RTL Power Estimation and Optimation Flow for 5G Radio Products

Divya Khanna di7151kh-s@student.lu.se
Yu Zhu yu0668zh-s@student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisor:
Pietro Andreani, LTH
Minh Do, Ericsson

Co-Supervisor:
Jonas Carlsson, Ericsson

Examiner: Erik Larsson

September 2, 2021

© 2021

Printed in Sweden
Tryckeriet i E-huset, Lund

List Of Acronyms

ASIC Application-Specific Integrated Circuit.

BE Back-End.

Calib Calibration.

CMOS Complementary Metal Oxide Semiconductor.

DUT Design Under Test.

FE Front-End.

GL Gate Level.

IC Integrated Circuit.

IO Input Output.

NMOS N-channel Metal-Oxide Semiconductor.

PE Power Estimation.

PMOS P-channel Metal-Oxide Semiconductor.

RTL Register Transfer Level.

SOC System on Chip.

VLSI VeryLarge IntegratedCircuit.

EDA Electronic Design Automation

Glossary

GL PE w. BE netlist Gate Level power estimation with back-end netlist, using the selected GL PE tool.

GL PE w. FE netlist Gate Level power estimation with front-end netlist, using the selected GL PE tool.

RTL PE w.Calibration RTL power estimation with calibration data, using the selected RTL PE tool.

RTL PE w/o Calibration RTL power estimation without using calibration data, using the selected RTL PE tool.

Abstract

Power reduction is becoming a critical design requirement for ASIC/SOC designers. Reducing both dynamic and leakage power is essential to meet power budgets for portable devices as well as to ensure that these ASICs meet their packaging and cooling costs. In addition, the power of an ASIC has a significant impact on its reliability and manufacturing yield.

Also, low-power has become a leading design criterion for 5G Radio products that demand increasingly higher performance and lower energy footprint. Traditionally, most automated power optimization tools have focused at gate-level and physical level optimizations. However, major power reductions are only possible by addressing power at the RTL and system levels. At these levels, it is possible to make the sequential modifications needed to reduce power and energy consumption via techniques like sequential clock gating, power gating, frequency scaling and other micro-architectural techniques.

With the increasing requirement of low power design, estimating power consumption must be done early in the process and waiting until the netlist is available can be too late. Designers want to get the accurate power estimate at RTL stage to shorten their design period. However, as there is no netlist available at RTL stage, the accuracy of the power estimated at RTL stage may not be acceptable.

This thesis begins with a review of several commonly used RTL PE methodologies, followed by the design of an automated RTL PE flow based on a commercially available EDA RTL power estimation tool, and finally, a sub-chip for a 5G device from Ericsson is used as the DUT to investigate the reasons for RTL PE inaccuracy and ways to improve the accuracies.

The estimated power consumption from this RTL PE flow with and without calibration is compared with GL PE with front-end netlist and back-end netlist to identify the critical reason for RTL power estimation inaccuracy, and then a guideline for improving RTL PE accuracy is listed in the thesis's result section.

Acknowledgements

We would like to take this opportunity to thank LTH (Lund University) and Ericsson for giving us this opportunity to pursue our master thesis. Moreover, we would like to thank our supervisor at Ericsson, Minh Do and our co-advisor Jonas Carlsson, for their continuous feedback, support and guidance. I would also like to thank Venkata Jagannadha for his great insight and patience to answer all our questions.

Last but not the least, I would like to thank our supervisor at the university, Pietro Andreani and our examiner, Erik Larsson for their great support and continuous feedback throughout this thesis work.

Divya Khanna, Lund 2021
Yu Zhu, Lund 2021

Popular Science Summary

Nowadays the most advanced chips has huge amount of transistors, Apple's M1 chip has about 16 billion transistors,[2]. Nvidia's RTX 3090 GPU even has 28 billion transistors,[3]. According to Moore's law, the number of transistor in integrated circuit will doubles about every two years, which means that in the future, the number of transistors in integrated circuits will get even higher.

With the increase of number of transistors in integrated circuits, Power consumption has become a big problem. On one hand, high power consumption will cause huge dissipation of electricity energy, on the other hamd, it will also increase the temperature which affects the stability of integrated circuits.

During the design process, designer need to write RTL code according to specification first, then synthesis the RTL code to get the front-end netlist and after that, P&R is done on the the front-end netlist in order to get the back-end netlist. it is a very timing consuming flow. For a big design, the designer spends several months or even several years, in order to complete the flow.

Normally designer can only get the power consumption once the front-end netlist or back-end netlist is generated, if the power consumption does not meets the requirement, designer may need to modify the RTL code and go through the whole flow again, which will make the whole design time even longer. If designers can get the power consumption at RTL stage, time can be saved.

During our thesis work, we want to look into a commercially available EDA RTL power estimation tool, which aids in the estimation of RTL power. We'll also look at how accurate the projected power is at the RTL stage compared to what's accessible at the GL PE with front-end netlist, and how we may enhance power estimation.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The demand for high-speed processing and efficient power design is increasing as electronic products such as televisions, computers, and mobile devices become more popular and successful. Furthermore, with the impending introduction of 5G and the Internet of Things (IoT), a large number of electronic devices will be able to connect and share data with one another, for example, machine to machine or machine to automobile communications,[6]. These criteria have resulted in a challenging high-level problem of balancing performance, power, and transistor density,[5]. All sectors are concerned, starting from the industries to the agriculture, telecommunication, health, etc. Human activities and technologies have a significant impact of the worldwide carbon footprint. It has been shown that cities cover 2% of Earth's surface but consume up to 78% of the world's energy. In the same time, it has been shown that developing smart and energy-efficient technologies may be an efficient solution to drastically reduce the energy cost and the environmental impact. These electronic devices are generally designed using a Very-Large-Scale Integration (VLSI) process that consists in building an integrated circuit (IC) by linking millions of transistors on a chip. All complex systems and communication devices are based on VLSI, including analog ICs such as sensors and operational amplifiers as well as digital ICs, such as microprocessors, Digital Signal Processors (DSPs), micro-controllers,[6].

In the current manufacturing system designs, the designers must consider performance and power efficiency of developed circuits, imposed by the market and applications. This consideration is becoming even more important, mainly for those critical power-constrained devices, such as battery powered mobile systems. Thus, the development of circuits for obtaining low-power dissipation and useful performance is an important research topic nowadays,[5].

Power consumption can be estimated using dedicated tools or simulations at different steps along the design flow as indicated by figure1.1. Right after design synthesis, power can be estimated using the resource number information coming from the synthesis tool and by taking into account the estimated timing. The Post-P&R power estimation takes into account the physical implementation details, including routing delays, so that timing information is more realistic. Finally, power measurements can be realized on ASICs after implementation,[6].

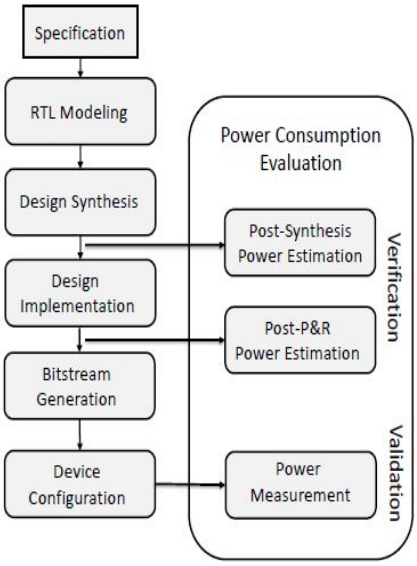


Figure 1.1: Different points of PE in digital design flow

However, if estimated power consumption is only available when front-end netlist or even back-end netlist available. If the estimated power consumption cannot meet the requirement, designer may need to modify the RTL code and do synthesis and PnR again, which consumes lot of time and money. If the power estimation can be available at RTL stage, lot of time could be saved.

This thesis will focus on an implementation of a power estimation flow at RTL stage based on the selected RTL PE tool . At the same time, as there is no netlist at RTL stage, the accuracy can't be guaranteed. The thesis also gives an guideline on the ways to improve the accuracy of the RTL power estimation flow.

1.1 Disposition

The disposition of the thesis is as follows. In the first chapter, we provide a general introduction about the importance of power estimation in digital design flow and different points where we can do power estimation in the flow. In chapter two, we provide the reader with a theoretical background of the subject. This chapter will go through ASIC design flow briefly, then we will talk about different sources of power dissipation in an IC and the power components that we look into and finally we will be talking about PE techniques. The development of our proposed implementation will be explained in chapter three. In this chapter, we will give the reader a complete understanding of what we have done in RTL PE. In chapter four, the results for this thesis works are discussed. The conclusions with possible future work can later be found in chapter five.

Chapter 2
Theory

In this chapter, the basic theory relevant for this thesis are covered. First part is about CMOS IC, in the second part we would be talking about CMOS power consumption followed by explanation about PE flow and RTL PE.

2.1 CMOS Integrated Circuit

CMOS is the most-commonly used technology of today's integrated circuit world, also it is the basis of some more advanced technology like FinFET and so on. The basic theory of CMOS is described in this part. It normally consist of NMOS and PMOS, take NMOS as an example in the Figure 2.1.

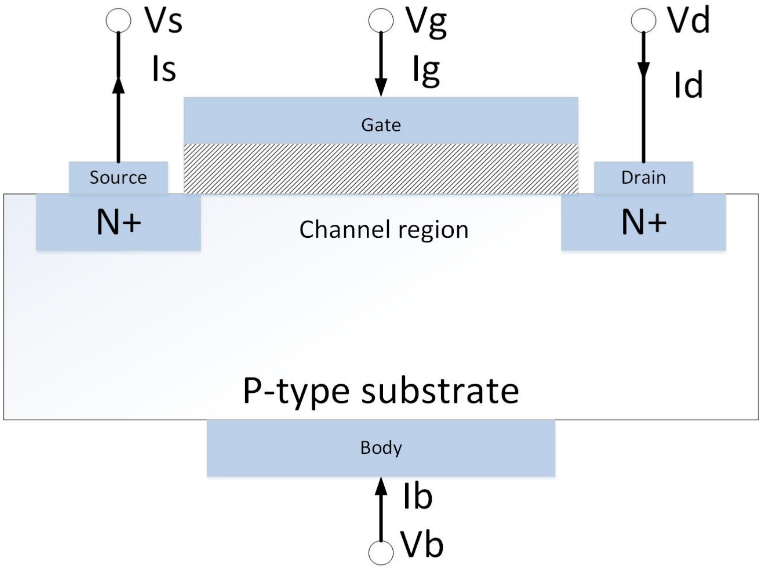


Figure 2.1: Structure of NMOS

In theory, if there is no voltage on the gate of a MOS, there should has no current between source and drain, because there are at least one PN junction is under reverse-biased. When the gate voltage is applied, and the voltage is strong enough, a N-channel will be generated (for NMOS) to connect source and drain, then electrons can flow between source and drain freely.

