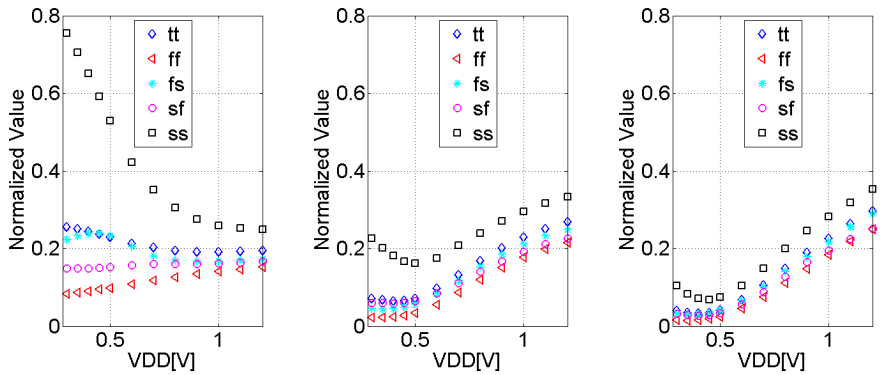


Fig. 4.11: Layout of the customized clock buffer

4.5.2 Post Layout Simulation

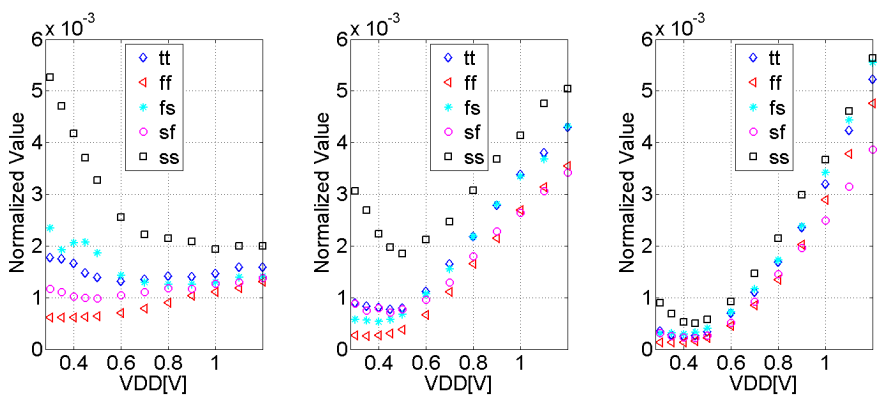
After completing the extraction of layout, post-layout simulation has been carried out and the simulation results are shown in Fig. 4.12 ∼ Fig. 4.14.

Delay, skew and slew comparison are separated for clarity. In all the three plots, the left one is for the standard cell buffer, the middle one is for schematic of the customized clock buffer and the right one is for layout of the customized clock buffer. A huge improvement is seen for the customized clock buffer, especially at the post-layout. At 400mV, the customized clock buffer improves the performance of the concerning parameter by 9.8×, 8.9× and 9.8× respectively. Worth noting is



(a) (b) (c)

Fig. 4.12: Corner simulation for delay: (a), Standard cell buffer (b), Customized buffer schematic (c), Customized buffer post-layout



(a) (b) (c)

Fig. 4.13: Corner simulation for skew: (a), Standard cell buffer (b), Customized buffer schematic (c), Customized buffer post-layout

