NTNU

Norwegian University of
Science and Technology

RTL Power Estimation Flow and Its Use in Power Optimization

Sondre Rennan Nesset

Master of Science in Electronics
Submission date: June 2018
Supervisor: Per Gunnar Kjeldsberg, IES
Co-supervisor: Knut Austbø, Nordic Semiconductor

Norwegian University of Science and Technology
Department of Electronic Systems

Problem description

Today's low-power IC products are becoming increasingly ubiquitous, for example in wearables or smart appliances, while at the same time the complexity of these chips makes them harder to handle during the design phase. One very important aspect is a smooth and reliable power estimation flow during the RTL-design phase. This enables effective and qualitative design-power tradeoffs as early as possible in the design cycle without going all the way to a final gate-level-implementation. The first part of this thesis is to utilize a commercial RTL power-estimation-tool and improve its modelling to establish a good correlation between the RTL-estimation and the full gate-level-implementation. The correlation shall be visible over a set of diverse scenarios for an actual design. The second part of this thesis will utilize the improved power-estimation-flow to visualize power-tradeoffs during the RTL-phase when introducing several design-optimization techniques to the design. Some of these techniques and their tradeoffs will be verified by full netlist power-simulations in order to prove acceptable correlation in power-estimation between design-phase and full implementation.

Abstract

The increased complexity and low-power requirements of integrated circuit design demands reliable and accurate power estimations in the RTL phase, for effective design tradeoffs early in the design phase. This thesis develops a methodology to correlate RTL and Netlist power estimations. With a reliable RTL estimation the designer could choose the most power efficient design early in the design phase, leading to a more power efficient IC design. In this work estimations are performed on a set of power scenarios to obtain a power profile of an actual design. The design is synthesized in Synopsys Design Compiler, layout is done in Synopsys IC Compiler, RTL estimation in Spyglass Power and netlist estimation in Primetime PX. For the default design the results yields deviations between RTL and netlist estimation below 5% for all scenarios. Due to inaccuracy in estimation of analog macros and IO PADs, the analog domain is excluded and only the digital domain is considered for exploration.

Several design optimization techniques are implemented in RTL and correlation verified by netlist power estimation. Clock gating is one of the most effective techniques to reduce dynamic power consumption. Both implementation of clock gating cells in the RTL code and automatically interfered clock gates by the synthesis tool is explored for power reduction. By implementing clock gating on a hierarchical level in unused logic, power savings of 82.4 % is achieved for low activity scenarios. The deviation is within 5% with the calibration data extracted from the netlist of the default design. Automatic inserted clock gates by the synthesis tool is explored, for a set of bit width threshold values in RTL estimation. The most interesting results is synthesized and verified by netlist power estimation. The netlist estimation shows a power reduction of 41.6% for low activity scenario and 15% for high activity scenario, by increasing the bit width to eight bits. The correlation is decreased to 20% deviation on total power, this inaccuracy needs to be addressed and is left for future work.

Much of the dynamic power consumption in integrated circuits comes from high number of transitions on high capacitance buses. Bus encoding schemes aims to minimize the number of transitions to reduce the power consumption. In this work T0 encoding is implemented between the CPU and RAM to reduce the number of transitions. This encoding introduce extra logic with power and area overhead. Since the design used in this thesis is quite small with a low capacitance bus, no power savings is achieved, due to the power overhead. This design was synthesized and performed netlist power estimation which gave deviation below 5.7%.

Sub-threshold leakage in CMOS circuits is becoming increasingly important challenge, since it is dominating more of the total power in smaller process technology nodes. The leakage currents are strongly influenced by the transistor threshold voltage, Vt. One way of reducing the leakage current is to optimize the Vt mix. Exploration of Vt mix is only performed in RTL estimation in this thesis, due to limited time. Netlist estimation demands a more complex place and route to verify timing constraints in all process, temperature and voltage corners. In this work only typical corner is explored, investigating correlation on other corners is left for future work.

Extracted calibration data from netlist improves correlation significantly, this thesis show that calibration data from similar designs yields good correlation. This way the methodology can be used by extracting calibration data from a netlist of same technology and similar design to get accurate RTL power estimations without netlist estimations. Achieving reliable power estimation early in the design phase. The methodology describes how accurate RTL power estimations could be achieved by isolating the design to only the digital domain, and thorough debugging of the power numbers against a gate level reference.

Preface

This report is a result of the master thesis in the fifth year of the Master of Science degree program in Electronics, Design of Digital Systems at the Department of Electronics and Telecommunication (IET) at the Norwegian University of Science and Technology (NTNU).

The company named Nordic Semiconductor ASA in Trondheim proposed the project title "RTL Power Estimation Flow and Its Use in Power Optimization". Tools from Synopsys is utilized, and Nordic Semiconductor has provided workplace and computer equipment with access to all the other design tools needed, which I am grateful for. The work has given me a lot of insight in the design flow of integrated circuits, low power design and design tools used in the industry. I would like to thank my supervisors Knut Austbø, Martin Olson and Jan Egil Øye from Nordic Semiconductor and professor Per Gunnar Kjeldsberg from NTNU, for their help and guidance throughout this work.

Trondheim June 14, 2018
Sondre Rennan Nesset

Contents

List of Figures

List of Tables

List of Aconyms

CMOS Complementary Metal Oxide Semiconductor

HDL Hardware Description Language

EDA Electronic Design Automation

RTL Register Transfer Level

SDC Synopsis Design Constraint

UPF Unified Power Format

FSDB Fast signal database

SoC System on Chip

ICGC Integrated Clock Gating Cell

1 Introduction

1.1 Motivation

The progress in chip technology has led to increasingly smarter, complex and power efficient devices. One of the most important parameters of hardware development is lowering the energy consumption. Low power consumption has benefits like increasing the time between battery charging, and new features for IoT, Internet of Things, devices where battery lifetime is crucial.