2.2 Sources of Power Dissipation

There are two major sources of power dissipation in CMOS integrated circuits, which are covered in the following section. Power consumption in CMOS IC can be divided in to two parts, dynamic power consumption and static power consumption.

2.2.1 Dynamic Power

Dynamic power is the power consumed by a device when the signals are changing, that is when the circuit is in active state. The sources of dynamic power are:

Switching Power: This is the power spent charging and discharging the capacitance of the output net; it is also called load power. Because such charging and discharging is the result of the logic transitions at the output of the cell, switching power increases as the logic transitions increases. Figure 2.2 shows the switching power consumption caused by the transistors, in an inverter.

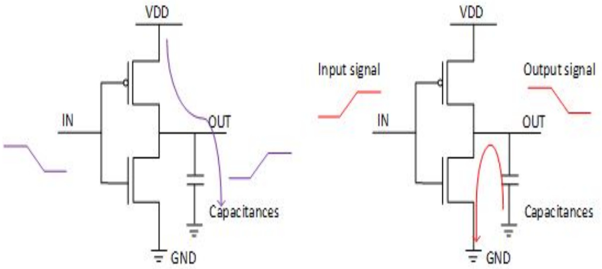


Figure 2.2: Switching activity of an Inverter

Switching power can be calculated by equation 2.1:

$$P_{Switching} = C_L \cdot V_{DD}^2 \cdot f_{clk} \cdot \alpha \tag{2.1}$$

Where C_L is the load capacitance (sum of net and gate capacitance), α is switching activity of the signal and f_{clk} is the frequency of the clock.

Internal Power: Internal power can be divided into two part, the first part is the short current power, the second part is the charge and discharge of internal capacitance in the slope of a cell.

Short circuit power

Internal power is the dynamic power dissipated within the boundary of a cell. It includes the power dissipation due to charging or discharging of capacitances internal to the cell during switching; and the power dissipation due to the momentary short circuit between the P and N transistors of a gate while both are turned on. It is also called short circuit power.

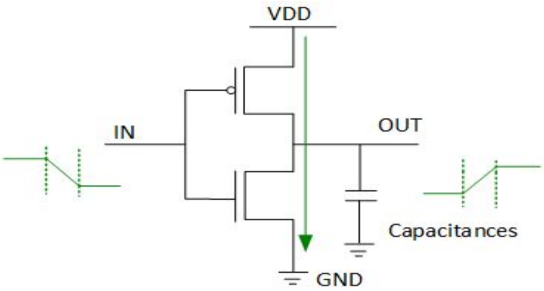


Figure 2.3: Short circuit power of an Inverter

Ideally, if the input signal needs to switch, it changes from zero to one immediately, but in real world, this situation does not exists. Input signal need time to switch, which means there exists a slope between zero and one. Looking at the Figure 2.3, we can see as the signal transitions from high to low, the P-type transistor turns on and the N-type transistor turns off. However, for a short time during signal transition, both the P and N-type transistors can be on simultaneously. During this time, current I_{sc} flows from VDD to GND, causing the dissipation of short-circuit power (P_{sc}). Short circuit power can be calculated by the following formula:

$$P_{sc} = V_{DD} \cdot I_{peak} \cdot t_{sc} \cdot f_{clk} \tag{2.2}$$

Charge and discharge of internal capacitance

Similar with switching power, there exist capacitance inside the slope of a cell, these capacitance will be charge and discharge when the cell is working, this part of power also be included in the internal power.

2.2.2 Static Power

Static power is the power consumed by a device when no signals are changing values. Since static power consumption is primarily caused by leakage, in CMOS devices, it is also known as *leakage power*. This is the power dissipated whenever the device is powered,but is in inactive state (when it is not switching). One of the primary reason is unwanted current in the transistors channels when they are switched off also, there are several sources of leakage power like intrinsic and gate leakage but they are all lumped together into a single value for modeling purposes.

The amount of leakage power dissipated by a gate can depend on process technology dimensions, gate oxide thickness and supply voltage. Whereas in Intrinsic Leakage, we include power dissipated due to sub-threshold leakage, current leakage between the diffusion layers and the substrate. The leakage power is state and voltage-dependent.

2.2.3 Power Components

Different components of power matrix that focus on PE and reduction:

- **Combinational Power :** It is the power consumed by the combinational part of the circuit and the nets driven by combinational logic. It is dominant in data path intensive designs. It is a direct impact of high data toggle and large combinational logic.
- **Sequential Power :** The power consumed by the sequential logic like latches, flip flops etc., in a design and the output nets of it.
- **Clock Power :** Power consumed by the clock tree.
- **Memory Power :** The power consumed by the memory block and the output nets connected to it.

Sequential and clock power is a direct impact of the clock in the design. If the clock is not gated sufficiently well, it will lead to high clock and sequential power. Clock gating using combinational and sequential techniques should be used to reduce the power.

2.3 Power estimation flow in ASIC design process

The journey of designing an ASIC (application specific integrated circuit) is long and involves a number of major steps – moving from a concept to specification to tape-outs. Although the end product is typically quite small (measured in milimeters and the transistors inside it are measured in nanometers), this long journey is interesting and filled with many engineering challenges,[1].

To ensure successful ASIC design, engineers must follow a proven ASIC design flow as can be seen in figure 2.4, which is based on design specifications and ASIC Design flow, with a focus on meeting the goal of right time to market,[1]. Important factor for any IC are performance, power, area and yield. With the increased demand on IoT, power consumption is becoming more critical.

Important factor for any IC are performance, power, area and yield. While ASIC flows are getting more and more matured it is good to have estimation of timing, area and power at the initial design phase so still there is some opportunity for tuning.

Some of these criteria can be checked as we go through the process, like for functionality of the design and we have verification engineers that check it's functionality for all the corner cases and then during place and route process we check

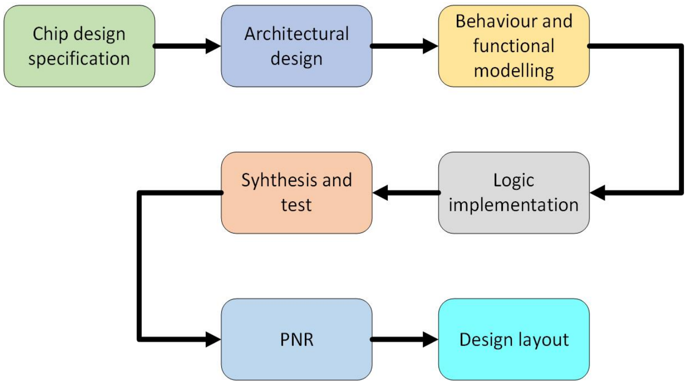


Figure 2.4: ASIC Design Flow

the area of the design. But, now coming on power, the question comes how and when do we measure power in whole design flow. Today we have lot of tools to help us do PE at different stages of design like Spyglass Power and PrimePower by Synopsys and Questa Auto check by Mentor Graphics.

In this thesis work we are using two commercially available EDA RTL and GL power estimation tools to do PE. The most common PE techniques we know are GL PE with FE (Front-End) and GL PE with BE (Back-End) netlist. FE netlist is the one generated at the stage when the logic synthesis has been completed and the BE netlist is generated after P&R process as a part of BE design flow. So, the BE netlist is more detailed as placement of different cells, clock synthesis and routing takes place during the process and hence, has all the information about parasitics, wire load, clock tree etc. Now coming to the point, what is the issue with BE PE, when it is giving us very precise PE? Or why do we want to do PE at earlier stage of digital design flow?

For sure, BE PE is the most precise PE technique, but for it to be that precise, we have to wait for the most of the design process to be completed and it comes at very later stage of design. If there is any issue, or any optimisation that has to be made, the changes have to be made in RTL and hence, go through all the steps of synthesis and layout again in order to reach to the new modified BE netlist. This can be very time consuming and costly. Also, it can take a few iterations before we reach the desired PE for the IC. So, here comes a need to have PE system at a early stage of the design. Using any commercially available RTL power estimation tool we are able to do RTL PE at the Design Phase of the ASIC design flow and this process can be started as soon as RTL is available. Though it would not be as accurate as BE PE, but it has a fast turn around and can give designer a better idea of the expected power of design at the very early stage of the design flow. There are four points we can do power estimation to obtain power metrics in the ASIC design flow:

- RTL PE

- RTL PE with calibration
- Gate Level PE with FE netlist
- Gate Level PE with BE netlist

2.4 Power Estimation Methodologies

Designers can get power consumption estimate of their design from GL PE with front-end or back-end netlist. However, it seems a bit too late to get estimated power consumption after synthesis or place and route step in the ASIC design process, because it takes time to do synthesis, place and route, and to generate activity file for netlist. People want to estimate power consumption directly at RTL stage to save time.