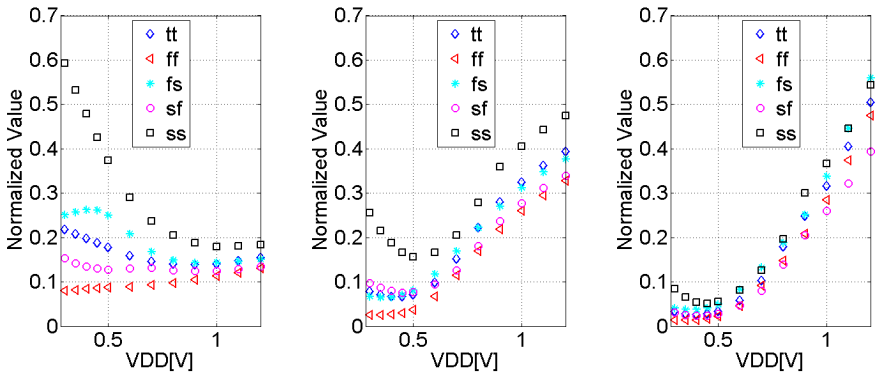


Fig. 4.14: Corner simulation for slew: (a), Standard cell buffer (b), Customized buffer schematic (c), Customized buffer post-layout

that, the minimum point of the curve has shifted to 400mV as expected. For super-V_T region, the cell experiences more timing imbalance than before that results in degraded performance compare to a standard cell buffer. However, although the buffer was aimed for 400mV, it still achieves acceptable performance up to 800mV as shown in Fig. 4.12 4.13 4.14. To conclude, the customized clock buffer reduces the effects of process variations dramatically.

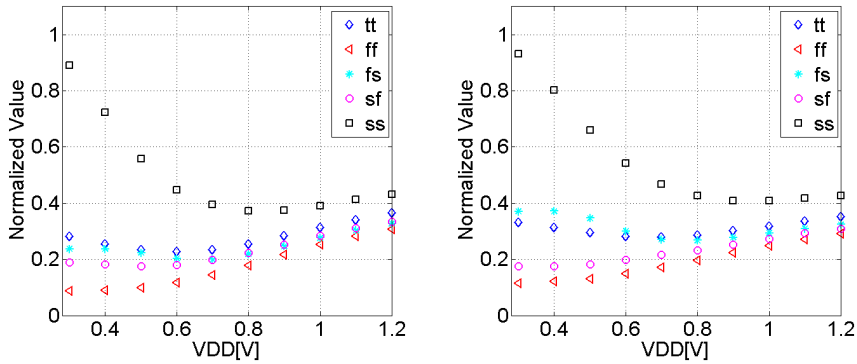
4.5.3 Energy Comparison

It is widely known that speed and energy always is a trade-off [6]. Table 4.1 the different power between the two clock trees constructed by standard-cell buffers and customized clock buffers.

Table 4.1: Power comparison for two clock trees

V_{DD}	1.2V		0.4V	
Buffer	customized	standard-cell	customized	standard-cell
Switching power	17.703μW	18.519μW	8.992nW	9.446nW
Internal power	16.968μW	7.391μW	7.556nW	8.754nW
Leakage power	1.144nW	0.504nW	93.440pW	42.32pW
Total power	34.673μW	25.92μW	16.642nW	18.243nW

Leakage power for customized buffer increases to twice as much as the standard-cell buffer in both 1.2V and 0.4V due to the fact of using SVT PMOS in the PUN as shown in section 3.1. The internal power for customized buffer increases at



(a) (b)

Fig. 4.15: Corner simulation for delay comparison: (a), CNBFX10 standard cell buffer (b), CNBFX103 standard cell buffer

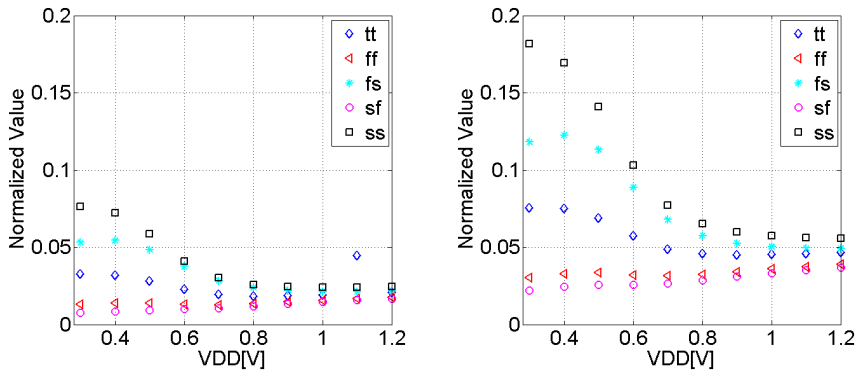
1.2V is because SVT PMOS results in longer short circuit conducting. At 0.4V however, both PMOS and NMOS are in the weak inversion region which results in comparable internal power between customized and standard-cell buffer. Since switching activity of the circuit are the same for both clock trees at the same V_{DD}, the switching power are about the same for both cases. In short, the customized buffer could improve clock tree perfromance at sub-V_{T} region while maintaining almost same amount of power consumption as the standard-cell buffer.

