

Full Chip Timing Analysis: Scope for efciency Improvement

Project Report

Submitted in partial fulfillment of the requirements
for the degree of

Master of Technology
In Electronics & Communication Engineering
(VLSI Design)

By

Hemal Chauhan
18MECV06



Electronics & Communication Engineering Department
Institute of Technology
Nirma University
Ahmedabad - 382 481
December, 2018

Full Chip Timing Analysis: Scope for efficiency Improvement

Project Report

Submitted in partial fulfillment of the requirements
for the degree of

Master of Technology
In Electronics & Communication Engineering
(VLSI Design)

By

Hemal Chauhan
18MECV06

Internal Guide:

Dr. N.M. Devashrayee
Professor,
Institute of Technology
Nirma University

External Guide:

Rahul Kalambe
Digital Design Engineer,
Intel Technology India Pvt Ltd.



Electronics & Communication Engineering Department
Institute of Technology
Nirma University
Ahmedabad - 382 481
May, 2020

Declaration

This is to certify that

1. The thesis comprises my original work towards the degree of Master of Technology in VLSI Design at Nirma University and has not been submitted elsewhere for a degree.
2. Due acknowledgment has been made in the text to all other material used.

Hemal Chauhan
18MECV06



Certificate

This is to certify that the project entitled “**Full Chip Timing Analysis: Scope for efficiency Improvement**” submitted by **Hemal Chauhan (18MECV06)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in VLSI Design, Nirma University, Ahmedabad. The record of work carried out by her under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this minor project, to the best knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. Usha Mehta
PG Coordinator - VLSI Design

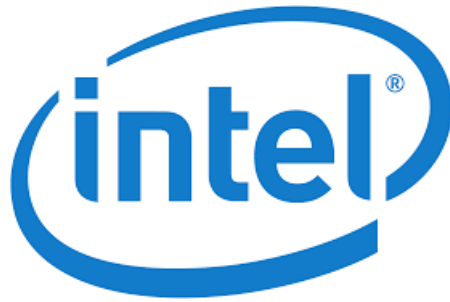
Dr. N. M. Devashrayee
Internal Guide

Dr Dhaval Pujara
Head, EC Dept.

Director
Institute of Technology

Date :

Place : Ahmedabad



Certificate

This is to certify that the project entitled “**Full Chip Timing Analysis: Scope for efficiency Improvement**” submitted by **Hemal Chauhan (18MECV06)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in VLSI Design, Nirma University, Ahmedabad. The record of work carried out by her under our supervision and guidance at **Intel Technology India Pvt. Ltd.** In my opinion, the submitted work has reached a level required for being accepted for examination.

External Guide:
Rahul Kalambe
Digital Design Engineer
Intel Technology India
Bangalore
Date :

Place : Bangalore

Acknowledgment

The development and evolution of this project has been a never ending source of both challenges and joy to me. The satisfaction that accompanies the successful completion of my project would be incomplete without the mention of the people and organization that made it possible, whose support rewarded my efforts with success.

My sincere thanks goes to Rahul Kalambe (Mentor) for leading me working on this exciting project.

Besides my team, I would like to thank the rest of my thesis committee: Dr. Usha Mehta and Dr. N.M. Devashrayee, for their encouragement, insightful comments, and hard questions during my course of project.

Finally, I must express my very profound gratitude to my parents for providing me with un-failing support and continuous encouragement throughout my years of study, researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Hemal Chauhan
18MECV06

Abstract

Leading high performance design extensively require clock-tree mesh (CT-MESH) for balanced clock distribution across SoCs. But STA tool PrimeTime can't model multi-driven clock mesh network in timing models. The mesh delays are then back annotated into STA run along with guard-bands to account for variation and aging. This adds overhead in terms of execution, increases turnaround time and adds signoff risks since, the designer needs to ensure correctness and accuracy of spine delays. CT-Mesh are a well-known clock distribution architecture meeting design requirements of distributing critical global clock signals on a chip. Clock network can have variations due to non-uniform switching activity in the design, intra-die process variations, asymmetric placement of circuit elements and manufacturing defects on atomic level. The mesh present in CT-Mesh averages out these undesirable variations between any two signal nodes contiguously distributed over the die. But its utilization is affected due to difficulty in analyzing them with sufficient accuracy.

This thesis describes an approach using PrimeTime to analyze clock mesh network accurately using SPICE simulation.

List of Figures

1.1.1 ASIC Flow	2
1.1.2 Static Timing Analysis	3
2.3.1 Clock Tree Mesh	6
2.3.2 Clock Mesh Analysis and Circuit Reduction	8
2.3.3 Clock Tree Mesh	9
2.3.4 Clock Mesh Analysis and Circuit Reduction	10
2.4.1 Clock Tree Mesh	11
2.4.2 Clock Mesh Analysis and Circuit Reduction	13
2.4.3 clock reconvergence pessimism removal from mesh	13
3.1.1 Flow chart for PT spice simulation	15
4.1.1 Hyperscale Model	19

List of Abbreviation

EDA	- Electronic Design Automation
RTL	- Register-Transfer Level
APR	- Automatic Place and Route
PT	- PrimeTime
DUA	- Design Under Analysis
STA	- Static Timing Analysis
HDL	- Hardware Description Language
SPICE	- Simulation Program with Integrated Circuit Emphasis