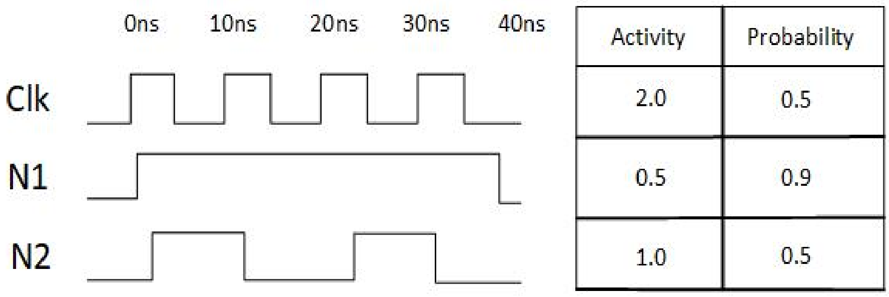
Power estimation techniques allow fast design exploration. These techniques can be considered at different levels of abstraction of the design, leveraging the time needed before having power estimates. Three methodology of RTL PE will be discussed in this part, among which, Simulation-based RTL PE methodology is the one been used in this thesis.

2.4.1 Probabilistic-based RTL Power Estimation methodology

Probabilistic-based RTL PE methodology is used when simulator is not available, when using this methodology, designer should provide the characteristic of input signal instead of real information of toggle. Normally they provided information are probability and activity, of the signal.

Probability: Defined as the percentage of the time that signal is high.

Activity: Activity is the number of toggles per unit time.



Activity	Probability
2.0	0.5
0.5	0.9
1.0	0.5

Figure 2.5: Example of signal activity and probability

As shown in the Figure 2.5, the time window is 0 ns to 40 ns, for the Clk signal, half of the time is high and half of the time is low, the probability of Clk is 0.5, at the same time, it toggle two times per clock cycle and so, the activity of Clk is 2.0 per clock cycle. The activity and probability of N1 and N2 can be calculated in the same way.With these two parameters, the probability of a node switch from zero to one and from one to zero can be calculated, which is the key information for calculating power consumption. If a design contains multiple clock domains, the activity for all signals are computed with reference to the fastest clock in the design,[7],[8],[9].

Using N2 node as an example. The probability is 0.5 which means that in a certain time it has 50% probability to be low and 50% probability to be high; the activity indicates that this signal toggle once per clock cycle. Using one clock cycle as time unit, at one clock cycle, the probability that N2 already being at high is 50%, it toggle once per clock cycle, then it must switch from one to zero in the time unit, as a result, the probability of switch from one to zero is 50%, also, the probability of switch from zero to one can also be estimated: also 50%. These parameters can be propagated through all nodes of the design according to corresponding function. After that estimated power consumption could be calculated,[6],[10].[11],[12].

2.4.2 Simulation-based RTL Power Estimation methodology

As the name suggests, available simulator is needed when using simulation-based PE. It consist of applying data stimuli to the inputs of the design under test and to perform a simulation to determine the corresponding outputs. After simulation, the switching information of every nodes are known. The power consumption could be calculated directly according to switching information,[6],[13],[14]. Depending on the abstraction level, the type of information that is required to obtain PE is different, going from current and voltage values, capacitance, clock frequency to the switching activities of all signals. For the gate level simulation: includes the use of logic parts such as NAND / NOR gates, latches, flip flops and interconnection networks. The most popular technique of assessment includes an event-driven model and the power consumption is predicted by computing the charging/discharging capacitance at the gate and by evaluating the activity of this node.

2.4.3 Statistical-based Power Estimation methodology

According to Monte Carlo method, the more random samples are used, and so the more accuracy the result will be. This method also works when estimating power consumption of integrated circuits when the power testbench for a certain design is not available. Random input signals are fed into the design, propagate the switch activity through every node and calculate power consumption. When the number of random input signal is enough, the averaged power consumption is acceptable,[6],[15],[16],[17].

Chapter 3

Method

The methodology of the thesis work will be discussed in this chapter. First section covers a brief introduction about the selected RTL PE tool, then will cover the topic of RTL power estimation, explaining the power flow and the goals used this thesis project. In the third section, integration of RTL power flow in a script is covered in detail and the last section of the chapter covers GL PE flow.

3.1 RTL PE tool

We selected a RTL PE tool from an EDA vendor to be used in this thesis. The selected tool helps users to estimate power consumption of their design at RTL design stage. It uses a fast area-based synthesis engine to map RTL code to netlist, unlike traditional synthesis, The tool only care about how the netlist looks like and how signals toggle on every internal net, the netlist generated by this tool is not a real netlist, it can be seen as a prototype.

After the fast synthesis, the selected RTL PE tool will use the information from activity files (VCD, FSDB, SAIF), lib files, initial RTL code, power parameter and the generated pseudo netlist to estimate power consumption. As an early design analysis tool with in-depth analysis at RTL design phase, the tool is widely used in industry. This thesis mainly focuses on the Lint and Power estimation flow.

3.1.1 The Lint Flow

The Lint flow gives RTL code a very deep check to find any typographical error like syntax error, misnamed variable and so on at the very early stage of design cycle, users can correct these errors as early as possible.

In Lint flow, users need to read RTL code, liberty files, parameter and some other critical information in the tool first, after that, the RTL code will be checked in different aspect according to different goals. Ericsson gives the code a very detailed check. Ericsson already has a script that can help user to run the Lint flow. Errors in lint flow need to be cleaned up before continuing to PE flow.

3.2 RTL Power Estimation

Power consumption is increasing with increase in device density on a single chip.In order to design power efficient ASIC that can meet increasing requirements on performance and functionality integration, it is important to estimate the power consumption of an ASIC very early in the design cycle. If the estimated power consumption is higher than the power budget then designers should be able to reduce it in a timely manner. General RTL power reduction techniques include the use of voltage domains, switchable power domains, and clock-gating. Because voltage and power domains require special logic circuits and make the power grid design more complicated, there is a need to verify that their implementation is correct .

In this section, the methodology of RTL PE flow is covered. This methodology was used on DUT and few other sub-blocks for this thesis work. In it we would be explaining about three different power goals that is Power audit, Power profiling and vector analysis, that we used in our entire RTL PE flow. Where "Power profiling" is the most useful power estimation and analysis goal, the other two goals are to check the validity of the design and input stimuli. Power estimation goals are readily usable at various phases of the IC design flow, such as Block/IP level and SoC integration level of the design.

The selected PE tool uses an area-based synthesis engine to map the RTL design onto a gate netlist. The engine can provide a reliable starting point for power calculation because it can use the majority of the cells in the specified target libraries. The tool also adds certain virtual cells and high fan-out nets to the clock tree to simulate them. When combined with simulation and parasitic data, the resulting netlist enables for an audit to confirm that the design, simulation data, and technology library are all consistent, as well as the generation of a list of the key parameters used in power estimation. The contribution of static (leakage) and dynamic (internal and switching) power to total power is also calculated by the RTL PE tool.

To run a Power goal for our project we did all the manual setup, which included coping various files generated after the basic design_read runs to a separate folder specifically designed for power estimation part. Editing some files to give more information about the wireload or library mapping, setting more parameters etc and then, once entire setup was done and we were in the PE working directory for the DUT then, there we ran power goals. Few of the power goals that we used for our project are explained below.

3.2.1 RTL PE Goals

- **Power_audit**

  The power_audit goal performs an audit to check the design, simulation data, and technology library for consistency and lists the key parameters used in power estimation. It produces a report, named *pe_ audit.rpt*, which is one of the most important report produced during the run. It summarizes information about the design.

- **Power_activity_check**

  In this goal we analyze the activity levels in the simulation testbenches, When the simulation results are available and generates a graph of activity over time. This will help in selecting relevant time intervals and time slices of simulations, that would be most useful for power estimation and reduction.

- **Power_est_profiling**

  The goal computes the estimated power as well as the activity and efficiency information for clocks, registers, and memories for time intervals of interest. It also uses this profiling information to identify inefficient clockgating, redundant memory access conditions, and similar power bugs in the design.

  It needs additional .lib file (which has list of all the libraries used for the design and at which corner and voltage) and another important requirement is to set the clock-gating threshold. This goal is a power reduction goal, that is reduces the power combinational, registers and clock portion of the design.

  Few of the important reports to look into after the goal has successfully run is the *pe_ summary.rpt*, which describes the various aspects of power consumption of the design, *pe_ wireload.rpt* which has information about all the hierarchical area and wireload of the DUT and *moresimple_ sevclass.rpt* to check for any warning or errors reported.

- **Power_calibration**

  Generates calibration data from a reference design. It generates the Design Constraint (DC) file that shows the percentage cell allocation in the current design. It also generates the clock buffer information from the reference design and the generated DC file also has the information about the advanced capacitance model based on input design and the corresponding SPEF file.

  The power_calibration goal automatically extracts models for key synthesis and back end characteristics from the netlist and applies them to the RTL design. Example of extracted models are:

  - Cell sizing: Impacts Combinational and Sequential Leakage (primary) and Dynamic (secondary)
  - Vt-mix: Impacts Combinational and Sequential leakage
  - Clock tree: Impacts all Clock Power and Sequential Dynamic
  - Capacitance model: Impacts all Switching activity

3.2.2 The RTL PE Flow

The flow aims to estimate power consumption at RTL stage. A simple flow chart of the selected power estimation methodology is shown in the Figure 3.1.

There are two types of methodologies for analysing power: Power estimation without calibration; and power estimation with calibration. The Figure 3.1 shows both methodologies. The flow in dotted line is unique for power estimation with calibration.