4.6 Minimum Strength Buffer Comparison

So far, only big buffers have been used in the clock tree, CNBFX103 is the second largest buffer availabel in the clock library. The idea of this section is to limit the tool to use the min-strength buffer and see how the performance differs from the previous clock tree. The main idea is investigating which range the performance of this set of buffer gives.

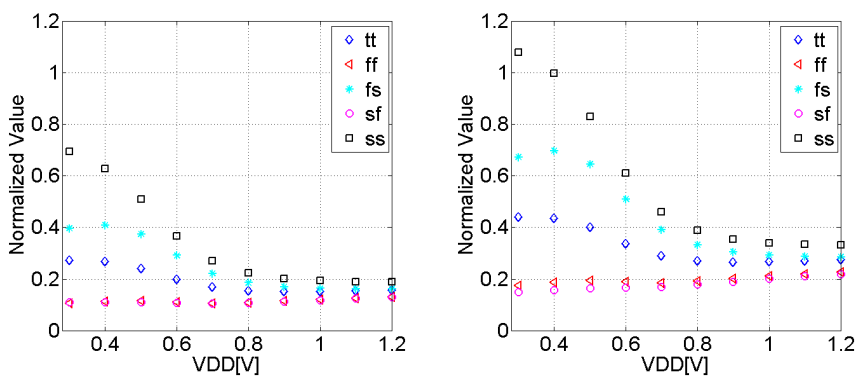
4.6.1 Clock Tree SPICE Simulation

The results are shown in Fig. 4.15 4.16 4.17. Fig. 4.15(a) 4.16(a) 4.17(a) corresponds to the clock tree constructed with CNBFX103 and Fig. 4.15(b) 4.16(b) 4.17(b) corresponds to clock tree constructed with CNBFX10. At 400mV, the min-strength tree experiences an increase of 1.07×, 2.26× and 1.62× for delay, skew and slew variation respectively.



(a) (b)

Fig. 4.16: Corner simulation for skew comparison: (a), CNBFX10 standard cell buffer (b), CNBFX103 standard cell buffer



(a) (b)

Fig. 4.17: Corner simulation for slew comparison: (a), CNBFX10 standard cell buffer (b), CNBFX103 standard cell buffer

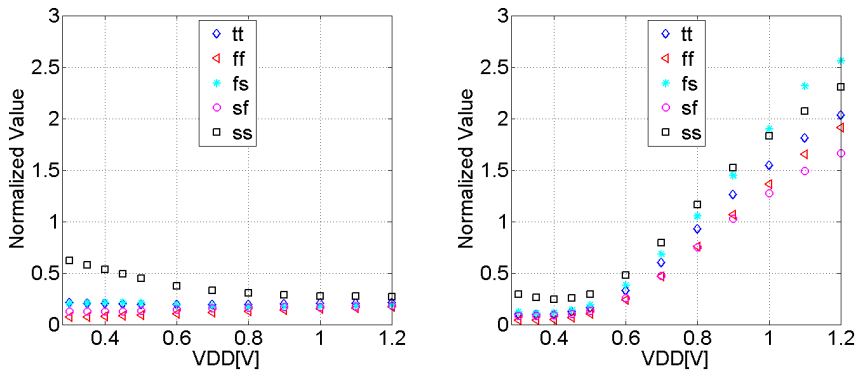


Fig. 4.18: Corner simulation for delay comparison: (a), CNBFX10 standard cell buffer (b), CNBFX10 customized clock buffer