Chapter 1

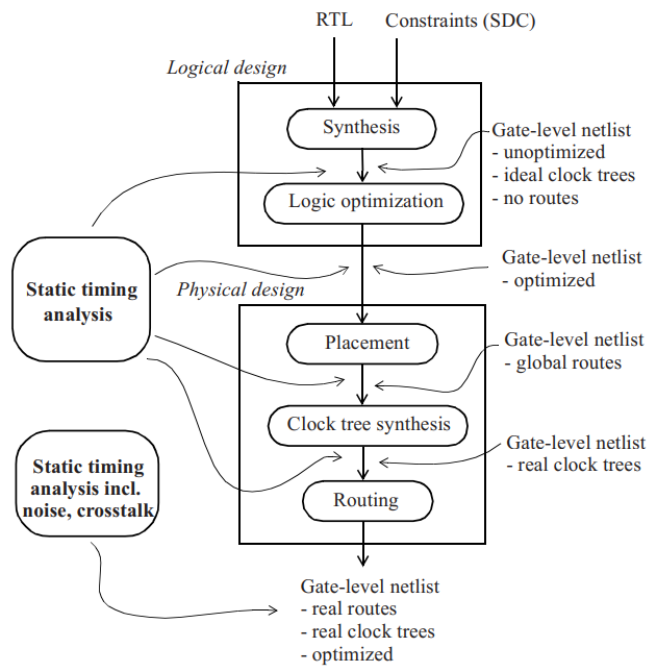
Introduction

1.1 ASIC Introduction

The electronics industry has achieved a phenomenal growth over the last few decades, mainly due to the rapid advances in integration technologies and large-scale systems design. The use of integrated circuits in high-performance computing, telecommunications, and consumer electronics has been growing at a very fast pace. ASIC (Application Specific Integrated Circuit) is a non-standard integrated circuit that is designed for a specific use or application. Generally, an ASIC design will be undertaken for a product that will have a large production run. It is a digital or mixed-signal circuit designed to meet specifications set by a particular project. Today, ASIC design flow is a mature process with many individual steps. ASIC design flow process is the backbone of every ASIC design project. In a CMOS digital design flow, the static timing analysis can be performed at many different stages of the implementation. STA is rarely done at the RTL level as, at this point, it is more important to verify the functionality of the design as opposed to timing. Also not all timing information is available since the descriptions of the blocks are at the behavioral level. Once a design at the RTL level has been synthesized to the gate level, the STA is used to verify the timing of the design. STA can also be run prior to performing logic optimization - the goal is to identify the worst or critical timing paths. STA can be rerun after logic optimization to see whether there are failing paths still remaining that need to be optimized, or to identify the critical paths. At the start of the physical design, clock trees are considered as ideal, that is, they have zero delay. Once the physical design starts and after clock trees are built, STA can be performed to check the timing again. In fact, during physical design, STA can be performed at each and every step to identify the worst paths.

The ASIC flow shown in the Figure 1.1.1 is divided into two parts known as Front end and Back end.

VLSI Front-End considers all the logical designing and verification part, It starts with architectural design and ends with synthesis. In RTL coding, with help of high level HDL coding (VHDL or verilog) the architecture of the chip, a basic skeleton of circuit will be designed as per specifications. Verification is the process of verifying the functional characteristics of the design by generating different input stimulus and checking for correct behavior of the design



flow.PNG

Figure 1.1.1: ASIC Flow

implementation. Synthesis is the process of transforming your HDL design into a gate-level netlist, given all the specified constraints and optimization settings. Logic synthesis is the process of translating and mapping RTL code written in HDL into technology specific gate level representation.

VLSI Back-End deals with further manufacturing and fabrication process. The back-end process is responsible for the physical implementation of a circuit. It transforms the RTL circuit description into a physical design (silicon), composed by gates and its interconnections. In some cases synthesis will also come under Physical Design. So, the main phases of the back-end process are Synthesis and Place&Route. Back end process is also known as Physical Design, and this thesis will throw more light on back end while going further. Static Timing Analysis is one of the many techniques available to verify the timing of a digital design. An alternate approach used to verify the timing is the timing simulation which can verify the functionality as well as the timing of the design. The term timing analysis is used to refer to either of these two methods - static timing analysis, or the timing simulation. Thus, timing analysis simply refers to the analysis of the design for timing issues. Given a design along with a set of input clock definitions and the definition of the external environment of the design, the purpose of static timing analysis is to validate if the design can operate at the rated speed. That is, the design can operate safely at the specified frequency of the clocks without any timing violations.

PrimeTime extracts the entire clock network starting from the clock source, including the mXn clock mesh and creates a spice deck for simulation. Next, Primetime invokes SPICE simulation (H-Spice engine) to analyze the clock network and obtain delays for all cells and nets in the clock network. Once the SPICE simulation completes, PrimeTime performs a model reduction

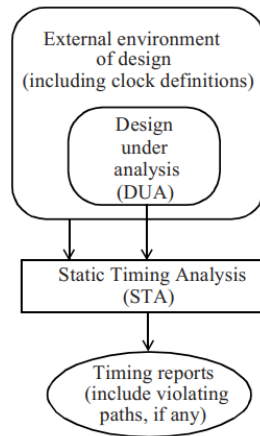


Figure 1.1.2: Static Timing Analysis

to retain only one of the parallel drivers on which it annotates the equivalent delay of the m parallel drivers. The final delay numbers come from the single remaining driver output to input pin of each of the n receivers. The purpose of circuit reduction is to avoid the large number of combinations of drivers and loads in a full-mesh analysis, while maintaining accurate driver-to-load timing results. If the mesh has n drivers and m loads, there are $n \times m$ timing arcs between drivers and loads in the mesh. However, by reducing the mesh circuit to a single driver, the number of driver-to-load timing arcs is reduced to just m .

1.2 Introduction to Clock Mesh Analysis

In high speed designs, accurate analysis of the global clock network is a critical component of static timing signoff methodology. Typically, client and server designs use clock spine implementation for global clock distribution. A clock spine consists of clock inverters which are laid out in stages where each stage of inverters drives a larger number of inverters. Usually, the last stage of clock inverters is shorted at the output. The clock spine can directly drive global driver cells or Drop-Off Points (DOPs) in a partition or have additional repeaters before reach

1.3 Motivation

- The main motivation of this project is to understand and analyze clock mesh network delay using PrimeTime and SPICE simulation.

1.4 Objective

- The core objective of this project is implementation of PTSim flow to analyze clock mesh network. It aims to improve accuracy, runtime.

- The project also aims to **improve the Full Chip Timing analysis based on new enhancements** and identify where and how manual flow can be automated.

Chapter 2

Literature Survey

2.1 Introduction to Static Timing Analysis

PrimeTime extracts the entire clock network starting from the clock source, including the mXn clock mesh and creates a spice deck for simulation. Next, Primetime invokes SPICE simulation (H-Spice engine) to analyze the clock network and obtain delays for all cells and nets in the clock network. Once the SPICE simulation completes, PrimeTime performs a model reduction to retain only one of the parallel drivers on which it annotates the equivalent delay of the m parallel drivers. The final delay numbers come from the single remaining driver output to input pin of each of the n receivers. The purpose of circuit reduction is to avoid the large number of combinations of drivers and loads in a full-mesh analysis, while maintaining accurate driver-to-load timing results. If the mesh has n drivers and m loads, there are n x m timing arcs between drivers and loads in the mesh. However, by reducing the mesh circuit to a single driver, the number of driver-to-load timing arcs is reduced to just m.