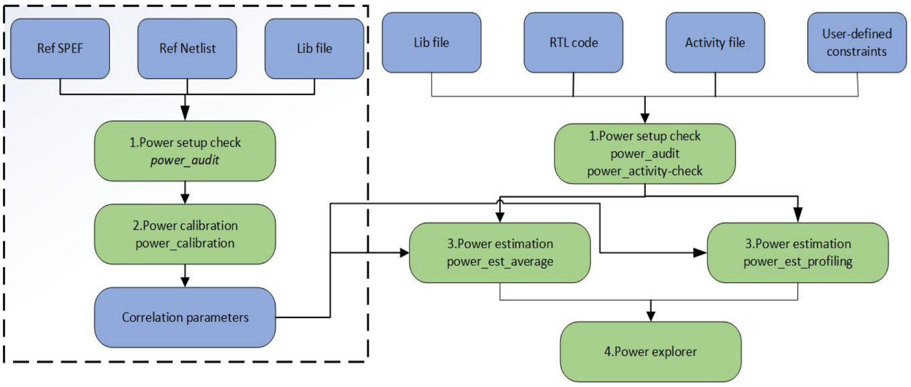
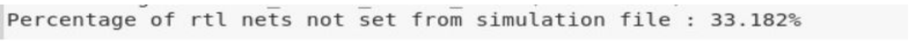


Figure 3.1: High level flow of the selected RTL PE tool and exploration methodology

Power Estimation without Calibration

As shown in the Figure 3.1, the first step of PE is reading the user-provided files like lib files, RTL codes, activity files and power parameters. The tool will read and analysis these files and parameters, by running the goal of *design_ read*. When all errors in the previous step has been fixed, then we go to the next step of running the goal of *power_ audit*. Running this goal, the tool will try to extract useful information from the files that has been read in, for example, tool will get voltage, capacitance, threshold voltage and information for power rail from each libraries. The tool also, tries to match the signal names found in the activity file with the net names given in RTL code. If a name is found in both sides, then the corresponding net is labelled as un-annotated. When all nets are checked, the tool will calculate the percentage of un-annotated nets and report it in *pe_ audit.rpt*, as shown in the Figure 3.2. It shows "the percentage of nets not set from simulation file" is 33.182% which means 33.182% nets in user-provided RTL codes did not have switching information from activity file and it results in causing inaccuracy in calculating power consumption. By going through all the un-automated nets reported and then matching them in test-bench and RTL code, until we got un-automated nets as close to 0%.



```
Percentage of rtl nets not set from simulation file : 33.182%
```

Figure 3.2: Percentage of un-annotated nets after running power_audit

The goal power_activity_check is used to analyse the activity of DUT. A graph of activity over time is generated as shown in the Figure 3.3, also all unannotated nets are reported in this goal. From the the Figure 3.3, we can see that the highest activity happens between 3e+10 fs and 5e+10 fs. It is better to define the power estimation time window where there is highest activity of the design because before 3e+10 fs, it could only the setup time for the design, 3e+10fs to 5e+10fs should be the time window where the design is processing data.

Once *power_ audit* and *power_ activity_ check* has been run successfully, other goals can be run. In our case, it is the goal of *power_ est_ profilling*, it gives a deep power analysis of RTL code. When running this goal, the tool first synthesizes, then does some optimizations on the created netlist. Then uses the activity and extracted parameters to calculate power consumption. After that, the tool reports the estimated power consumption, that is broken down in detail for different components like combinational, sequential, clock and memory.

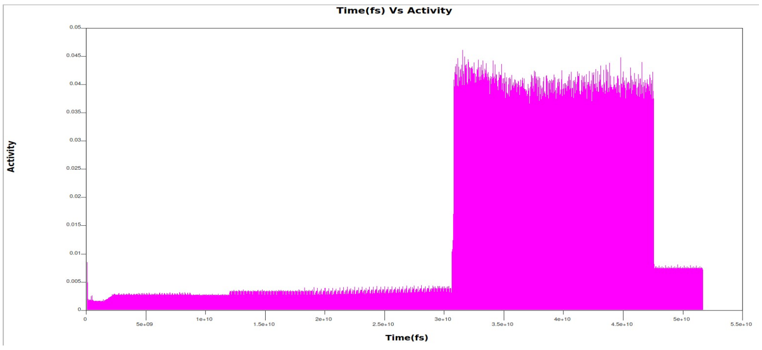


Figure 3.3: Graph of activity over time after power_activity_check

Power Estimation with Calibration

Power flow with calibration is to improve the accuracy of the estimated power consumption. It has two steps: extracting calibration parameters and estimating power consumption.

Extracting calibration parameters

The flow of extracting calibration data is shown in dotted line in the Figure 3.1. In this step, the tool needs reference netlist, SPEF file as shown in the Figure 3.1, in order to extract correlation parameters. RTL power estimation is done at very early design stage as soon as the first RTL code is ready and passed through simplified verification to ensure design functionality. The idea is to do RTL power estimation for every IP (small block) to provide feedback to each block designer on how power-efficient his/her block is so that designer can improve his/her design. Building calibration database does not need to do based on a big sub-chip (or subsystem), it can be done for a relatively small block with well-defined design structure (e.g. with 50% combinational and 50% sequential gates, no memory). This block can either be selected from the list of available blocks or be designed specially for this purpose. The calibration database will be used for improving power estimation accuracy for other blocks, sub-chip and subsystem without any significant degradation. Here is the list of the parameters that the RTL PE tool will extract after completing correlation step:

1. **Clock Tree**

   The RTL PE tool forms a dummy clock tree from netlist which will affect all the power consumption.

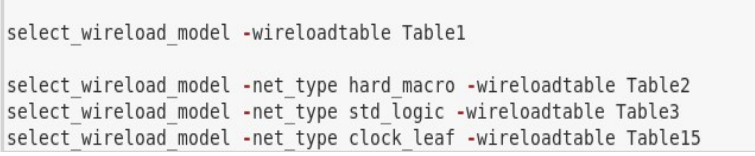
2. **Slew**

   Rising slew and falling slew are also extract from reference netlist, this parameter will affect internal power consumption.

3. **Information about Design Implementation**

   The power calibration process also gives us information on the standard cells used in the design including cell size, Vt (Threshold voltage) group. For example, they could be high leaking, or belong to cell group of different gate length.

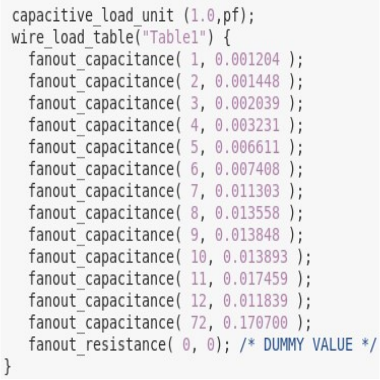
4. **Capacitance**



```
select_wireload_model -wireloadtable Table1

select_wireload_model -net_type hard_macro -wireloadtable Table2
select_wireload_model -net_type std_logic -wireloadtable Table3
select_wireload_model -net_type clock_leaf -wireloadtable Table15
```

Figure 3.4: Wireload table of different units



```
capacitive_load_unit (1.0,pf);
wire_load_table("Table1") {
  fanout_capacitance( 1, 0.001204 );
  fanout_capacitance( 2, 0.001448 );
  fanout_capacitance( 3, 0.002039 );
  fanout_capacitance( 4, 0.003231 );
  fanout_capacitance( 5, 0.006611 );
  fanout_capacitance( 6, 0.007408 );
  fanout_capacitance( 7, 0.011303 );
  fanout_capacitance( 8, 0.013558 );
  fanout_capacitance( 9, 0.013848 );
  fanout_capacitance( 10, 0.013893 );
  fanout_capacitance( 11, 0.017459 );
  fanout_capacitance( 12, 0.011839 );
  fanout_capacitance( 72, 0.170700 );
  fanout_resistance( 0, 0); /* DUMMY VALUE */
}
```

Figure 3.5: Expanded wireload table

The Figures 3.5 and 3.4 show the parameters that relate to capacitance and resistance. Figure 3.4 specifies different wire load tables for different kinds of nets and Figure 3.5 shows detailed information in wire load table (Table 1 for example). Table 1 in the Figure 11 shows the relationship between fanout and capacitance or resistance. For example, in the first item in Table 1, it means that when fanout is 1, the capacitance will be 0.001204 pf. These parameters will affect all switching power, as it is true for all the wire of the design.

Estimating Power Consumption

After power calibration process is done, and all correlation parameters are extracted, The RTL PE tool calculates power consumption in the same way as the power estimation without calibration does. The only different is that the setup file for goals is needed to be changed, which will be discussed in the Chapter four. The accuracy of the estimated power highly depends on the reference netlist.

3.3 Integrating power goals in the Lint flow

As mentioned in the above section, there is lot of manual work to be done in order to make proper setup ready to run RTL PE goals. So to overcome all these issues, we integrated power goals in the pre-existing script of Lint flow.

In this section the integration of power goals in the Lint flow is described in detail. Also in this section, the goal to read in the design and parameters is named as design_read. The Figure 3.6 describes the step by step flow of running the power goals. Further more detailed explanation of some steps is given below.

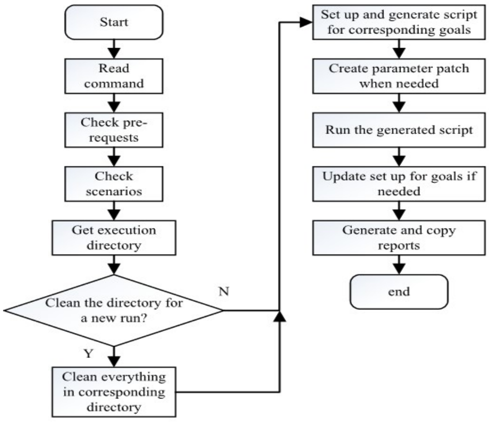

Start
Read command
Check pre-requests
Check scenarios
Get execution directory
Clean the directory for a new run?
N
Y
Clean everything in corresponding directory
Set up and generate script for corresponding goals
Create parameter patch when needed
Run the generated script
Update set up for goals if needed
Generate and copy reports
end

Figure 3.6: Flow chart for the selected RTL PE tool

- **Check Pre-requisites**

  It is used to detect design name. To make Power flow completely independent from the Lint flow, both flows are expected to be run in different directories. The default mode is Lint flow, but since both flows have its own default work directories, if the user stands in the default directory of power flow, the script will change the mode to power automatically.

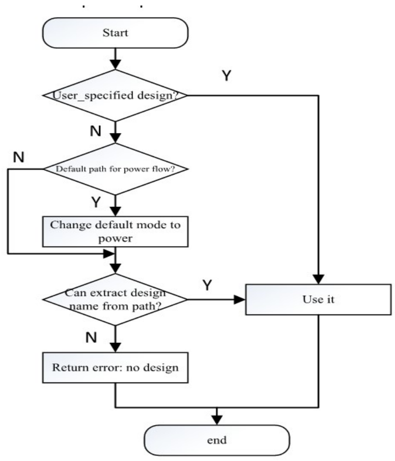

Start
User_specified design?
Y
N
Default path for power flow?
N
Y
Change default mode to power
Can extract design name from path?
Y
Use it
N
Return error: no design
end

Figure 3.7: Flow chart for checking pre-requisites

- **Check Scenarios**

  It is a common practice that the same block can be used to run for different scenarios with different parameters. Script supports user to write down different scenarios with different parameters in one single YAML file,[18]. and then run all the scenarios in one command.