To minimize the power consumption several RTL, Register Transfer Level, design optimization techniques are implemented by the designers. The designers need reliable power estimations to make tradeoffs between different implementations. Power estimation can be performed at different design phases, as indicated by Figure 1. Estimation during the last step gives highly accurate numbers, since every gate and wires inserted by the synthesis tools are taken into account. Netlist simulations is time consuming and brings the designer back to RTL design of the design process if the power specifications are not met. RTL power estimation is faster and does not demand a netlist, but decreases the accuracy of the estimations. Reliable power estimations during the RTL phase enables effective design-power tradeoffs early in the design cycle. This thesis aims to decrease the gap in accuracy between RTL and Netlist estimations of a SoC, System on Chip, design.

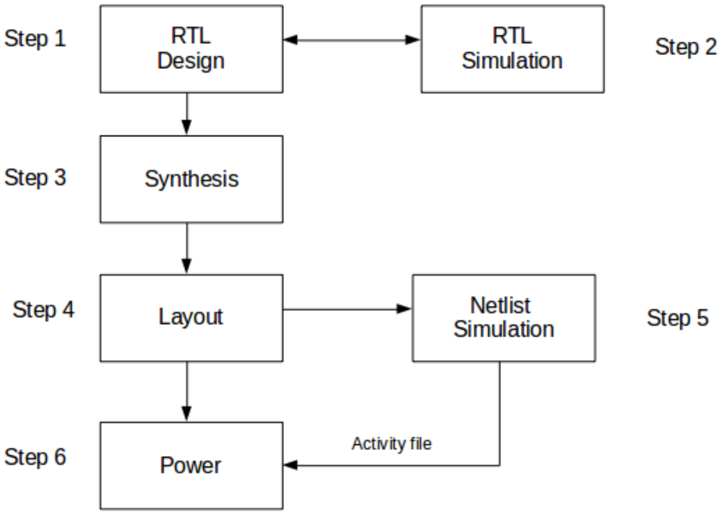

Figure 1: Design phases

1.2 Objectives, limitations and main contributions

This work will explore power estimations in Spyglass Power for RTL and Primetime PX for netlist estimations, on a real SoC design. The goal is to develop a effective methodology for a reliable power estimation flow in the RTL phase, establishing good correlation to the full gate level implementation. The methodology should give good correlation on both total power and on a component level.

Second part of the thesis will be to explore different power optimization techniques. These techniques will be implemented and estimated on the RTL level. The most interesting designs will be synthesized for full netlist estimation to verify and investigate correlation when some design changes is done to the design. This way similar design in the same technology can be used to extract calibration data for high fidelity and accurate power estimations in RTL before a netlist is ready. In this work a relatively small design is explored for fast estimations and synthesis. Only the digital domain is considered for power estimations, due to low accuracy of analog macros. Also only typical process, temperature and voltage corner is explored since gate level implementation on multiple corners is time consuming and left for future work.

The main contributions in this project can be summed up in the following parts:

1. Exploration of correlation between RTL and netlist estimations
2. Development of methodology for reliable RTL power estimations
3. Evaluations of different power optimization techniques

1.3 Thesis structure

This report is divided into the following chapters:

Chapter 2: Theory present useful background theory for the project. CMOS power contributors, how power is estimated and different low power techniques.

Chapter 3: Previous work and tools presents the tools used for estimation and describes the file formats needed to obtain power estimation in modern power analysis tools.

Chapter 4: Methodology explore estimations in the tools and development of the methodology.

Chapter 5: Implementation of low power techniques describes the implemented design techniques and their correlation between netlist and RTL estimation.

Chapter 6: Discussion discuss the results, their limitations and further work that should be performed

Chapter 7: Conclusion summarize the results of this project

2 Theory

This Chapter describes the relevant theory for this thesis. First the CMOS power consumption is described, then some important RTL power reduction techniques, different power estimation techniques and the parameters in the circuitry which effects the estimation.

2.1 CMOS Power consumption

There are two major components to power dissipation in digital CMOS circuits, dynamic and static power consumption [1]. Static power is due to leakage currents, while dynamic consumption is a result of switching activity in the circuits. Total power consumption is given by Equation 1.

$$\begin{aligned} P_{avg} &= P_{switching} + P_{short-circuit} + P_{leakage} \\ &= (C_L \cdot V_{DD}^2 \cdot f_{clk} \cdot \alpha) + (V_{DD} \cdot t_{sc} \cdot I_{peak} \cdot f_{clk}) + (I_{leakage} \cdot V_{DD}) \end{aligned} \tag{1}$$

$$\begin{aligned} &\texttt{Where:} \\ &\qquad V_{DD} = \texttt{supply voltage} \\ &\qquad f_{clk} = \texttt{clock frequency} \\ &\qquad \alpha = \texttt{activity factor} \\ &\qquad C_L = \texttt{external load capacitance} \\ &\qquad t_{sc} = \texttt{time duration of the short circuit current} \\ &\qquad I_{peak} = \texttt{total internal switching current} \\ &\qquad I_{leakage} = \texttt{leakage current} \end{aligned} \tag{2}$$

2.1.1 Static power

The leakage current can arise from four main sources [28]:

- Sub-threshold leakage: a transistor in the weak inversion region will leak current from the drain to the source.
- Gate leakage: leakage currents from the gate through the oxide to the substrate due to gate oxide tunneling and hot carrier injection.
- Gate induced drain leakage: the current from drain to the substrate induced by a high field effect in the MOSFET drain.
- Reverse bias junction leakage: electron/hole pairs in the depletion regions causes carrier drift.