2.2 Introduction to Clock Mesh Analysis

In high speed designs, accurate analysis of the global clock network is a critical component of static timing signoff methodology. Typically, client and server designs use clock spine implementation for global clock distribution. A clock spine consists of clock inverters which are laid out in stages where each stage of inverters drives a larger number of inverters. Usually, the last stage of clock inverters is shorted at the output. The clock spine can directly drive global driver cells or Drop-Off Points (DOPs) in a partition or have additional repeaters before reach

2.3 Limitations of Static Timing Analysis

While the timing and noise analysis do an excellent job of analyzing a design for timing issues under all possible situations, the state-of-the-art still does not allow STA to replace simulation completely. This is because there are some aspects of timing verification that cannot yet be completely captured and verified in STA.

Some of the limitations of STA are: In high speed designs, accurate analysis of the global clock

network is a critical component of static timing signoff methodology. Typically, client and server designs use clock spine implementation for global clock distribution. A clock spine consists of clock inverters which are laid out in stages where each stage of inverters drives a larger number of inverters. Usually, the last stage of clock inverters is shorted at the output. The clock spine can directly drive global driver cells or Drop-Off Points (DOPs) in a partition or have additional repeaters before reaching a DOP. Local Clock Tree Synthesis (CTS) in a partition starts from the DOP cell. While it is quite acceptable to analyze other types of clock distribution architectures using any static timing analysis tool. The PrimeTime tool fails when it comes to static timing analysis of clock mesh architectures. The primary reason being its inability to model the delay between mesh drivers and mesh receivers through the mesh net (Figure 2.4.1). This is an inherent deficiency in PrimeTime tool as in such scenarios it fails to understand how waveform propagation will take place through clock mesh net with multiple drivers and how the transition profile will be across all nodes. Hence designers have implemented SPICE simulation based mesh approach.

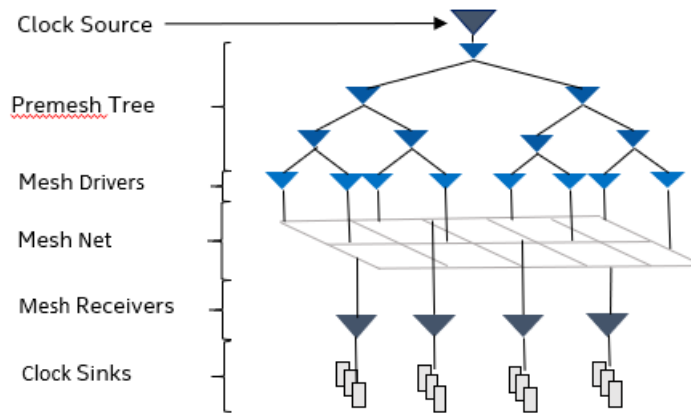


Figure 2.3.1: Clock Tree Mesh

SPICE based independent simulations

The fall back to limitation of STA based approach is to run SPICE simulations to model such multi-driver clock mesh architectures. the key advantage here is the ability of SPICE to model multiple driver scenarios, hence it can estimate delay and transition profile across clock mesh net with good accuracy. The current design practice is to extract the design up to transistor level and run SPICE simulation across various scenarios. While this approach has become the mainstream way of doing mesh analysis, they do come with their own limitations.

- i. **The accuracy of simulation** depends on getting right transition profile at inputs, which is not available to start with. Hence designers have to assume approximate value or range of values which limits simulation accuracy or add to overall run time.

ii. For independent simulations, netlist extracted can be different from what PrimeTime consumes. In many cases, the netlist can be flat. Hence it can result in design correlation issue at a later stage.

iii. Runtime: Running a single iteration of simulation may itself take hours depending on size of design. Hence for complex and bigger designs its not feasible to run simulation for smaller designs ECOs. This limitation adds to analysis time and overall time required for design analysis and optimization.

iv. Manual approach: While this is fairly better approach as compared to STA based analysis, this is also manual approach where after extraction, pruning, simulation and measurement data generation, all data needs to be manually analyzed at transistor level and back annotated on to PrimeTime manually in standard cell level as constraints. This is not only time taking but also prone to human error.

v. Number of handoffs: In a complex design environment, various teams/designers would be working on different aspects of designs. Hence a single analysis like this could require multiple handoffs, hence adding to overall design analysis time and affecting the signoff process.

To solve the above identified problems, PTSim is used. PTSim relies on a quick spice simulation of the global clock network natively through the PrimeTime SimLink interface. Transistorlevel Spice simulation is followed by a fine-grained user-configurable stage by stage delay and transition back annotation from the global clock root to the DOPs.

Analysis of multi-driven nets with PrimeTime

PrimeTime extracts the entire clock network starting from the clock source, including the mXn clock mesh and creates a spice deck for simulation. Next, Primetime invokes SPICE simulation (H-Spice engine) to analyze the clock network and obtain delays for all cells and nets in the clock network. Once the SPICE simulation completes, PrimeTime performs a model reduction to retain only one of the parallel drivers on which it annotates the equivalent delay of the m parallel drivers. The final delay numbers come from the single remaining driver output to input pin of each of the n receivers. The purpose of circuit reduction is to avoid the large number of combinations of drivers and loads in a full-mesh analysis, while maintaining accurate driver-to-load timing results. If the mesh has n drivers and m loads, there are n x m timing arcs between drivers and loads in the mesh. However, by reducing the mesh circuit to a single driver, the number of driver-to-load timing arcs is reduced to just m.