- **Get Execution Directory**

  The flowchart of checking execution directory for the running power goals is shown in the Figure 3.8. Optionally, script also allow user to specify the execution directory by themselves.

- **Clean Everything**

  Sometimes, users modifies the code or some parameters in the model and want to clean everything in corresponding directory to avoid any garbage data, then this part of the script will be used. It will clean every set up files in work directory as well as all reports in result directory for a new run.

- **Set up and generate scripts for goals**

  To take advantage of resources such as storage, license and so on, the real command to submit the task is different with which user typed. This command is generated by script automatically in this part, also this generated command is based on IP Kit and LSF queue system.

- **Create Parameters Patch**

  This part of the code will only be activated when YAML file is used, script will generate a patch for all parameters in YAML file to make sure the RTL

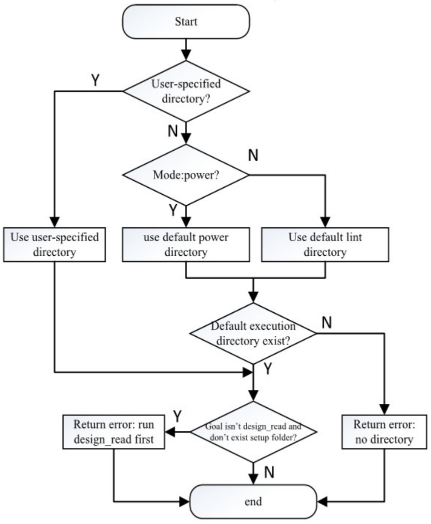

Start
User-specified directory?
Y
N
Mode:power?
N
Y
Use user-specified directory
use default power directory
Use default lint directory
Default execution directory exist?
N
Y
Return error: run design_read first
Y
Goal isn't design_read and don't exist setup folder?
Return error: no directory
N
end

Figure 3.8: Flow chart for checking execution directory

PE tool uses correct parameters when running both Power flow and Lint flow.

- **Update Setup Files for Goals**

  This part of the script will only be activated when running design_read, Authority of modify setup file will be return to script here. Figure 3.9 shows the flow chart of this part, "goal" here means all the power goals expect design_read. When running till here, script will update the goal setup file according to modes. The goal setup file for Lint flow and Power flow are totally different, in the same way, different goals also has its individual setup. For example, for power flow with calibration, the calibration mode needed to be turn on and the path for calibration data should be provided, while for the power flow without calibration, the calibration mode is kept turned off.

- **Copy Reports**

  Copying the final generated analysis reports, is the last part of the whole flow, several critical reports are copied to latest directory results directory. If the user runs a goal several times in the same execution directory, the "latest directory" always point the latest run, user can have a chance to compare the results with the latest run and previous run.

3.4 Gate-level power estimation using the selected GL PE tool

The GL PE tool is used for accurately analyzing power dissipation of cell-based designs. It is intended as an advanced solution for ASIC and structured custom

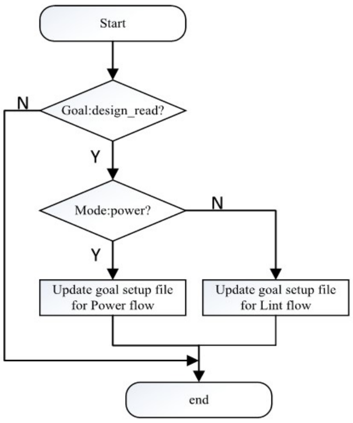

Start
Goal:design_read?
N
Y
Mode:power?
N
Y
Update goal setup file for Power flow
Update goal setup file for Lint flow
end

Figure 3.9: Flow chart for updating goal setup files

circuit designers. The activity vectors are either RTL or gate-level simulation results in the Value Change Dump (VCD) format, Fast Signal Database (FSDB) format, or Switching Activity Interchange Format (SAIF).

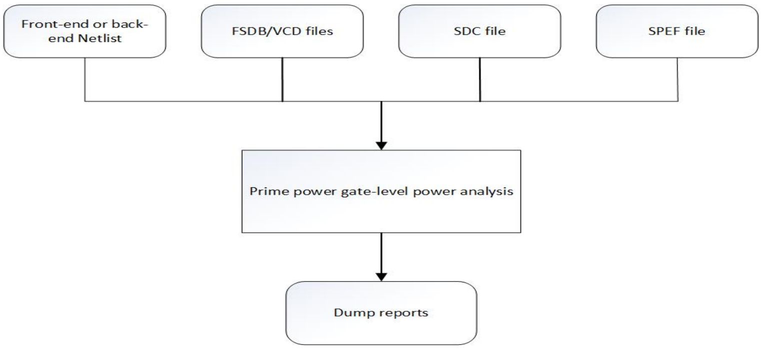

Front-end or back-end Netlist
FSDB/VCD files
SDC file
SPEF file
Prime power gate-level power analysis
Dump reports

Figure 3.10: Analysis on input files for the selected GL PE tool

GL PE tool uses front-end or back-end netlist as input and as they are produced much later in the design cycle, they have much more information about parasitic capacitances of each node, slew rate, glitch, wire load and what cells are finally used. Therefore, the GL PE tool is able to collect more detailed information of the design than RTL PE tool does. The power estimated by GL PE tools is more accurate than the estimate provided by RTL PE tools. We give following inputs to the GL PE tool to perform power analysis:

- Logic library: A cell library containing timing and power characterization information for each cell.
- Gate-level netlist: A flat or hierarchical gate-level netlist in Verilog, VHDL,

containing leaf-level instantiation of the library cells.

- Design constraints: An SDC file containing design constraints to calculate the transition time on the primary inputs and to define the clocks.
- Switching activity: The design switching activity information for averaged power analysis or accurate peak power analysis.
- Net parasitics: A parasitics file (SPEF) containing net capacitances for all the nets present in the design.

We specify all the above mentioned files in a document, along with the corner we are running the design at, hierarchy , block name and time window stating, which timing part is considered from FSDB (the window is decided based on highest activity of DUT in the whole time frame).

Power Report Analysis

Once the GL PE tool has run successfully without any errors, lot of reports are generated, covering each and every aspect of the power estimation. In this section we would discuss about some of the reports we found to be really important for our project.

- glpe_avg_activity_hier.rpt : Reports Toggle count/net, Glitch count/net, Toggle rate/period/net and Glitch rate/period/net in all available hierarchical levels of the design.
- glpe_hier_power_<freq>MHz.rpt : Reports power estimates at all available hierarchical levels of the design (as of the netlist). It is segregated by specifying - hierarchy option while generating reports.
- glpe_summary.rpt
- glpe_total_power_<freq>MHz.rpt : Reports a summary on total power estimates of the design per component. The estimated clock-tree power is also included.

Chapter 4

Comparison and Result

The flow described in the Chapter 4 is designed for power estimation at RTL level, as the netlist generated by the RTL PE tool is just a pseudo netlist, the accuracy of the estimated power needs to be discussed. In this chapter four flows are: RTL power estimation w. calib, RTL power estimation w/o calib, GL power estimation w. FE netlist and GL power estimation w. BE netlist will be compared to evaluate the accuracy of RTL power estimation flow.

4.1 Synchronize setup between four flows

To estimate the power for all four flows, they must first be synchronized and compared in terms of setup, time period, analysis corner, and configurations.

- **Lib files:**

  All four flows requires lib files, which have the information about different technology nodes and variety of cells and so, the final estimated power highly depend on which lib is used. As different technology nodes and thresholds affect the leakage power and internal power differently.

  The lib files that used in all the four flows were made absolutely same. Also, apart from the type of lib files, corner should also be matched, different corner will cause huge different of power estimated. especially leakage power.

- **Activity File :**

  Some design have special parameters which are mentioned in Activity file, and different parameters could cause the design work under differently. The activity file should match the scenarios that the design is working and it should be the same of all four flows.

  Both RTL and GL PE tools access the activity file, they try to map the name of net in the design with the name of net in the activity file. When a net in the design match with a net in the activity file, the net will be labelled as "annotated", if the percentage of annotated net close to 100 percentage. activity file and design are matched.

- **Time Window for power estimation :**

  The design has setup stage and running stage, in setup stage, most nets are not toggling and so consuming relatively small power. At the running stage, most of nets are toggling therefore design consumes much higher power in comparison to the setup stage. Since we are interested mainly on the power of the design consuming during its operational mode, the running state is of our interest. A suitable time window has to be selected to make sure tool only analysis power of running stage. Also, the time window for all flows should be the same.

- **Clock buffer, clock gate cell and clock gate threshold :**

  In RTL power estimation flow and GL power estimation with FE netlist, there are no real clock tree, tools will estimate a virtual clock network in the design. To make the clock power more accurate, it is to provide the clock buffer and clock gate cell that will be used in the real back-end netlist.

  Also, the number of clock-gates itself could cause extra power consumption, normally there exists threshold in real netlist, which allows implementing a clock gate for the register whose width is greater then the threshold. Designer wants one clock gate be able to gate several bits output, because if the clock gate only gate one bit register output, the power that is saved by gating clock may be less then what clock gates itself consumed.

  The clock-gating threshold will affect both clock power and sequential power. It is better to synchronize this parameter in all flows.

4.2 Comparison between four flows

The comparison of four flows will be discussed in this section of the chapter. The factors that affect power are summarised in the Table 4.1

Table 4.1: Factors that affect power consumption

	Leakage	**Internal**	**Switching**
Netlist/Prototype	yes	yes	yes
Number of cells	yes	yes	yes
Lib(Threshold voltage)group	yes	yes	yes
Cell size	yes	yes	yes
Slew rate	no	yes	no
Wire load	no	no	yes

The netlist that has been used in the four flows are different. The RTL PE tool only generates a pseudo netlist, whereas GL PE tool uses the real netlist (generated by the synthesis flow).