The substrate injection and subthreshold effects, is primarily determined by fabrication technology considerations. This leakage can be explained with a simple model that describes the parasitic diodes of a CMOS inverter [6], as shown in Figure 2

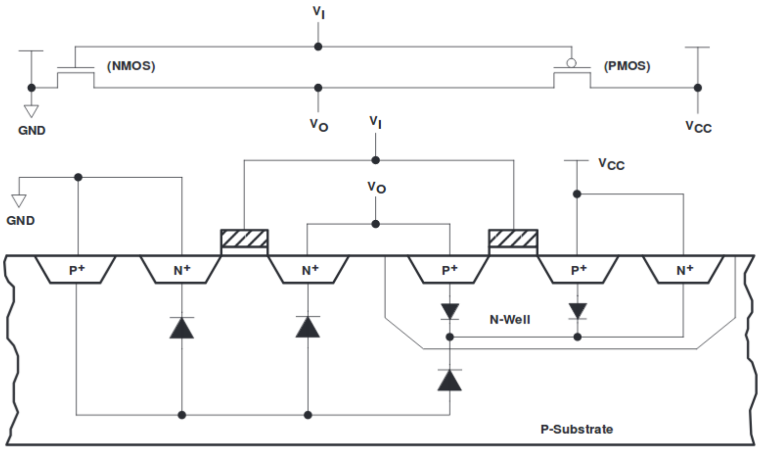

Figure 2: Model describing parasitic diodes present in CMOS inverter [6]

The source drain diffusion and N-well diffusion form parasitic diodes. Because parasitic diodes are reversed biased, their leakage currents contribute to static power consumption. The leakage current is described by the following equation [6]

$$I_{leakage} = i_s(e^{qV/kT} - 1) \tag{3}$$

Where:

$i_s =$ reverse saturation current
$V =$ diode voltage
$k =$ Boltzmann's constant
$q =$ electric charge
$T =$ temperature [K] (4)

2.1.2 Dynamic power

The primary source of dynamic power consumption is switching power, caused by the power required to charge and discharge the output capacitance. Figure 3 illustrates the switching power in a CMOS transistor.

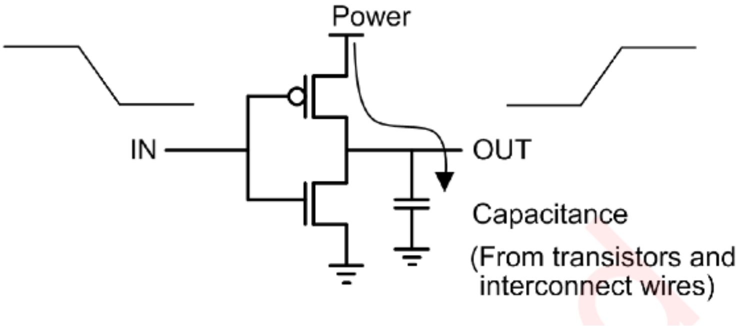

Figure 3: Dynamic power [28]

Equation 1 shows the switching component of power.

$$P_{switching} = C_L \cdot V_{DD}^2 \cdot f_{clk} \cdot \alpha \tag{5}$$

Where:

$P_{switching} =$ capacitive-load power consumption
$V_{DD} =$ supply voltage
$f_{clk} =$ output signal frequency
$\alpha =$ activity factor
$C_L =$ external load capacitance

Hence the switching power is dependent on the activity of the circuit, frequency, supply voltage and load capacitance from the transistors and interconnect wires.

In addition the short circuit power contributes to dynamic power due to the current that flows when the transistors are switching from one logic state to the other. This is a result of the current that flows from the supply voltage to ground when the p-channel transistor and the n-channel transistor is turned on at the same time during the logic transition [6]. The switching frequency, the rise and fall times and the internal nodes of the device will have an effect on the switching current. The short circuit power consumption can be calculated by equation 6

$$P_{short-circuit} = V_{DD} \cdot t_{sc} \cdot I_{peak} \cdot f_{clk} \tag{6}$$

Where:

$P_{short-circuit} =$ Short circuit power consumption
$V_{DD} =$ supply voltage
$t_{sc} =$ time duration of the short circuit current
$I_{peak} =$ total internal switching current
$f_{clk} =$ input signal frequency

2.2 Clock distribution and power gating

Decreasing the voltage is the most effective way of reducing power consumption, since dynamic power is proportional to Vdd squared. When reducing the supply voltage, the delay of signals propagating through gates increases, leading to a slower performance. Equation 7 is a approximation of the drive current of a MOSFET transistor.

$$I_{DS} = \mu \cdot C_{OX} \cdot \frac{W}{L} \cdot \frac{(V_{GS} - V_T)^2}{2} \tag{7}$$

Where:

$I_{DS} =$ drive current
$V_{GS} =$ gate-source voltage
$\mu =$ carrier mobility (8)
$C_{OX} =$ gate capacitance
$V_T =$ threshold voltage
$W, L =$ dimension of transistor

By lowering the supply voltage, hence V_{GS}, to decrease the dynamic power the threshold voltage needs to be decreased to maintain performance. Thus there is a conflict due to increased leakage current [2]. This problem could be solved by using multiple Vt cells, using higher Vt for high performance parts of the circuit. In multivoltage designs different parts of the circuit might require different frequencies and lowering the V_{DD} on selected blocks helps to reduce power. Figure 4 shows a simple multivoltage power domain design.