4.6.2 Customized Buffer

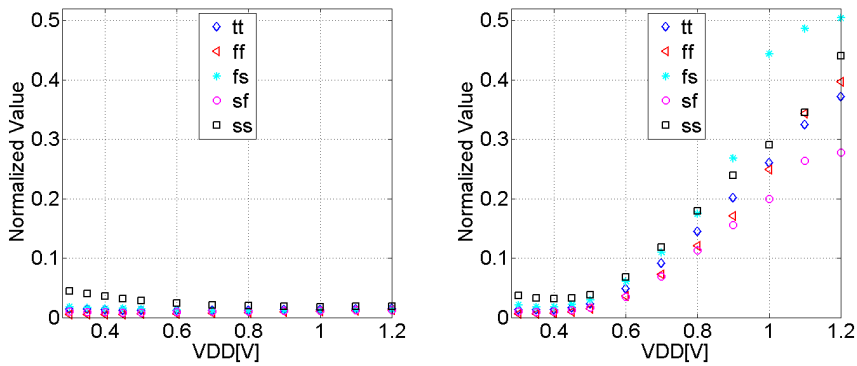
The customization method has been performed on the min-strength buffer as well. The results are shown in Fig. 4.18 $\sim$ Fig. 4.20. The three parameters has been improved by 2.3×, 1.25× and 1.86× respectively. This improvement is not big as compare to the customized CNBFX103 buffer. And the performance degrades dramatically when supply voltage is increased. The variation that customized buffer experiences is 9×, 32× and 13.4× for delay, skew and slew when comparing to the standard cell bufferat 1.2V. The absolute values of delay, skew and slew increases to 9.5×, 26.9×, and 17.3× compare to the standard CNBFX103 buffer as well. Therefore, base on all the facts that has been presented in this and the previous one section, small buffer is not worthy to be customized.

4.7 Clock Buffer Groups

As mentioned in chapter 3, different buffers would have different driving strength. The phrase buffer grouping is to group buffers base on their driving strength. Also, because inverters are often used in constructing clock tree, it might be good to have inverters in some of the comparison group. Therefore, the groups are divided as shown in table 4.2.

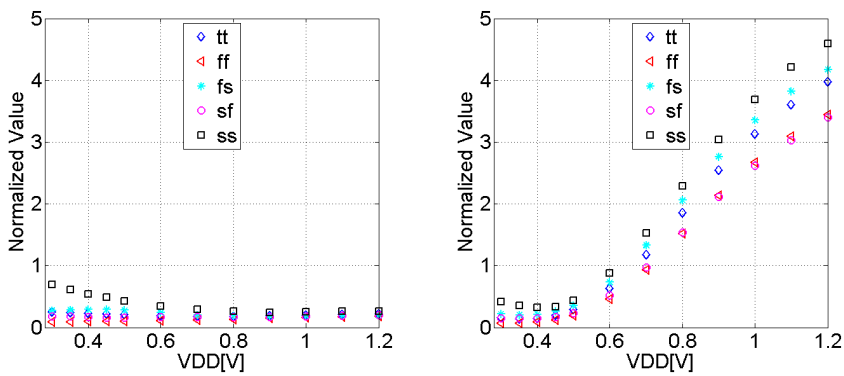
Since it might be useful to have small buffer in the design, thus the last group is included. The clock tree level is limited to 1 as discussed in section 4.1.