In high speed designs, accurate analysis of the global clock network is a critical component of static timing signoff methodology. Typically, client and server designs use clock spine implementation for global clock distribution. A clock spine consists of clock inverters which are laid out in stages where each stage of inverters drives a larger number of inverters. Usually, the last stage of clock inverters is shorted at the output. The clock spine can directly drive global driver cells or Drop-Off Points (DOPs) in a partition or have additional repeaters before reaching a DOP. Local Clock Tree Synthesis (CTS) in a partition starts from the DOP cell. While it is quite acceptable to analyze other types of clock distribution architectures using any atatic timing analysis tool. The PrimeTime tool fails when it come to static timing analysis of clock

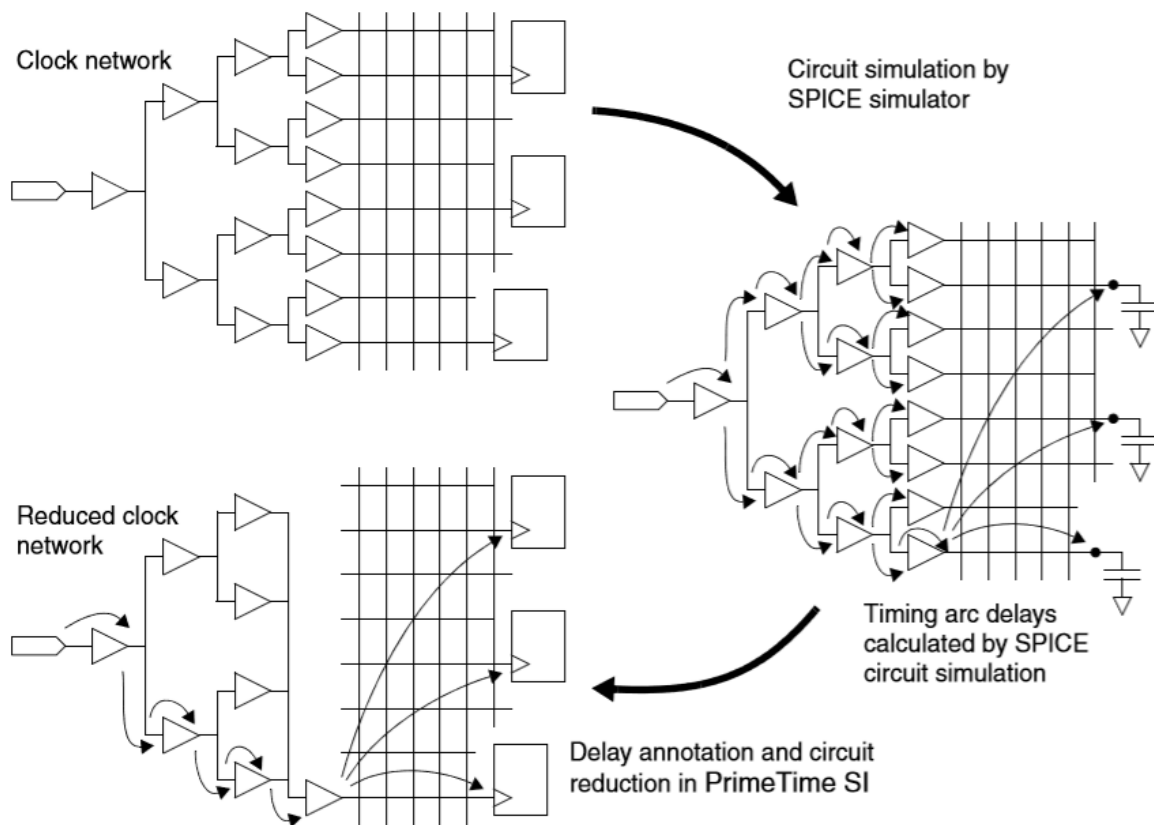


Figure 2.3.2: Clock Mesh Analysis and Circuit Reduction

mesh architectures. The primary reason being its inability to model the delay between mesh drivers and mesh receivers through the mesh net (Figure 2.4.1). This is an inherent deficiency in PrimeTime tool as in such scenarios it fails to understand how waveform propagation will take place through clock mesh net with multiple drivers and how the transition profile will be across all nodes. Hence designers have implemented SPICE simulation based mesh approach.

SPICE based independent simulations

The fall back to limitation of STA based approach is to run SPICE simulations to model such multi-driver clock mesh architectures. The key advantage here is the ability of SPICE to model multiple driver scenarios, hence it can estimate delay and transition profile across clock mesh net with good accuracy. The current design practice is to extract the design up to transistor level and run SPICE simulation across various scenarios. While this approach has become the mainstream way of doing mesh analysis, they do come with their own limitations.

- i. **The accuracy of simulation** depends on getting right transition profile at inputs, which

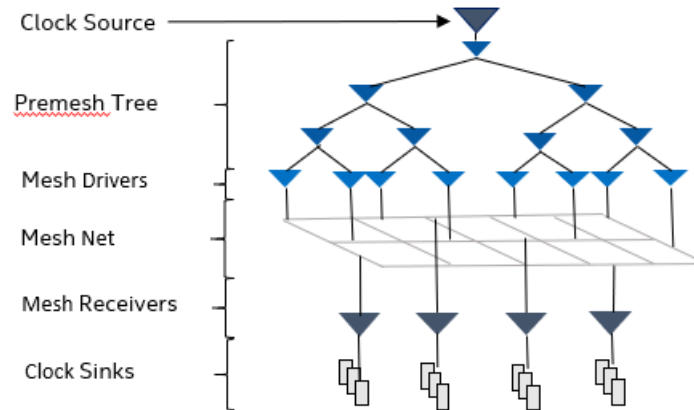


Figure 2.3.3: Clock Tree Mesh

is not available to start with. Hence designers have to assume approximate value or range of values which limits simulation accuracy or add to overall run time.

ii. For independent simulations, netlist extracted can be different from what PrimeTime consumes. In many cases, the netlist can be flat. Hence it can result in design correlation issue at a later stage.

iii. Runtime: Running a single iteration of simulation may itself take hours depending on size of design. Hence for complex and bigger designs its not feasible to run simulation for smaller designs ECOs. This limitation adds to analysis time and overall time required for design analysis and optimization.

iv. Manual approach: While this is fairly better approach as compared to STA based analysis, this is also manual approach where after extraction, pruning, simulation and measurement data generation, all data needs to be manually analyzed at transistor level and back annotated on to PrimeTime manually in standard cell level as constraints. This is not only time taking but also prone to human error.

v. Number of handoffs: In a complex design environment, various teams/designers would be working on different aspects of designs. Hence a single analysis like this could require multiple handoffs, hence adding to overall design analysis time and affecting the signoff process.

To solve the above identified problems, PTSim is used. PTSim relies on a quick spice simulation of the global clock network natively through the PrimeTime SimLink interface. Transistorlevel Spice simulation is followed by a fine-grained user-configurable stage by stage delay and transition back annotation from the global clock root to the DOPs.