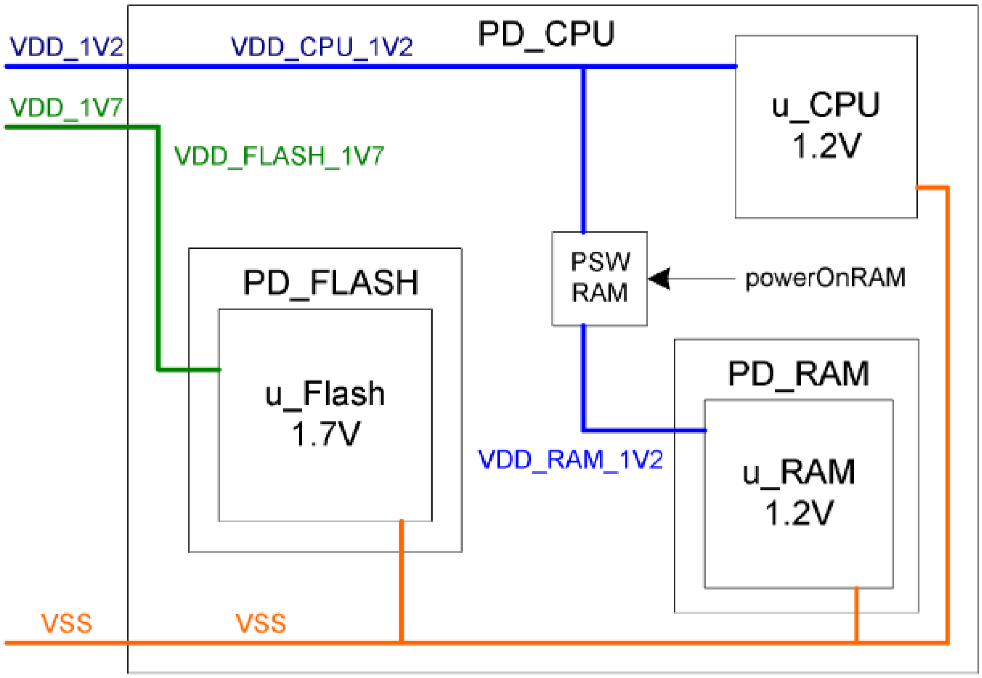

Figure 4: Power domain

In this Figure the circuit consist of multiple supply rails where the flash requires 1.7 V and the CPU and RAM 1.2 V. The circuit also consist of a power switch to power gate the RAM when it is not in use. Power gating is a method to completely turn off the power supply, to avoid leakage power for sub circuitry in idle mode. Figure 5 shows the schematic of a multi-threshold power gated CMOS circuit.

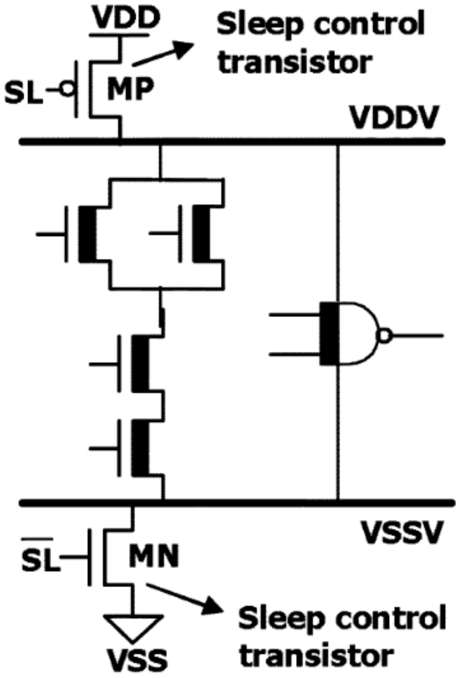

Figure 5: Power gating [10]

The high-threshold voltage sleep control transistors have low leakage and introduce power gating to the circuit. The low-threshold transistors in the circuit ensures high performance.

The clock distribution network synchronize the clock to ensure that sequential elements receive the rising clock edge at the same time. To achieve this it is common to insert buffers in the clock path. Figure 6 shows a clock distribution network where the clock source is distributed to multiple buffers and registers.

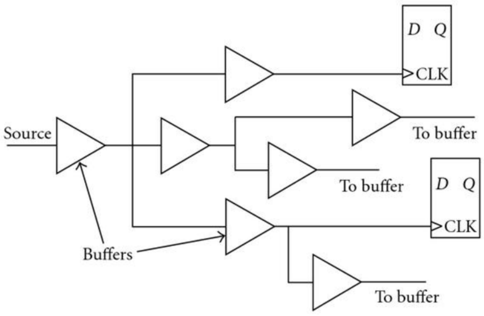

Figure 6: Clock tree [8]

2.3 Clock gating

The major source of power consumption in digital design is the clock tree which may consume as much as 45% of the system power[22]. Reduction of the dynamic clock tree power can lead to a considerable reduction in overall power. Since the clock buffers have the highest toggle rate in the system, and they tend to have high drive strength to minimize clock delay. Clock gating can be inserted manually by the designer in RTL or automatically by the synthesis tool. Clock gating is applied to register banks where a group of flip-flops share the same clock

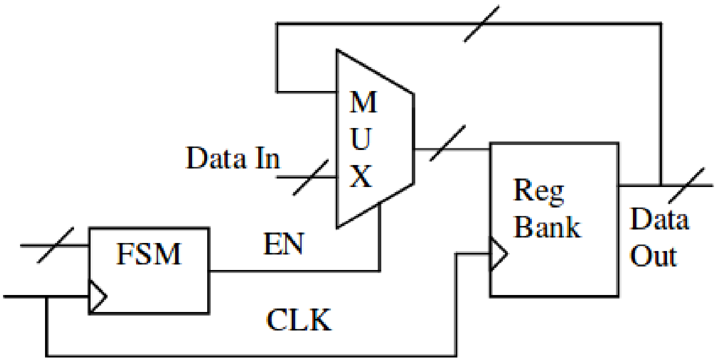

Figure 7: MUX register [22]

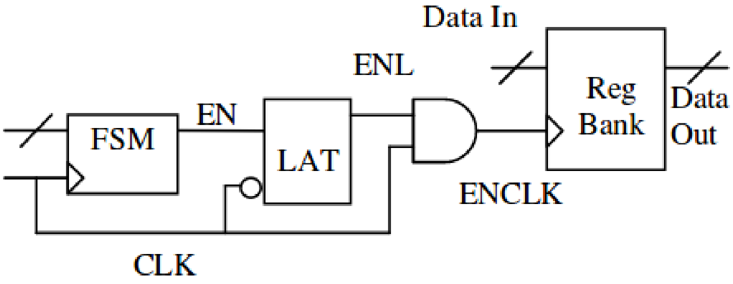

Figure 8: Latch based register [22]

and synchronous control signals. With no CG the synthesis tool in general implement register banks by using a feedback loop and a multiplexer shown in Figure 7. A clock gate eliminates the unnecessary activity associated with reloading the registers and eliminates the MUX and feedback net by inserting a Latch and a AND gate shown in Figure 8. The latch ensure that no glitch on the enable signal from propagating to the register clock pin. This clock gating style is called icgc, integrated clock gating cell.

Applying CG for registers with equivalent control signals using one CG element is known as Register-Based CG. A minimum bit width of 3 can overcome the power savings achieved [4]. Using the Register-Based CG, clock gating can be applied at different hierarchical levels, both module, enhanced and multistage hierarchical levels.