In Fig. 4.21 4.22 4.23, the solid line in each plot represents the performance of the CNBFX103 clock tree. As can be seen in all the plots, not much difference



(a) (b)

Fig. 4.19: Corner simulation for skew comparison: (a), CNBFX10 standard cell buffer (b), CNBFX10 customized clock buffer

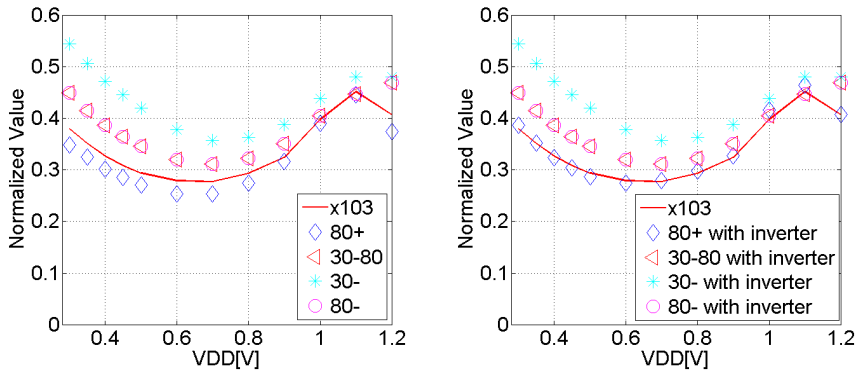


(a) (b)

Fig. 4.20: Corner simulation for slew comparison: (a), CNBFX10 standard cell buffer (b), CNBFX10 customized clock buffer

Table 4.2: Buffer grouping

Buffers Only	Buffers + Inverters
X10∼X30	X10∼X30
X30+∼X80	X30+∼X80
X80+	X80+
X80-	X80-



(a) (b)

Fig. 4.21: Delay comparison for single level clock tree by buffer grouping: (a), Groups without inverter (b), Groups with inverter

that inverters could do to the performance. The only improvement is in the skew comparison, group 80+ with inverter has 58% fewer skew than that without inverter at 400mV and below. Furthermore, increasing the buffer strength will always result in better performance in all cases.

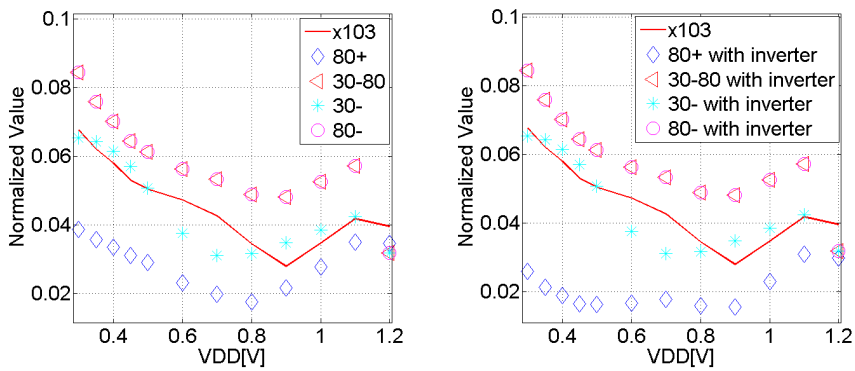
Finally, table 4.3 lists of all the used buffers for different buffer grouping.

Table 4.3: Buffers used in different clock tree generated by buffer grouping

buffer group	single level structure
X10∼X30	7*X27+5*X24
X30∼X80	2*X58+2*X62+X55+X52
X80+	4*X103
X80-	2*X62+2*X58+X55+X52
X10∼X30 with inverter	7*X27+5*X24
X30∼X80 with inverter	2*X62+2*X58+X55+X52
X80+ with inverter	4*X124
X80- with inverter	2*X58+2*X62+X55+X52