Analysis of multi-driven nets with PrimeTime

PrimeTime extracts the entire clock network starting from the clock source, including the mXn clock mesh and creates a spice deck for simulation. Next, Primetime invokes SPICE simulation (H-Spice engine) to analyze the clock network and obtain delays for all cells and

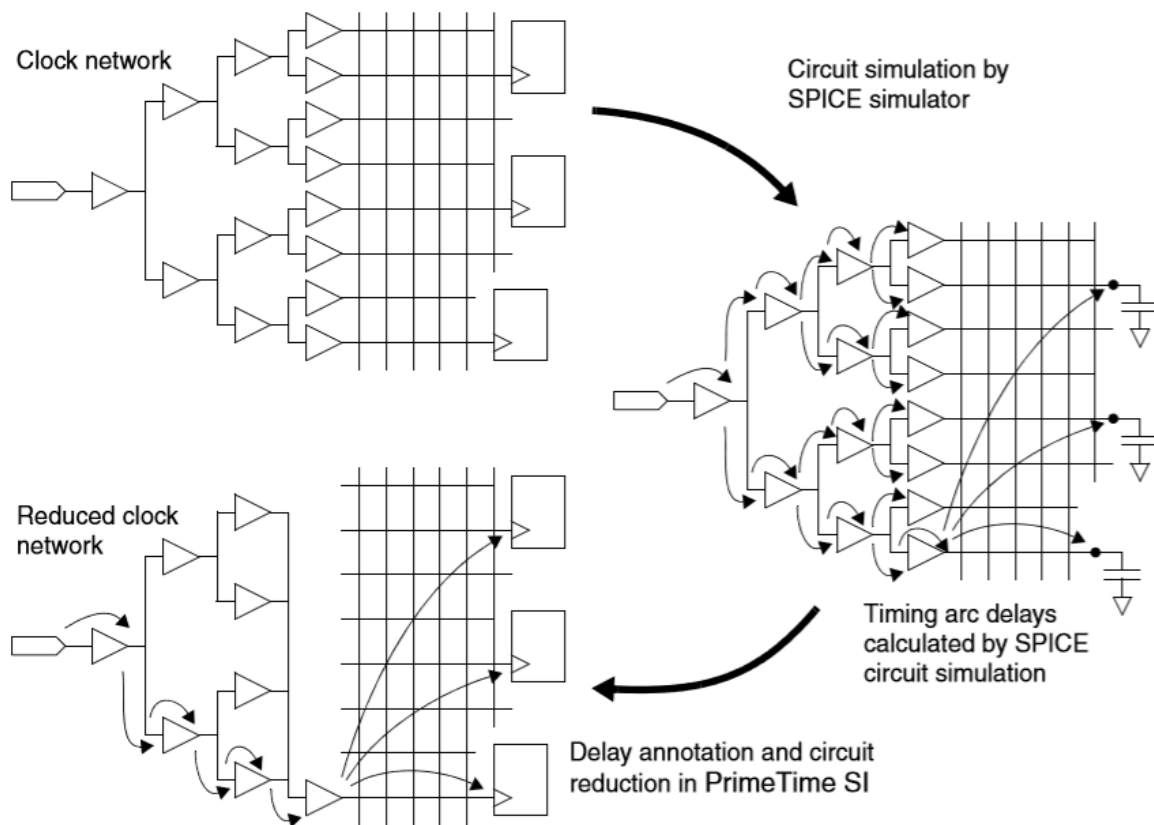


Figure 2.3.4: Clock Mesh Analysis and Circuit Reduction

nets in the clock network. Once the SPICE simulation completes, PrimeTime performs a model reduction to retain only one of the parallel drivers on which it annotates the equivalent delay of the m parallel drivers. The final delay numbers come from the single remaining driver output to input pin of each of the n receivers. The purpose of circuit reduction is to avoid the large number of combinations of drivers and loads in a full-mesh analysis, while maintaining accurate driver-to-load timing results. If the mesh has n drivers and m loads, there are $n \times m$ timing arcs between drivers and loads in the mesh. However, by reducing the mesh circuit to a single driver, the number of driver-to-load timing arcs is reduced to just m .

2.4 Introduction to Clock Mesh Analysis

In high speed designs, accurate analysis of the global clock network is a critical component of static timing signoff methodology. Typically, client and server designs use clock spine implementation for global clock distribution. A clock spine consists of clock inverters which are laid out in stages where each stage of inverters drives a larger number of inverters. Usually, the last stage of clock inverters is shorted at the output. The clock spine can directly drive global driver cells or Drop-Off Points (DOPs) in a partition or have additional repeaters before reach-

ing a DOP. Local Clock Tree Synthesis (CTS) in a partition starts from the DOP cell. While it is quite acceptable to analyze other types of clock distribution architectures using any static timing analysis tool. The PrimeTime tool fails when it comes to static timing analysis of clock mesh architectures. The primary reason being its inability to model the delay between mesh drivers and mesh receivers through the mesh net (Figure 2.4.1). This is an inherent deficiency in PrimeTime tool as in such scenarios it fails to understand how waveform propagation will take place through clock mesh net with multiple drivers and how the transition profile will be across all nodes. Hence designers have implemented SPICE simulation based mesh approach.

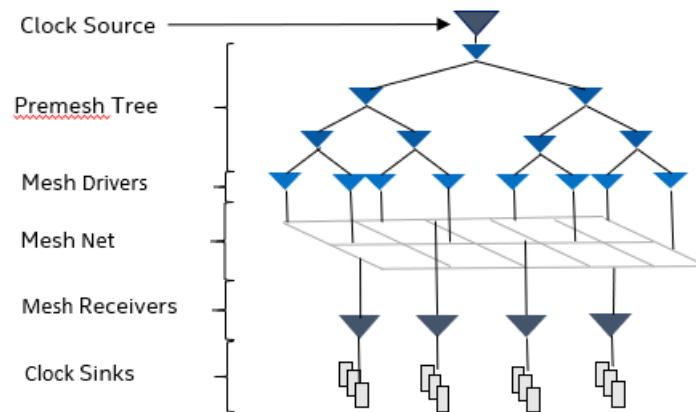


Figure 2.4.1: Clock Tree Mesh

SPICE based independent simulations

The fall back to limitation of STA based approach is to run SPICE simulations to model such multi-driver clock mesh architectures. The key advantage here is the ability of SPICE to model multiple driver scenarios, hence it can estimate delay and transition profile across clock mesh net with good accuracy. The current design practice is to extract the design up to transistor level and run SPICE simulation across various scenarios. While this approach has become the mainstream way of doing mesh analysis, they do come with their own limitations.