The area of the cells surely affects all power components. If a cell with smaller size is selected, the leakage power will get smaller compared with using larger size cell. Also, if a large cell is selected, the internal capacitance and pin capacitance will both getting higher, which will increase internal power and switching power respectively.

Technological libraries provide detailed information on timing, power, etc. for all cells used by the design with different sub-threshold voltage values, different voltage supply options, with different gate-length, fan-outs, and at different corners (i.e. process, voltage supply, temperature). The selection of correct library, correct corner plays a huge role in any of the flows to produce a realistic and trusted power estimation results.

Since the netlists utilized in the four flows differ in terms of lib group, cell, and wire load, as well as the number of cells in the netlists, all of these factors could alter the design's power estimation.

4.2.1 RTL PE without calibration vs RTL PE with calibration

Calibration extracts the clock tree, wire load table, size group, lib group, and slew rate from the reference netlist, as explained in the previous chapter. Because there is no reference netlist when running RTL PE without calibration, the wire load is considered to be zero, and all other factors are set to their default values.

Table 4.2: Difference between RTL PE with calib and without calib

	Leakage Power	Internal Power	Switching Power	Total Power
Total power	-53.53%	9.92%	236.23%	2.83%
Combinational power	-51.52%	52.96%	249.92%	13.79%
Sequential power	-64.21%	3.78%	216.24%	-25.86%
Memory power	0.00%	0.00%	395.86%	0.05%
Clock power	-0.84%	-2.15%	224.36%	109.54%

Table 4.2 shows the ratio of RTL PE with calib and without calib. the reference netlist is of a sub-chip that is used as DUT for our project. The percentage change in Table 4.2 is calculated by equation 4.1.

$$PwrDiff = \frac{P\_WithCalibration - P\_WithoutCalibration}{P\_WithoutCalibration} * 100\% \tag{4.1}$$

Looking at the leakage power first, as described in chapter 2, the value of leakage power highly depends on the libraries used, Table 4.3 lists the libraries that have been used in the design, in the table they are classified by threshold voltage. The number in the table represent relative value, 1 means highest speed or high leakage power, that is smaller technology node is used.

Table 4.3: Libs used in the design(memory lib not included)

	SLVT	LVT-GL1	LVT-GL2	LVT-GL3
speed	1	2	3	4
leakage power	1	2	3	4

In RTL PE flow without calibration, as there were no constraints defined for lib group therefore the tool used LVT-GL1 cells automatically for the DUT. Unfortunately, in the reference netlist there are 95% LVT-GL3 cells and 5% LVT-GL2 cells used, and there are 0% LVT-GL1 cells used. That is why, there is about 50% decrease in the *total leakage power* and most of it is contributed by combinational and sequential sources. There is no change in memory leakage power, as the memory libraries defined for both runs are same.

Internal power is affected by several factors like slew rate and cell size. The DUT is a very big design, slew rate would not change internal power much, as the value given for it in design constraints file is very small, the critical factor is cell size here. When the size of the cell gets bigger, the internal capacitance also increases and so, the power consumed by charging and discharging of internal capacitance will also increase. For the combinational circuits, the internal power increases by about 52%, which shows that quite big cells are used for combinational part of the design and so, the DUT has large combinational area. Also, the number of combinational cells in design are much more compared to sequential cells, and so there is only 3.78% increase in switching power between the two flows.

Switching power is affected by wireload. When running RTL PE without calibration the tool uses zero wireload, while RTL PE with calibration uses the wireload that is extracted from reference netlist. The wireload in the latter case, therefore, is nonzero causing the total switching power change about 230%.

4.2.2 RTL PE vs GL PE with Front-end netlist

This section focuses on the difference between RTL power estimation and GL PE with front-end netlist. At RTL level, there are no glitch exists in corresponding activity file, hence, the activity file for front-end netlist is generated without SDF file.

Table 4.4 shows the power difference in percentage between the RTL PE without calibration and GL PE with FE netlist. The Table 4.5 shows the power difference in percentage between the RTL PE with calibration and GL PE with FE netlist. Note that the percentage values showed in Tables 4.4 and 4.5 are calculated by equation 4.2 and equation 4.3, respectively.

.

The Tables 4.6 and 4.7 show the lib groups used in the three flows for combination cells and sequential cells.

Table 4.4: Power difference between RTL PE without calibration and GL PE w. FE

	Leakage Power	Internal Power	Switching Power	Total Power
Total Power	30.72%	13.30%	-50.46%	5.49%
Combination Power	-9.76%	-3.08%	-57.70%	-22.79%
Sequential Power	234.08%	36.23%	-65.19%	80.71%
Memory Power	-0.01%	-0.21%	-98.60%	-1.15%
Clock Power	-52.37%	-42.92%	-35.17%	-42.06%

Table 4.5: Power difference between RTL PE with calibration and GL PE w. FE

	Leakage Power	Internal Power	Switching Power	Total Power
Total Power	-39.26%	24.54%	66.57%	8.47%
Combinational Power	-56.25%	48.39%	48.01%	-12.14%
Sequential Power	19.57%	41.39%	10.10%	33.97%
Memory Power	0.00%	-0.22%	-93.04%	-1.10%
Clock Power	-52.77%	-44.15%	110.27%	21.40%

$$PwrDiff\_RTL\_PE\_FE = \frac{Pwr\_WithoutCalib - Pwr\_FE}{Pwr\_FE} * 100\% \tag{4.2}$$

$$PwrDiff\_RTL\_PE\_Calib\_FE = \frac{Pwr\_WithCalib - Pwr\_FE}{Pwr\_FE} * 100\% \tag{4.3}$$

Table 4.6: Combinational cells library group used in all three flows

	SLVT	LVT-GL1	LVT-GL2	LVT-GL3
RTL PE w/o calibration	0.0%	100.0%	0.0%	0.0%
RTL PE w. calibration	0.0%	0.0%	15.8%	84.2%
GL PE w. FE netlist	5.0%	9.0%	56.4%	29.6%

Table 4.8 shows the comparison of number of cells used in all flows, the numbers in this table are scaling factor, we have used number of clock gate cells in RTL PE without calibration, as the base of calculating the scaling factor, as they were least in the number. Taking an example, the scaling factor of front-end combination cell is 127.171 which means that the number of combinational cells in front-end netlist is 127.171 times of the number of clock gate cells in RTL PE without calibration.

Table 4.7: Sequential cells library group used in all three flows

	SLVT	LVT-GL1	LVT-GL2	LVT-GL3
RTL PE w/o calibration	0.0%	100.0%	0.0%	0.0%
RTL PE w. calibration	0.0%	0.0%	0.6%	99.5%
GL PE w. FE netlist	1.4%	3.3%	20.6%	74.7%

The scaling factor for RTL PE with calibration and front-end netlist are calculated using the equation 4.4 and 4.5 respectively.

Table 4.8: Comparison of cell number in all three flows

	Combination cells	Sequential cells	Clock gate cells
RTL PE w/o calibration	108.79	31.43	1.00
RTL PE w. calibration	108.78	31.43	1.00
GL PE w. FE netlist	127.17	12.33	1.09

$$Scaling = \frac{\#Cells\_Withcalibration}{\#Cells\_ClockGateCell\_WithoutCalibration} \tag{4.4}$$

$$Scaling = \frac{\#Cells\_FrontendNetlist}{\#Cells\_ClockGateCell\_WithoutCalibration} \tag{4.5}$$

RTL PE w/o calibration vs GL PE w. Front-end netlist

As shown in table 4.6, RTL PE Without calibration uses 100% of LVT-GL1 lib for combination cell, while GL PE only uses only 9.03% of LVT-GL1 lib for combination cell and rest is divided between LVT-GL1 and LVT-GL2 cells. As discussed, LVT-GL1 will contribute the second highest to the leakage power, ideally the combinational leakage power in RTL PE should be higher then that in GL PE if other factors are perfectly matched. However, the combination cells that RTL PE, used were of low threshold than that of GL PE, which pulled the leakage power down for RTL PE, but as from table 4.8, combinational cells used in front-end netlist are about 20% more than that what used for in RTL PE, which could be the reason for 30% increase in *total leakage power*. Also, the different of number of cells can explain the reason of smaller combination internal power in RTL PE flow.

The number of sequential cell in RTL PE is almost twice than that in GL PE as shown in table 4.8, at the same time, as shown in table 4.7, in the GL PE more than 70% of sequential cells are LVT-GL3 which could cause less leakage compared to LVT-GL1, while in RTL PE flow 100% high leakage power cells (LVT-GL1)¨are used. These two are critical factors would pull the leakage power up for RTL PE flow. Also, the number of cell can make the internal power higher in RTL PE.

The *clock power* could be excluded of the discussion, because for RTL PE and GL PE with FE netlist, both flows estimate a virtual clock tree as clock tree only is actually available at back-end stage. The netlist for GL PE is a real front-end netlist, while the netlist for RTL PE flow is just a pseudo netlist, that is why it is common for both flows get different clock power. It is common to have smaller *switching power* in RTL PE, for all types of cells, be it combinational, clock or memory, because as described before, zero wireload is assumed in RTL PE without calibration and for GL PE with front-end netlist there was some estimated wireload given. The *total switching power* for RTL PE is about 50% less than that for GL PE with front-end netlist.

RTL PE w. calibration vs GL PE w. Front-end netlist

As described in previous chapter, calibration introduces different parameters into RTL PE flow, due to which it can be seen in table 4.6, that 84% of combinational cells for calibration flow are LVT-GL3, whereas only 30% LVT-GL3 for GL PE flow. This tells us why after calibration, the combination leakage power becomes smaller then that in GL PE.

It is common for sequential leakage power in RTL PE flow to be more then that in GL PE flow after calibration, the critical factor is the number of cells here. As shown in table 4.7, after calibration, the percentage of LVT-GL3 is 99% and 75% respectively for the two flows, their number are close compare with other groups, which means that the increase in the leakage power for RTL PE with calibration flow is caused by number of cells which is almost 2.5 times for RTL PE with calibration flow.