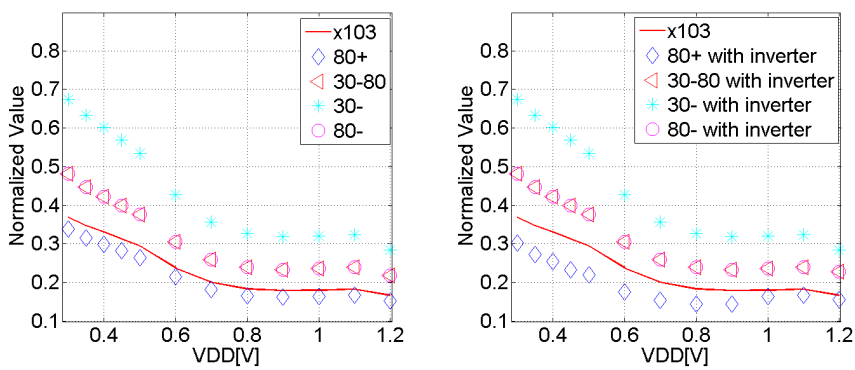
4.8 Multi-Mode Multi-Corner

Multi-Mode Multi-Corner (MMMC) takes into account multiple process corners when performing the timing analysis in the flow. Simulation has been carried out in order to see if MMMC improves the performance, the results are shown in Fig. 4.24.



(a) (b)

Fig. 4.22: Skew comparison for single level clock tree by buffer grouping: (a), Groups without inverter (b), Groups with inverter



(a) (b)

Fig. 4.23: Slew comparison for single level clock tree by buffer grouping: (a), Groups without inverter (b), Groups with inverter

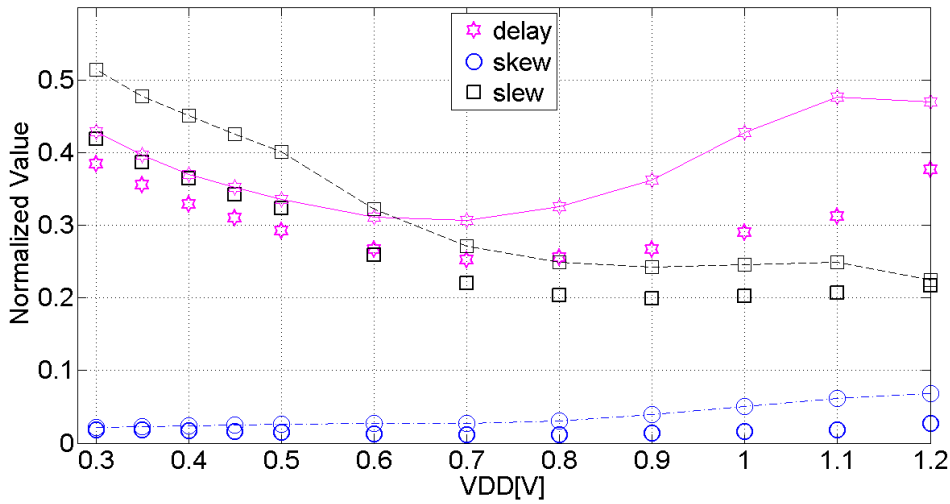
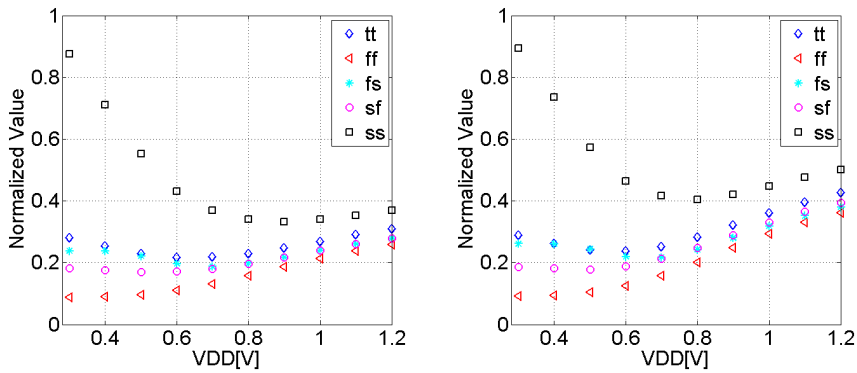