2.4 Bus encoding

A large part of the total power consumption in a circuit is due to the transitions on the buses. This is caused by the large capacitance and switching activity, as described is Equation 5 effects the dynamic power. The literature provides many encoding schemes to achieve reduction in the bus transition activity. For the instruction bus Gray code[16] and T0 encoding [11] are the most common encoding techniques. Since addresses are mostly sequential the Gray code ensures that only one bit is changed between two consecutive sequential data words. The T0 encoding adds an extra bit line along the address bus, where the bit is set if the consecutive words are sequential, in which case the address is not put on the bus. For data buses the data are typically not sequential and other encoding schemes are used. Bus-Invert coding [16] computes the number of transitions on the bus compared to the previous data, if the transition count is over half of the bit width, the data is inverted and put on the bus. A extra bit line is added to signal the inversion. Variants of these encoding schemes can be combined to minimize the number of transitions for both address, data and multiplex buses. All the schemes introduces overhead in power and area, therefore the capacitance of the bus needs to be high enough to achieve power savings from bus encoding.

2.5 Corners

The scaling of IC technology are significantly impacted by process variations, both inter-die and intra-die variations [24]. The inter-die variations model the average variations across the die, while intra-die model the individual local variations within the same die. These variations represent the extreme values and a circuit may run slower or faster then specified at higher or lower temperatures and voltages then the typical corner [18] and need to be considered during development. These variations are modelled in the liberty technology files.

2.6 Vt mix

One way of reducing the leakage current is to use libraries with multiple Vt cells. The subthreshold leakage depends exponentially on Vt, while the delay has a much weaker dependence on Vt [2]. Many libraries comes wih Low Vt, Standard Vt, and High Vt. Figure 9 shows the relation between leakage power and delay of these cells for a 90nm Library.

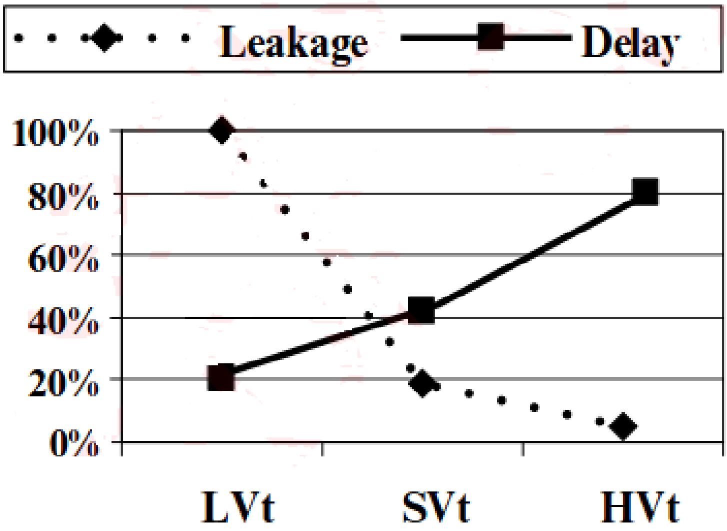

Figure 9: Leakage vs Delay for a 90nm Library [2]

In synthesis the target is to minimize the number of high leakage Vt transistors by using them only when required to meet timing. This can be done by performing a initial synthesis targeting a primary library, then optimize by targeting additional libraries with different thresholds. To meet the minimum performance it is practice to syntheses with high leakage, high performance library first, then replace cells not in the critical path with lower leakage cells.

2.7 Parasitic

A parasitic element is a circuit element with unwanted attributes that alters the performance of a circuit. This could be resistance, inductance or capacitance. Line resistance is caused by non ideal wires, and is determined by the length, area and conductivity of the material. Line capacitance is caused by interconnects separated from the semiconductor substrate by an insulating layer. These effects the dynamic power consumption of the circuit, as the switching power is dependent on the load capacitance. In Netlist estimation these effects are modelled by the layout and could be estimated accurately, while in RTL estimation these parameters are not present. One method is to create wireload tables based on statistics from layout as calibration data for RTL estimations.

2.8 Input transition time

The DC current from a CMOS circuit can be modeled as in Figure 10, where the peak current occurs between 0V and VDD. This current peaks every time the input voltage makes a logical transition, and the power dissipation is dependent on how fast this transition is executed [30].

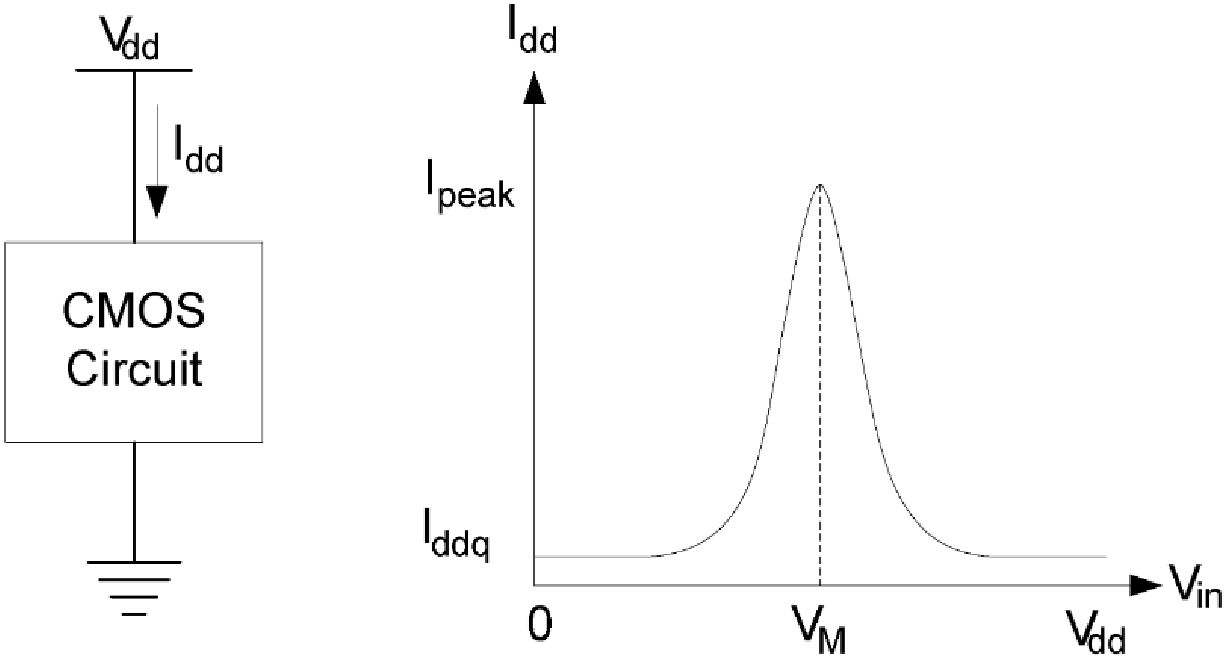

Figure 10: DC current CMOS circuit

The transition time have an impact of the short circuit power consumption in the design, as described in Equation 6.