Switching power can be divided into two part: switching power of pin capacitance and switching power of wire capacitance. Pin capacitance is defined in lib files, it could be affected by the size of cells, while wire capacitance is defined in wire load table after calibration. The size group and wire load table comes from reference netlist are too large for the DUT. The large size group can also explain the high internal power after calibration.

It can be seen in Table 4.5 that even after calibration the switching power for memory in RTL PE flow is still very low. There are two reasons that can explain this, the first is that the average toggle activity of memory output in RTL PE flow is smaller then that in GL PE flow and the second reason, is that there are no memory units in the block whose calibration data is used for DUT and hence, there is no wireload model for the memory.

4.2.3 GL PE w. Front-end netlist vs GL PE w. Back-end netlist

The comparison of GL PE w. FE and BE netlist will be discussed in this section. Table 4.10 and 4.11 shows the lib group that used in both netlists, table 4.12 shows the scaling factor of cell number for both netlists, the scaling factors are calculated by the same way as described above. The baseline is the number of cells in clock network, table 4.9 shows the difference of power consumption between GL PE w. front-end netlist flow and GL PE w. back-end netlist flow and the percentage in it is calculated by equation 4.6.

Table 4.9: Difference between GL PE FE and BE netlist

	Leakage Power	Internal Power	Switching Power	Total Power
Combination Power	65.27%	133.33%	67.63%	74.96%
Sequential Power	45.91%	17.51%	-23.68%	22.35%
Memory Power	0.00%	2.15%	87.67%	2.28%
Clock Power	73.83%	133.33%	26.07%	70.28%

Table 4.10: Combinational cells ratio front-end vs back-end

	SLVT	LVT-GL1	LVT-GL2	LVT-GL3
GL PE w. Front-end netlist	5.00%	9.03%	56.40%	29.58%
GL PE w. Back-end netlist	4.57%	8.62%	31.48%	55.33%

Table 4.11: Sequential cells ratio front-end vs back-end

	SLVT	LVT-GL1	LVT-GL2	LVT-GL3
GL PE w. Front-end netlist	1.44%	3.26%	20.56%	74.74%
GL PE w. Back-end netlist	7.26%	1.35%	20.74%	70.65%

$$PwrDiff = \frac{Pwr\_BE - Pwr\_FE}{Pwr\_FE} * 100\% \tag{4.6}$$

Focusing on the *leakage power* first, we see that leakage power is high for all the components except for memory. This is because, the number of cells in BE are higher than in FE and each cell contributes to leakage. Also in BE, the ratio of SLVT cells used is also quite big compared to FE for sequential circuits, and these SLVT cells are high leaking cells, again adding to the leakage power.

Coming to the *switching Power*, from the table 4.9 it differs a lot for different components of a design. For combinational circuit there is increase of about 67%, whereas for sequential circuit it decreases by 24%, this variation can be explained

Table 4.12: Comparison of cell number of in two flows

	Combinational	Sequential	Clock network
GL PE w. Front-end netlist	116.564	8.914	1
GL PE w. Back-end netlist	167.619	9.750	1.919

Table 4.13: Breakdown composition of sequential leakage power

	SLVT	LVT-GL1	LVT-GL2	LVT-GL3
GL PE w. Front-end netlist	12.92%	12.21%	11.97%	62.90%
GL PE w. Back-end netlist	45.82%	2.89%	10.97%	40.31%

by having a look at the table 4.10, 4.11 and 4.12. There has been increase in LVT-GL3 combinational cells and overall increase in total number of combinational cells for BE, therefore switching power is bound to increase. For sequential circuit, even though there was a very small increase in total number of sequential cells but percentage of LVT-GL3 cells also reduced a bit for Back-end, thereby explaining decrease in switching power for sequential circuits.

It is easy to see that the contribution of SLVT lib when combine table 4.13 and table 4.11, only 1.44 % SLVT lib contribute almost the same percentage of leakage power that 20% LVT-GL2 lib cause. 7.26% SLVT cells contribute nearly half of total leakage power.

The clock network in the BE netlist is the real clock network since the BE netlist is obtained after PnR, whereas the clock network in the FE netlist is simply what the GL PE tool predicts from the information in the FSDB, libs, RTL, and any other files provided for the run. As a result, all of the clock's power matrices have increased dramatically. It's worth noting that the switching power discrepancy for memory block is due to the very low power value. For instance, if the value in GL PE with FE netlist is 1uW and becomes 3uW in GL PE with BE netlist, the inaccuracy is 200 percent. Because the memory lib for both the netlist is identical, it is dominated by the wireload.

4.3 Improving accuracy of RTL power estimation flow

The projected power usage can be influenced by a number of things. This section will go over some methods that can be used to improve the accuracy of the RTL PE flow.

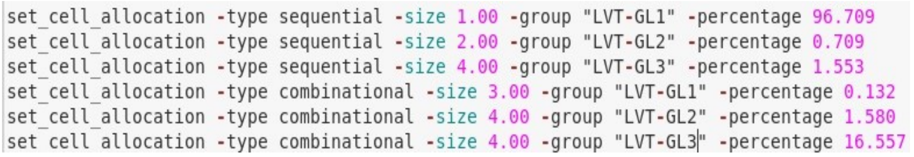
RTL power estimation with DUT calibration

As stated in this chapter, the number of cells, cell size group, and lib group all have a significant impact on leakage and internal power. We attempted to align the size group and the library group, as well as to improve the accuracy of the projected power usage.

The RTL PE tool will automatically extract the size group and lib group while extracting calibration data. When using the DUT netlist to extract calibration data, the size and library groups can be considered as aligned automatically. The Table 4.14 compares the RTL PE for DUT with DUT calibration and GL PE with FE netlist for DUT. It provides more accurate internal and leakage power values at the RTL PE stage. However, there is still inaccuracy between the two flows. The reason for this is that, as illustrated in Figure 4.1 shows how RTL PE tool defines the lib and size group and only uses a percentage of the specified libs. When specifying the lib and size group, RTL PE tool will try to get close to the defined percentage values. However, cannot completely replicate the structure of a generated pseudo netlist. It may also explain the high switching power error.

Table 4.14: Difference between GL PE w. FE netlist and RTL PE with DUT calibration, for DUT

	Leakage Power	Internal Power	Switching Power	Total Power
Combination Power	4.45%	34.81%	108.90%	38.74%
Sequential Power	52.38%	43.57%	128.85%	51.06%
Memory Power	0.00%	2.15%	-66.10%	2.28%
Clock Power	-52.77%	-42.29%	291.92%	101.21%



```
set_cell_allocation -type sequential -size 1.00 -group "LVT-GL1" -percentage 96.709
set_cell_allocation -type sequential -size 2.00 -group "LVT-GL2" -percentage 0.709
set_cell_allocation -type sequential -size 4.00 -group "LVT-GL3" -percentage 1.553
set_cell_allocation -type combinational -size 3.00 -group "LVT-GL1" -percentage 0.132
set_cell_allocation -type combinational -size 4.00 -group "LVT-GL2" -percentage 1.580
set cell allocation -type combinational -size 4.00 -group "LVT-GL3" -percentage 16.557
```

Figure 4.1: Lib and size group

The wire load table is taken from the DUT's real netlist, yet there is still an error, indicating that the generated pseudo netlist by the RTL PE tool does not match the wireload specified in the real netlist, as the pseudo netlist has numerous huge fan-out nodes.

As described in this chapter,the inaccuracy of the generated pseudo netlist by RTL PE tool in comparison to front-end netlist is very hard to be controlled. But since it affects the accuracy in estimating power during RTL PE with calibration process, we need to find solutions to address it. A suitable power scaling factor can be used to improve the accuracy of power estimation.

Power scaling factor

This section shows how to get the suitable power scaling factor.

The RTL PE tool will retrieve calibration data from an appropriate reference netlist. A appropriate reference netlist is required to obtain a suitable power scaling factor; the reference netlist should have a structure that is similar to that of the DUT. In this thesis, a suitable sub-block for DUT is chosen, which is referred to as "Block A."

For Block A, run GL PE with FE netlist to get a reference power consumption.

Run RTL PE for Block A with Block A calibration data. When running Block A with Block A calibration, the lib and size group of Block A are aligned automatically.

The number of cell should be aligned because this factor can affect all types of power consumption, Table 4.15 is used as a simple example to calculate the power scaling factor (use combinational power as an example).

Table 4.15: Example for calculate power scaling factor for Block A

Combination Power	Leakage Power	Internal Power	Switching Power	Number of cell
RTL PE w. calib of Block A	10 mW	15 mW	20 mW	300
GL PE w. Front-end netlist	18 mW	25 mW	30 mW	600

To get the power scaling factor, we need to use suitable mathematical model to predict the power consumption when the number of cell in both side are same. In this thesis, equation 4.7 is used to calculate the power consumption, which in turn is used to calculate the power scaling factor.

$$Predicted\_Power\_RTL\_PE = \frac{\#Cells\_RealNetlist}{\#Cells\_PseudoNetlist} * Expected\_Power\_RTL\_PE \tag{4.7}$$

According to the equation 4.7, if the number of cells in the pseudo netlist are made same to the one in GL PE with FE netlist that is 600, then the power consumption are expected as 20 mW, 30 mW, 40 mW respectively. The power scaling factor can be calculated by using the equation 4.8. The power scaling factor for this example is 0.9, 0.83, 0.75 for leakage, internal and switching respectively as shown in table 4.16. These factors can be treated as the calibration, of the inaccuracy of pseudo netlist generated by the RTL PE tool.

$$PwrScaling\_BlockA = \frac{PwrConsumption\_RealNetlist}{Predicted\_PwrCumption\_RTL\_PE} \tag{4.8}$$

When calculating the power scaling factor, the factor of number of cell is excluded from the calculation. So, when applying the scaling factor on the DUT,

Table 4.16: Calculating power scaling factor for Block A

Combination Power	Leakage Power	Internal Power	Switching Power	Number of cell
Expected RTL PE w. calib of Block A	20 mW	30 mW	40 mW	600
GL PE w. Front-end netlist	18 mW	25 mW	30 mW	600
Power scaling factor	0.90	0.83	0.75	1.00

the factor of number of cell should be added back. The power scaling for DUT can be calculated, using the equation 4.9.

$$PwrScaling\_DUT = PwrScaling\_BlockA * \frac{\#Cells\_RealNetlist}{\#Cells\_PseudoNetlist} \tag{4.9}$$

Table 4.17: Difference between GL PE w. FE and RTL PE w. Calib for DUT with Block A calibration

	Leakage Power	Internal Power	Switching Power	Total Power
Combination Power	-4.12%	10.93%	-10.97%	-4.11%
Sequential Power	-7.44%	19.67%	-32.53%	9.62%
Memory Power	0.00%	-0.21%	-90.76%	-1.08%
Clock Power	-52.77%	-44.22%	227.94%	73.10%

Table 4.17 shows the difference between GL PE with FE netlist and RTL PE with calib, for DUT with Block A calibration data. The results have been compared after aligning lib and size groups and corresponding scaling factor for both the flows. From the table we can see that, the combination power and sequential power are close to the power obtained from GL PE wi. FE netlist flow. It should be noted that the memory power is not scaled because there is no memory in the reference netlist; additionally, because RTL PE and GL PE with FE netlist, both estimate a virtual clock tree network, it is difficult to say which one is correct and hence it is meaningless to do scaling of the clock power.