Fig. 4.24: MMMC simulation result

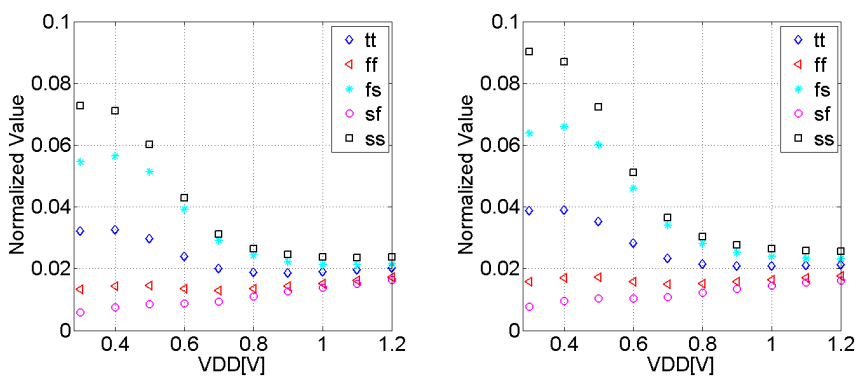
Different process corners has been setup in the design configuration file and loaded when setting up the place and route setup. Therefore, the tool has knowledge of the timing of these corners and tries to construct a clock tree which functions well in all the corners. After generating the clock tree, the clock tree performance for TT corner has been used in the plot as the source data for MMMC clock tree. Different voltage libraries are again loaded into the saved design and output the timing for the clock tree respectively. As seen in Fig. 4.24, MMMC does not help improve timing.

Corner simulation for the generated clock tree has been done as well and results are shown in Fig. 4.25 4.26 4.27. For all the three parameters, MMMC does not help improving the reliability for sub-V_{T}.



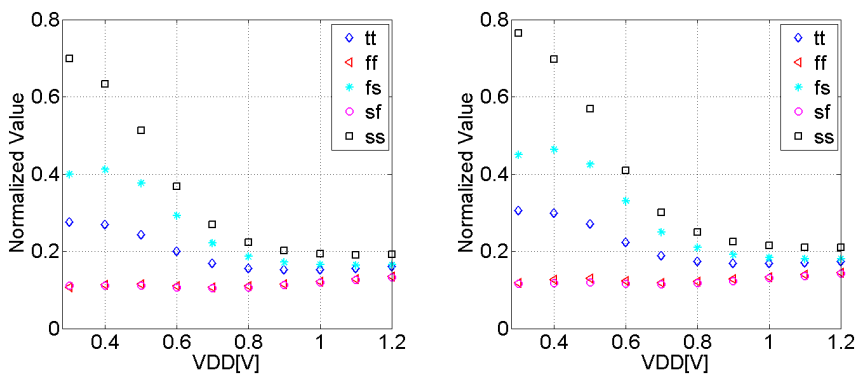
(a) (b)

Fig. 4.25: Corner simulation of delay for MMMC: (a), Without MMMC (b), With MMMC



(a) (b)

Fig. 4.26: Corner simulation of skew for MMMC: (a), Without MMMC (b), With MMMC



(a) (b)

Fig. 4.27: Corner simulation of slew for MMMC: (a), Without MMMC (b), With MMMC

Chapter 5

Conclusion

This thesis project has presented several comparison work on constructing clock tree for sub-V_T. It has been simulated that doing CTS at super-V_T is much more worthy than doing CTS directly at sub-V_T region. Also, one level clock tree with max-fan-out constraint is preferred than multi-level structures. Evenly distribution of all registers in the design, buffer grouping and MMMC turns out not helpful improving performance. The most effective way of improve the robustness of the clock tree in sub-V_T is developing customized buffers. Although small buffer as CNBFX10 is not worthy customizing, big buffer such as CNBFX103 improves the performance dramatically when applying dual-V_T customization. Power estimation has been employed and shows that the customized buffer improves the performance dramatically while keeps the power consumption the same magnitude as the standard cell buffer.

In short, sub-V_T CTS should be done in super-V_T region. Single level clock tree and large customized buffer should employed in the design to ensure robustness.