2.9 Fanout

The fanout is determined by the number of gates driven by a gate. The input capacitance of each gate will effect the total load capacitance. This is illustrated in Figure 11 where the load capacitance, C_L seen by the first inverter is dependant on the input capacitance of the fanout gate[30]. The load capacitance can be expressed as the sum of input capacitance seen by the gate.

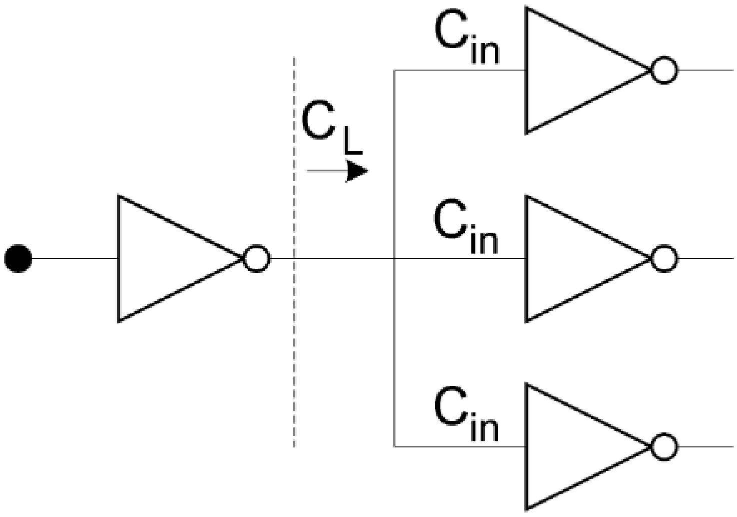

Figure 11: Load capacitance due to fanout

The fanout of each gate have an impact on the dynamic power consumption, given by Equation 5. As the fanout capacitance effects the power consumption required to charge and discharge the transistors when switching activity occurs. The input capacitance of each fanout gate is dependent on the technology parameters and the size of the transistor [7].

2.10 RTL power estimation flow

Figure 12 shows a typical estimation flow for RTL level.

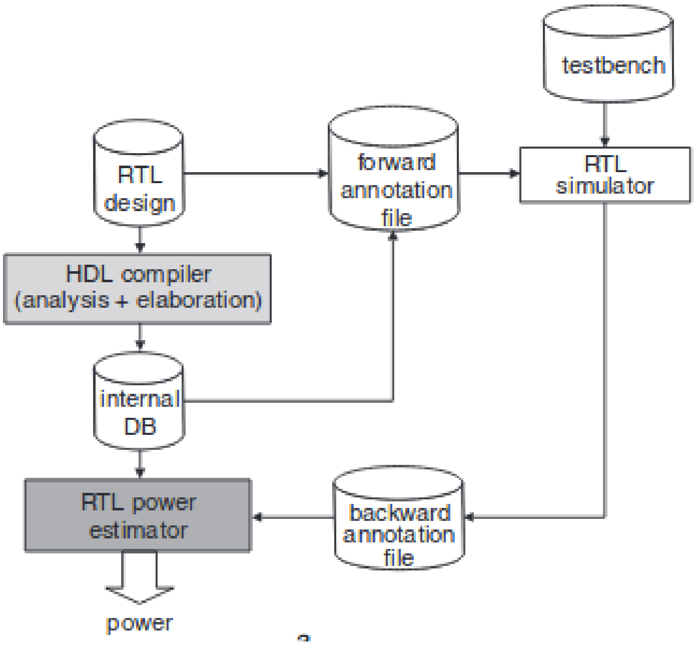

Figure 12: Power estimation flow [14]

After analysis and elaboration by the HDL compiler, the design is translated into a technology-independent format containing RTL modules (macros), gates, memory elements, and MUXes.

The forward annotation file contains the list of nets to be monitored during RTL simulation. The RTL simulator produces a backward annotation file, consisting of all the nets specified in the forward file, annotated with switching activity and static probability values. An RTL power estimator takes the internal database produced in the first step and this activity information, and calculates the power estimate.

2.10.1 Analytical methods

Analytical methods attempt to relate power consumption of RTL descriptions to fundamental quantities that describe the physical capacitance and activity of a design. These techniques can be divided into complexity- and activity-based models. The complexity based model relies on the fact that the complexity of a chip architecture can be described roughly in terms of gate equivalents. The gate equivalent specifies to number of reference gates required to implement a function, specified in a library database or by the user. The power can be estimated by multiplying the approximate number of gates by the average power consumed by each gate. An example is given in equation 9 by the Chip Estimation System [9]

$$P = \sum_{i\epsilon(fns)} GE_i(E_{typ} + C_L^i V_{dd}^2) f \cdot A_{int}^i \tag{9}$$

where GE_i is the gate equivalent count for block i, E_{typ} is the average power consumed by a gate, C_L is the average capacitance load including fan-out and wiring, f is clock frequency, and A_{int} is the percentage of gates switching each clock for this block. One disadvantage is that the energy consumption is based on a single reference gate. Liu and Svensson improved the technique by applying custom estimation for different design entities like logic, memory, interconnect and clock [13]

These complexity-based methods require little information, the disadvantage is that they do not model circuit activity accurately. A fixed activity factor is typically provided by the user. This may give a good estimate of total chip power, but the power between modules is likely to be incorrect, making it difficult to perform meaningful architectural trade-offs.