- i. The accuracy of simulation** depends on getting right transition profile at inputs, which is not available to start with. Hence designers have to assume approximate value or range of values which limits simulation accuracy or add to overall run time.
- ii. For independent simulations**, netlist extracted can be different from what PrimeTime consumes. In many cases, the netlist can be flat. Hence it can result in design correlation issue at a later stage.
- iii. Runtime:** Running a single iteration of simulation may itself take hours depending on size of design. Hence for complex and bigger designs its not feasible to run simulation for smaller designs ECOs. This limitation adds to analysis time and overall time required

for design analysis and optimization.

iv. Manual approach: While this is a fairly better approach as compared to STA based analysis, this is also a manual approach where after extraction, pruning, simulation and measurement data generation, all data needs to be manually analyzed at transistor level and back annotated on to PrimeTime manually in standard cell level as constraints. This is not only time taking but also prone to human error.

v. Number of handoffs: In a complex design environment, various teams/designers would be working on different aspects of designs. Hence a single analysis like this could require multiple handoffs, hence adding to overall design analysis time and affecting the signoff process.

To solve the above identified problems, PTSim is used. PTSim relies on a quick spice simulation of the global clock network natively through the PrimeTime SimLink interface. Transistorlevel Spice simulation is followed by a fine-grained user-configurable stage by stage delay and transition back annotation from the global clock root to the DOPs.

Analysis of multi-driven nets with PrimeTime

PrimeTime extracts the entire clock network starting from the clock source, including the $m \times n$ clock mesh and creates a spice deck for simulation. Next, Primetime invokes SPICE simulation (H-Spice engine) to analyze the clock network and obtain delays for all cells and nets in the clock network. Once the SPICE simulation completes, PrimeTime performs a model reduction to retain only one of the parallel drivers on which it annotates the equivalent delay of the m parallel drivers. The final delay numbers come from the single remaining driver output to input pin of each of the n receivers. The purpose of circuit reduction is to avoid the large number of combinations of drivers and loads in a full-mesh analysis, while maintaining accurate driver-to-load timing results. If the mesh has n drivers and m loads, there are $n \times m$ timing arcs between drivers and loads in the mesh. However, by reducing the mesh circuit to a single driver, the number of driver-to-load timing arcs is reduced to just m . PrimeTime analysis of multi-driven nets is illustrated in Figure 2.4.2

The circuit has various delay values from the single mesh driver to many loads on the mesh. For clock reconvergence pessimism removal (CRPR), the shortest delay from the mesh driver to the nearest mesh load represents the amount of delay that is shared by all the timing arcs from the driver to the loads. To account for the common path shared by the n receivers, the shortest delay from the single retained driver to any of the n receivers is added to the driver delay and subtracted from all other receiver delays since that is common mode for all the receivers as shown in Figure 2.4.3. To gain the most benefit of CRPR, this shared delay should be accounted for before the CRPR common point. PrimeTime SI uses the output of the single retained mesh driver as the CRPR common point.