Bibliography

[1] M. Alioto, "Ultra-low power vlsi circuit design demystified and explained: A tutorial," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 59, no. 1, pp. 3–29, 2012.

[2] M. Seok, D. Blaauw, and D. Sylvester, "Robust clock network design methodology for ultra-low voltage operations," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 1, no. 2, pp. 120–130, 2011.

[3] X. Zhao, J. Tolbert, C. Liu, S. Mukhopadhyay, and S.-K. Lim, "Variation-aware clock network design methodology for ultra-low voltage (ulv) circuits," in *International Symposium on Low Power Electronics and Design (ISLPED) 2011*, 2011, pp. 9–14.

[4] S. G. Narendra and A. P. Chandrakasan, *Leakage in nanometer CMOS technologies*. Springer Heidelberg, 2006.

[5] Y. Tsividis and C. McAndrew, *Operation and Modeling of the MOS Transistor*. Oxford University Press New York, 1999, vol. 2.

[6] M. Alioto, "Understanding dc behavior of subthreshold cmos logic through closed-form analysis," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 7, pp. 1597–1607, 2010.

[7] A. Hössinger, "Simulation of ion implantation for ulsi technology," 2000.

[8] D. Schroder, *Advanced MOS devices*, ser. Modular series on solid state devices. Addison-Wesley Pub. Co., 1987. [Online]. Available: http://books.google.se/books?id=kSFTAAAAMAAJ

[9] B. Mohammadi, Y. Sherazi, and J. Rodrigues, "Sizing of Dual-VT Gates for Sub-VT circuits," 2012.

[10] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolic, *Digital integrated circuits*. Prentice hall Englewood Cliffs, 2002, vol. 2.

[11] M. Prakash, "Library characterization and static timing analysis of asynchornous circuits," 2007.

[12] P. Meinerzhagen, O. Andersson, Y. Sherazi, A. Burg, and J. Rodrigues, "Synthesis Strategies for Sub-VT Systems," 2011. [Online]. Available: http://dx.doi.org/10.1109/ECCTD.2011.6043593

Appendix A

Appendix 1

A.1 SPICE Simulation for Evenly Distributed Sinks

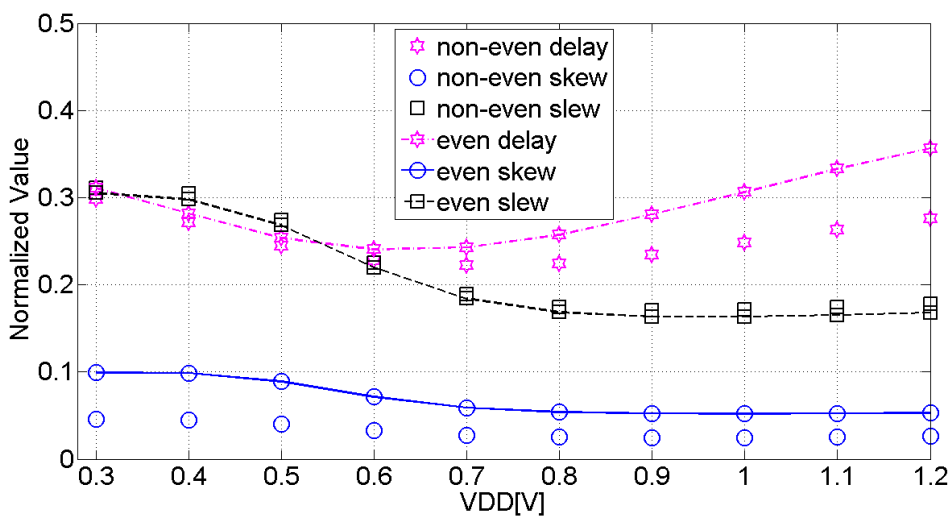


Fig. A.1: SPICE simualtion comparison for clock tree performance between evenly distributed design and non-evenly distributed