Activity-based models focus on using the concept of entropy as a measure of the average activity in a circuit. Najm [17] observes that power is proportional to the product of capacitance and activity, with area as a measure of capacitance and entropy as a measure of activity.

$$P \propto Capacitance \cdot Activity \propto Area \cdot Entropy \tag{10}$$

Running RTL simulations of the design to measure the input and output entropies of the functional blocks and using the equations as a measurement to predict the average power. No timing information is provided in the calculations, therefore glitching power is not accounted for. Capacitance is assumed uniformly distributed over all nodes.

2.10.2 Empirical methods

This method tries to relate power consumption based on measurements of existing implementations and provide a macromodel. Macromodelling can be divided into fixed-activity and activity-sensitive models. One proposal for a fixed-activity macromodelling strategy is the

Power Factor Approximation (PFA) method [23]. Equation 11 approximate the power consumed by a given architecture

$$P = \sum_{\text{i } \epsilon \text{ (all blocks)}} \kappa_i G_i f_i \tag{11}$$

where each block is characterized by PFA proportionality constant κ_i, a measure of hardware complexity G_i and an activity frequency f_i. This method can be viewed as a general technique for an entire library of RTL level functional blocks. The drawback is that the model does not account for the influence data activity have on power consumption.

Activity sensitive models try to account in some way for the influence that data activity have on power. On power analysis tool called SPA [12, 20, 21] is developed and based on the concept of activity profiling. Prior to the power analysis a RTL simulation of the design is carried out. The design entities and signals in the data and control paths are monitored and recorded. These statistics are used in power models that account for activity and complexity.

3 Previous work and tools

This Chapter describes the tools used in RTL and Netlist power analysis, and the file formats used by this tools for power modelling. Spyglass Power and Primetime PX from Synopsys was chosen for the power analysis. Spyglass support a integrated solution which covers all aspects of power analysis. Supports SystemVerilog, works at both RTL and gate level, and supports activity data from the file formats FSDB, VCD and SAIF. Questasim from Mentor with the plugin Codelink was used for RTL simulation, debugging and dumping of activity data. Previous work and the estimation flow recommended by Synopsys is also described in this Chapter.

3.1 TC1-mini

The chosen design is TC1, Test Chip 1, from Nordic Semiconductor. TC1 is a relatively small design used to test different MCU architectures. The chip consist of three MCUs, with five different RISCV processors. This work will focus most on MCU0, which consist of a RISCV core, TCM memory, a cryptographic accelerator and a bus matrix for the memory mapping. Figure 13 shows the architecture of TC1.

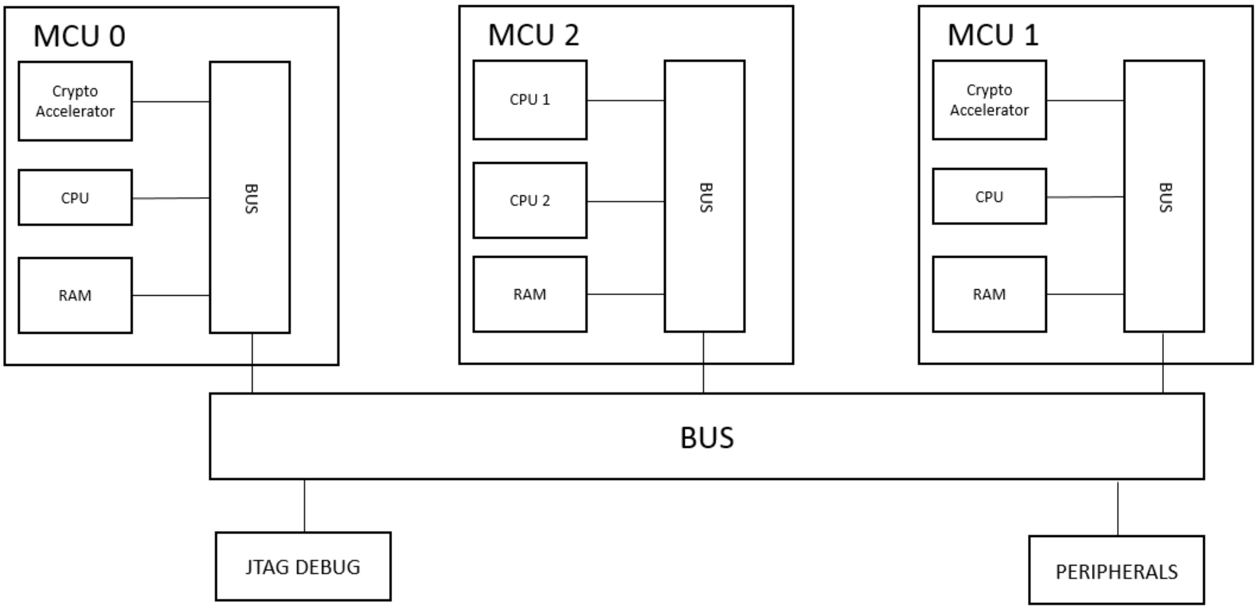

Figure 13: TC1 architecture

3.2 Questasim

Questasim by Mentor is a tool for simulation, verification and debugging of RTL code. In this work Questasim is used for simulation of different power scenarios, to verify correct behavior and dumping the activity data for power estimation.

3.3 Synthesis in Synopsys Design Compiler

The Design Compiler software synthesize a block-level RTL design to generate a gate-level Netlist. This includes reading the RTL design, loading libraries, technology data and floorplan constraints. The tool generates output data which is required by physical design and layout tools.

3.4 Layout in Synopsys IC Compiler

The IC Compiler place and route system is a chip-level physical implementation tool. It includes flat and hierarchical design planning, placement, clock tree synthesis, routing and optimization.

3.5 Estimating power in Spyglass

Figure 14 shows an overview of the information needed to estimate power in Spyglass.