Till now, RTL PE projected power consumption has been close to that of the FE netlist. However, there is still a difference in FE and BE netlist power consumption, which is influenced by cell group, size group, and cell count. But, because BE netlist is typically only available once the entire design is complete, it is more practical and therefore preferable to make the estimate of RTL PE flow close to the estimate of GL PE with front-end netlist.

Improving the accuracy of RTL power estimation

1. Check the lib and make sure the lib and corner that provided to RTL PE tool are correct. It is better to provide only useful lib (i.e. do not provide the libs that will not be used) because when calibration step has not been done (or not available) the libs that are used by the RTL PE tool might be totally wrong. In case useless libs are provided the RTL PE tool will include it in the default lib list and use it even that libs should not be used at all. For example, in the reference netlist 0% of LVT-GL1 lib is used, but when running RTL PE for this block, if the LVT-GL1 still is provided, the RTL PE tool will use 100% LVT-GL1 lib causing library cell usage being completely incorrect.

2. Select a suitable reference netlist for the RTL PE tool to extract calibration data, if possible, use the reference netlist to calculate power scaling factor for all types of power consumption.

3. Assign lib groups and size group carefully when running RTL PE flow with calibration. After extracting calibration data, the RTL PE tool will generate a Design Constraint (DC) file, should check that file properly and modify it if needed.

4. Use correct lib of clock network buffer and clock gate, also the clock gate threshold should be set to a suitable value.

5. When the estimated power is calculated, check the leakage power first, if the leakage power are acceptable, then the library and corner are correct.

6. Check the number of combinational cells and sequential cells (number of cells, not number of nets driven by cells), if these number are not correct, a suitable scaling need to be set.

7. If the lib group, appropriate scaling, and wireload tables are acceptable, the cell size is considered, and a suitable size group is chosen, the cell size might be the critical cause of error in all forms of power.

Chapter 5
Conclusion

In this chapter a summarized version of the whole thesis project is presented as part of the final conclusion. Also, some comments are mentioned that can be considered as possible future work based on the work done in this thesis project.

As mentioned before, there were two main objectives for this master thesis project titled "*RTL Power Estimation and Optimation Flow for 5G Radio Products*". An automated RTL PE flow has been built in this thesis, to fulfill the first goal. The Design-Under-Test (DUT) was a sub-chip from Ericsson, we used the netlist of the sub-blocks (Block A and Block B) as the reference netlist, in order to extract calibration data that was used for running RTL PE with calibration. The four flows that we worked on were: RTL PE without calibration, RTL PE with calibration, GL PE with front-end netlist and GL PE with back-end netlist and compared them in this thesis, to fulfill the requirements of second goal. The PE results from GL PE flows have been treated as the reference.

Because the RTL PE tool only constructs a pseudo netlist to estimate power consumption, the generated netlist's structure and cells may differ from the real netlist. From the reference netlist, RTL PE with calibration flow may extract parameters such as lib group, size group, slew rate, and wireload table. If the lib group, size group, and slew rate are near to the real netlist using the calibration data from the reference netlist, the leakage and internal power could be close to what is acquired during the flow of GL PE. However, even if the reference netlist is the sub-chip itself, the accuracy of *switching power* cannot be guaranteed, because the structure of the pseudo netlist may differ from that of the real netlist. The selected RTL PE tool, at the RTL PE stage, is unconcerned about timing and may be able to map very large fan-out nodes. Real netlist, on the other hand, cannot use too many huge fan-out nodes.

When comparing RTL PE without calibration to RTL PE with calibration, and using GL PE with FE netlist findings as a reference, we can observe that by tweaking the following knobs, such as Cell mix, switching activity, and library selection, we were able to acquire substantially better power estimation. We can't do much with cell mix in RTL PE without calibration, but we have more control over a few knobs in RTL PE with calibration flow. We now know that at GL PE with BE netlist, a large number of cells with LVT-GL3, LVT-GL2, and certain

SLVT cells are used to close the timings, after conducting numerous runs and combinations of trying different cell mixes.

We also know from our work that during the RTL PE without Calibration stage, the tool tries to pick the smallest cells available in the libraries, as we just have a pseudo netlist for reference and no actual clock in the design at that point. As a result, the simulation is rather optimistic. To get a more precise power estimation, it's better to use the calibration data. It is also necessary to utilize a sufficient power scaling factor, which necessitates the usage of a proper reference netlist by designers.

5.1 Future work

As described in chapter 4.3, the a suitable power scaling factor can improve the accuracy of estimated power consumption significantly. In order to calculate the power scaling factor in this thesis we assumed a simplified (linear) dependency between the total number of cells in the pseudo netlist and the total number of cells in the front-end netlist (shown in the equation 4.7). However, the dependency should be more complex taking into account all individual cell types and libraries. This can well be a good topic for the future works.

The big difference in estimating clock power between RTL PE with/without calibration and GL PE with FE netlist is another issue that we could not find a way to fix in this thesis. It is clearly a good topic for the future works.

References

[1] https://www.einfochips.com/blog/asic-design-flow-in-vlsi-engineering-services-a-quick-guide, Jul. 15, 2021

[2] https://en.wikipedia.org/wiki/Apple_M1, Jul. 15, 2021

[3] https://en.wikipedia.org/wiki/GeForce_30_series, Jul. 15, 2021

[4] S. Rennan Nesset."RTL Power Estimation Flow and Its Use in Power Optimization" Available: `https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2558598`

[5] M. R. Perleberg et al., "ASIC power-estimation accuracy evaluation: A case study using video-coding architectures," *IEEE 9th Latin American Symposium on Circuits & Systems (LASCAS),* 2018, pp. 1-4, doi: 10.1109/LASCAS.2018.8399919.

[6] Y. Nasser, J. Lorandel, J. -C. Prévotet and M. Hélard, "RTL to Transistor Level Power Modeling and Estimation Techniques for FPGA and ASIC: A Survey," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 40, no. 3, pp. 479-493, March 2021, doi: 10.1109/TCAD.2020.3003276.

[7] S. Gupta and F. N. Najm, "Power Macromodeling For High Level Power Estimation," Proceedings of the 34th Design Automation Conference, 1997, pp. 365-370, doi: 10.1109/DAC.1997.597174.

[8] N. R. Potlapally, A. Raghunathan, G. Lakshminarayana, M. S. Hsiao and S. T. Chakradhar, "Accurate power macro-modeling techniques for complex RTL circuits," VLSI Design 2001. Fourteenth International Conference on VLSI Design, 2001, pp. 235-241, doi: 10.1109/ICVD.2001.902666.

[9] S. Gupta and F. N. Najm, "Analytical model for high level power modeling of combinational and sequential circuits," Proceedings IEEE Alessandro Volta Memorial Workshop on Low-Power Design, 1999, pp. 164-172, doi: 10.1109/LPD.1999.750417.

[10] J. Monteiro, S. Devadas and B. Lin, "A Methodology for Efficient Estimation of Switching Activity in Sequential Logic Circuits," 31st Design Automation Conference, 1994, pp. 12-17, doi: 10.1145/196244.196252.

[11] P. H. Schneider and S. Krishnamoorthy, "Effects of correlations on accuracy of power analysis-an experimental study," Proceedings of 1996 International Symposium on Low Power Electronics and Design, 1996, pp. 113-116, doi: 10.1109/LPE.1996.547490.

[12] F. N. Najm, "Transition density: A new measure of activity in digital circuits," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 12, no. 2, pp. 310–323, 1993.

[13] S. Alipour, B. Hidaji and A. S. Pour, "Circuit level, static power, and logic level power analyses," 2010 IEEE International Conference on Electro/Information Technology, 2010, pp. 1-4, doi: 10.1109/EIT.2010.5612180.

[14] C. X. Huang, B. Zhang, A.-C. Deng, and B. Swirski, "The design and implementation of powermill," in In Proceedings ofthe International Symposium on Low Power Design. Citeseer, 1995.

[15] Burch, Najm, Yang and Trick, "McPOWER: a Monte Carlo approach to power estimation," 1992 IEEE/ACM International Conference on Computer-Aided Design, 1992, pp. 90-97, doi: 10.1109/ICCAD.1992.279392.

[16] M. G. Xakellis and F. N. Najm, "Statistical Estimation of the Switching Activity in Digital Circuitsy," 31st Design Automation Conference, 1994, pp. 728-733, doi: 10.1109/DAC.1994.204196.

[17] Y. Park and E. Park, "Statistical power estimation of cmos logic circuits with variable errors," Electronics Letters, vol. 34, no. 11, pp. 1054– 1056, 1998.

[18] https://yaml.org/, Jul. 15, 2021