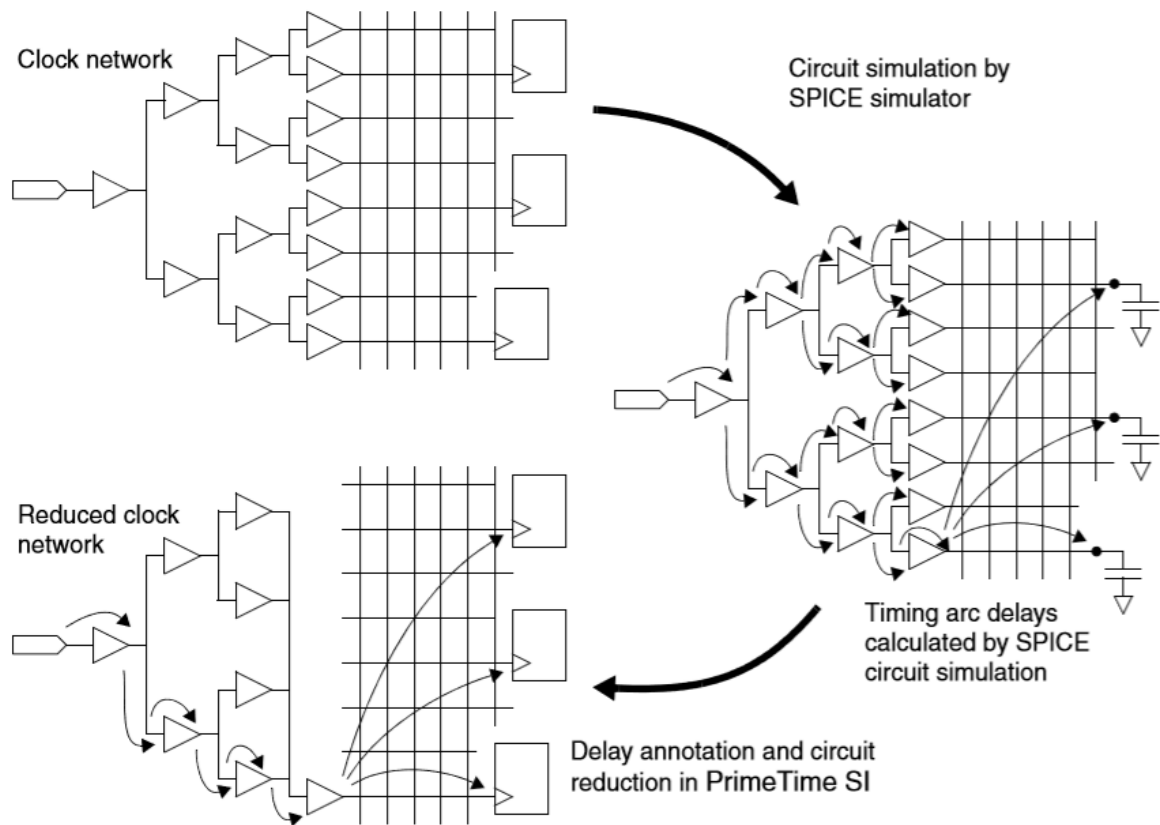


Figure 2.4.2: Clock Mesh Analysis and Circuit Reduction

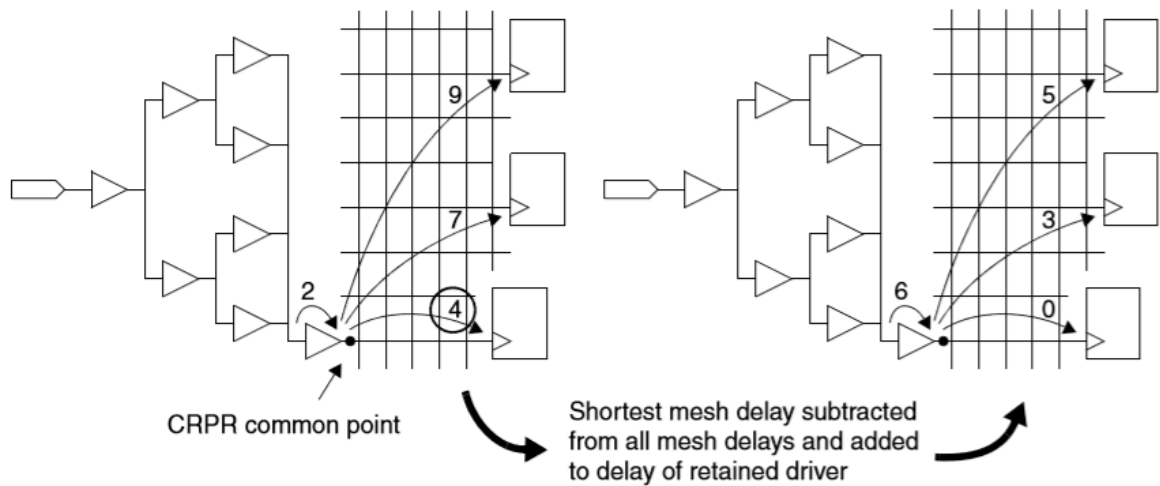


Figure 2.4.3: clock reconvergence pessimism removal from mesh

Chapter 3

Clock spine analysis using SPICE within PrimeTime

3.1 PrimeTime spice simulation flow

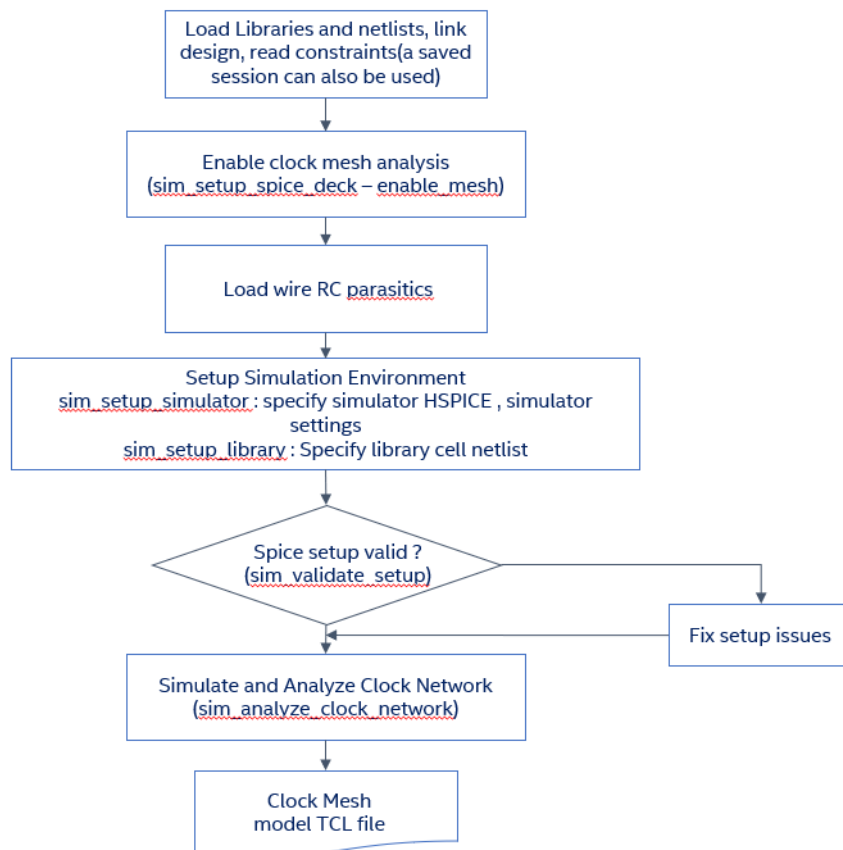


Figure 3.1.1: Flow chart for PT spice simulation

Steps for performing clock spine analysis with Primetime invoked SPICE simulator are shown in Figure 3.1.1. The simulation is run on a PrimeTime timing database that includes the clock spine network. Before analysis, it is important to validate SPICE setup through SPICE simulation of simple library cells such as buffers or inverters. Once spice setup validation is successful, designer can proceed with the clock network simulation by invoking the `sim_analyze_clock_network` command within PrimeTime. Once clock network simulation is done, a TCL file containing transitions and delays for all stages of the clock spine network is generated. The output TCL file can be reused for future runs at the same timing corner if the clock spine has not changed. The analysis is done for all timing corners.

Clock Network Simulation Commands:

i. sim_setup_simulator:

The `sim_setup_simulator` command sets up the simulation environment by specifying the simulator location, simulator type, and working directory.

Here is an example of the `sim_setup_simulator` command.

```
sim_setup_simulator -simulator /usr/bin/hspice -simulator_type hspice -work_dir ./tmp_dir
```

ii. sim_setup_library:

The `sim_setup_library` command performs library setup tasks such as mapping the transistor models to the gate-level models.

Here is an example of the `sim_setup_library` command.

```
sim_setup_library -library $lib_name -sub_circuit /u/xtalk/si_lib_gen/unit/gem/hspice -header /u/xtalk/si_lib_gen/unit/gem/model_hspice
```

iii. sim_setup_spice_deck:

The `sim_setup_spice_deck` command specifies the setup options to write out the SPICE deck.

Here is an example of using this command to enable clock model generation flow.

```
sim_setup_spice_deck -enable_clock_mesh
```

iv. sim_validate_setup:

Setting the SPICE simulation environment incorrectly can cause problems in the clock mesh analysis flow. To help ensure that the setup is correct, you can use the `sim_validate_setup` command. This command invokes the simulator specified by the `sim_setup_simulator` command on the timing arcs of a given library cell and verifies that the basic characterization setting in the library matches the SPICE setup specified by the `sim_setup_simulator` and `sim_setup_library` commands. The `sim_validate_setup` command supports only combinational cells. The `sim_validate_setup` command checks for the presence of a SPICE executable, SPICE netlists, SPICE model files, and applicable licenses. It checks for necessary SPICE options and data such as I/O rail settings and correct unit sizes. It also checks the model license and compatibility of the SPICE version with the model. If no errors are found, the delay and slew values computed by SPICE and PrimeTime SI are displayed. Before you use the `sim_validate_setup` command, the library must be fully validated for accuracy. This command assumes that the library is correct and accurate, and any mismatch between PrimeTime SI and SPICE is caused by an incorrect simu-

lator setting, not library errors. You can only use the command when no designs are loaded into PrimeTime SI.

Here is an example of the `sim_validate_setup` command.

```
sim_validate_setup -from A -to Y -lib_cell [get_lib_cells $lib_name/IV170BQ]
```

v. `sim_analyze_clock_network`:

The `sim_analyze_clock_network` command extracts the clock network from the specified clock root and invokes the simulator specified by the `sim_setup_simulator` command. It creates a SPICE deck, links to the simulation environment, runs the simulation, and back-annotates the results onto the design in PrimeTime SI.

Here is an example of the `sim_analyze_clock_network` command.

```
sim_analyze_clock_network -from [get_ports CLK1] -output ./clock_mesh_model.tcl
```


Chapter 4

Hyperscale

4.1 Hyperscale Block Model

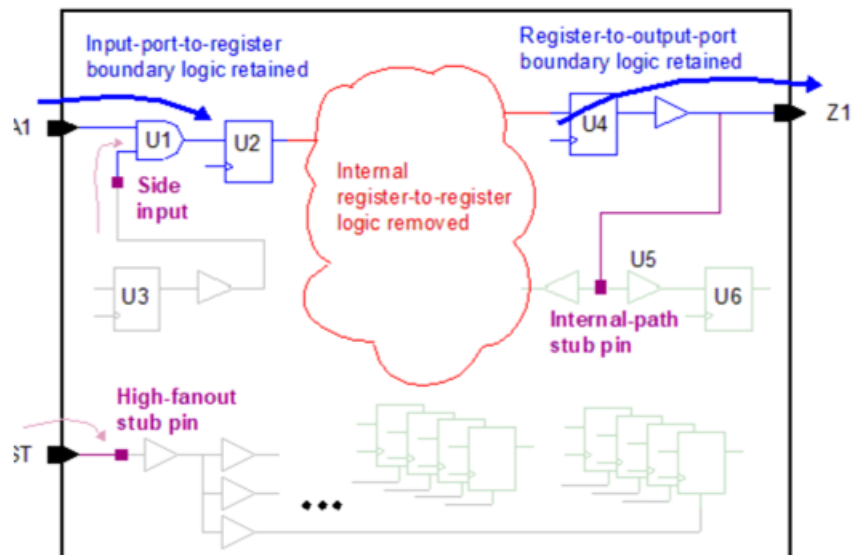


Figure 4.1.1: Hyperscale Model

A Hyperscale block model includes the following types of logic:

- Input-to-register boundary logic**
- Register-to-output boundary logic**
- High-fanout pins to represent the timing of removed high-fanout logic**
- Side Input and stub pins**

The Hyperscale block model includes all the logic necessary for accurate timing analysis at the top level. As a result, the model is as fast and compact as possible for analysis with flat-like accuracy. In hierarchical analysis, the tool analyzes the timing of lower-level blocks separately from the top level. In a bottom-up hierarchical flow, the top-level analysis uses a HyperScale

model to represent each lower-level HyperScale block. Each model includes the interface logic of the block boundary but excludes the internal register-to-register logic of the block.

Chapter 5

Results

Advantages of PTSim are mentioned below in comparison to regular spice approach:

Accuracy – Delays annotated by PTSim has better correlation with Silicon than regular spice approach since no additional collaterals are required and no guardband is added.

Skew - Skew observed across global drivers on Silicon has better correlation with PTSim reported values compared to Spice simulations. Values reported by spice are also penalized by additional guardbands for skew.

Runtime – PTSim is 2.5x faster than regular spice approach.

Chapter 6

Conclusion and Future Work

It can be concluded that PTSim provides an accurate simulation of clock spines. Since PTSim is native to PrimeTime and SPICE simulation is invoked within PrimeTime, enabling PTSim was very straightforward. Through PTSim, we were able to analyze timing paths through clock spine accurately since variation, aging and crosstalk were natively modeled by PrimeTime.

We are pursuing these for future projects.

- In early stages of the project, clock spine may not be fully built and it may be required to have clocks stamped at the DOP input with additional guardband. Once the clock spine is built, clock stamping can be moved to PLL/compensator output and allowed to be propagated to DOPs. There could still be some broken connectivity or DRC issues, due to which we can see inaccurate arrival times at DOPs.
- There are limitations to `sim_analyze_clock_network`, it requires us to specify both the startpoint and endpoint of the global clock network, which can be tedious. We are working with EDA vendor to improve the behavior.

References

- [1] J. Bhasker, Rakesh Chadha (2009). Static Timing Analysis for Nanometer Designs : A Practical Approach.
- [2] PrimeTime Clock Mesh Analysis: <https://solvnet.synopsys.com/>
- [3] G. Jung et al., "Skew variation compensating technique for clock mesh networks," APC-CAS 2008 - 2008 IEEE Asia Pacific Conference on Circuits and Systems, Macao, 2008, pp. 894-897.
- [4] VLSI Expert Website: <https://www.vlsi-expert.com/>
- [5] Signoff semiconductor Website: <https://www.signoffsemi.com/>
- [6] All of VLSI Website: <https://allofvlsi.blogspot.com/>
- [7] Documents provided by Intel Technology India Pvt Ltd.
- [8] W. Liu, C. Sitik, E. Salman, B. Taskin, S. Sundareswaran and B. Huang, "SLECTS: Slew-Driven Clock Tree Synthesis," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 4, pp. 864-874, April 2019.