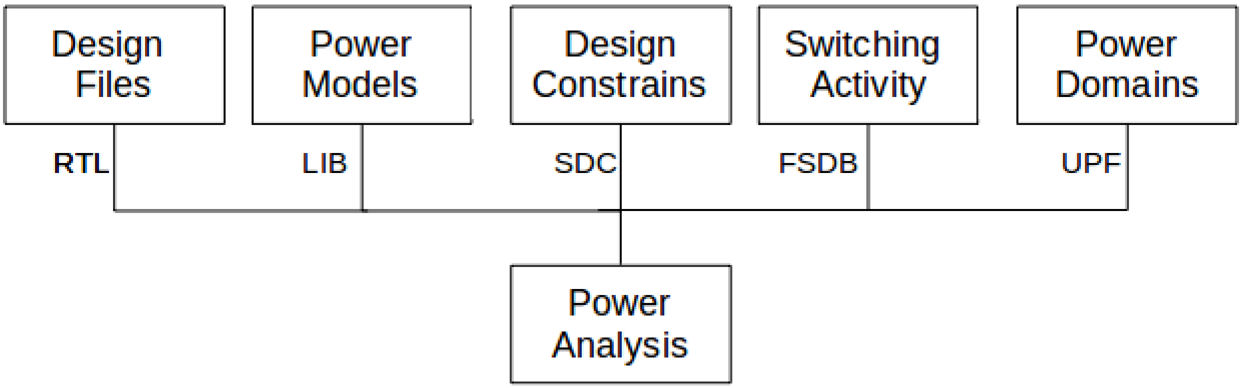

Figure 14: Estimation flow Spyglass

The design files is the RTL code which describes the design. The tool analyzes the RTL code and translate it to gate-level information, for power analysis.

A power model is needed to estimate leakage and internal power dissipated for each type of cell, this is provided by the power models in the lib files. To model switching activity several file formats are available, FSDB (Fast Signal Database), VCD (Value Changed Dump), SAIF(Switching Activity Interchange Format) which is described in Section 3.7.2.

The activity files are dumped from RTL simulation in Questasim. Based on the RTL code and firmware of the given scenario, the FSDB dumper outputs the activity files from RTL. Which logs input and output of modules and registers, wires and signals specified in the code.

The SDC files sets parameters that affects the power. This could be definition of the clocks in the design, input transition times, which affect the internal current consumption. Also the output capacitance load, needs to be set for every external output as it affects the switching power, given by Equation 1.

The UPF files describes the power intent in the design. The file format consist of standard syntax for describing power supplies, power switches, level shifters, isolation, memory retention and power states.

3.6 Calibration

Spyglass comes with a feature to extract calibration data from the reference synthesis and back end characteristics [28]. The parameters generated by the calibration is:

- Cell sizing
- Vt-mix
- Clock tree
- Capacitance model

It generates files which shows the percentage of cell allocation with drive strength in the design. These parameters effects combinatorial and sequential leakage and dynamic power. The Vt-mix gives an percentage of different vt cells used in the netlist, which impacts combinatorial and sequential leakage. The clock buffer information which impacts the clock power and sequential dynamic. The capacitance model based on the design and SPEF file impact all switching power.

This calibration provides extra information to the RTL estimation and gives more accurate estimation dependent on the technology used and the design.

3.7 Estimating power in Primetime

The Primetime PX tool estimates power on the Netlist level and can therefore extract more information then RTL estimation. With a synthesized Netlist and a layout, the tool will have a complete overview and more data to perform estimation. The required data for performing power analysis in Primetime is:

- 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, or Synopsys database format, containing leaf-level instantiating 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 capacitance for all the nets.

As the list shows Primetime require most of the same files as Spyglass, but have more accurate information since it uses a gate-level netlist instead of RTL code, and also uses the whole SPEF file. Where Spyglass uses calibration data with percentages.

3.7.1 Power modeling

Liberty technology libraries (LIB-files) contain information about characteristics and functions of the components in an ASIC library. Among many other attributes, the text file contains information regarding each cells area, timing and function [25]. This file is provided by the technology vendor and is unique for each technology. For power estimation the library files contains information about leakage and internal power, look-up tables for different process, temperature and voltage corners are included for each cell in this technology.

The vendors does measurements of the leakage currents at multiple sets of inputs to generate a leakage model, which can be used as a look-up table to estimate static power consumption [26]. Listing 1 shows a look-up table for a two input NAND gate. The supply voltage is set, and two different inputs provide different leakage value described at line 8 and 13.

Listing 1: Leakage

```
1  library(leakage_power_example) {
2      leakage_power_unit : "1pW";
3      cell (NAND2) {
4      cell_leakage_power : 1.0 ;
5      leakage_power() {
6          power_level : "VDD1";
7          when : "!A1 !A2" ;
8          value : 1.5 ;
9      }
10     leakage_power() {
11         power_level : "VDD1";
12         when : "!A1 A2" ;
13         value : 2.0 ;
14     }
15 }
```

The short circuit power is modeled by the input voltage transition time and load capacitance. These values are represented in a two dimensional table in the LIB files. Listing 2 shows an example of internal power look-up table for an AND gate, where the power is a 3x3 dimensional table dependent on the transition time and capacitance.

Listing 2: Internal power

```
library(internal_power_example) {
    power_lut_template(output_by_cap_and_trans) {
        variable_1 : total_output_net_capacitance ;
        variable_2 : input_transition_time ;
        index_1 ("0.0, 5.0, 20.0");
        index_2 ("0.0, 1.0, 20.0");
    }

    cell(AND2) {
        pin(Z) {
        direction : output ;
        internal_power {
            power(output_by_cap_and_trans) {
                values ("2.2, 3.7, 4.3", "1.7, 2.1, 3.5", "1.0, 1.5, 2.8");
            }
            related_pin : "A B" ;
            }
        }
        pin(A) {
            direction : input ;
        }
        pin(B) {
            direction : input ;

        }
    }
}
```

The switching power is not modeled directly into the LIB files, the toggle information is dependent on the firmware running on the SoC and extracted from a simulation testbench. The

LIB files includes information about capacitance and supply voltage, which is used by the tool to calculate the switching power.

3.7.2 Switching activity

The switching activity of a design refers to how often different nets changes the signal level. This information is dumped from an RTL or Netlist simulation tool, and used to estimate the dynamic power consumption of the circuit. Different file formats are listed below.

Value change dump

VCD is an event-based format that logs every value change made by each signal, and the time at which the change occurred.

Switching Activity Interface Format

The Switching Activity Interface Format (SAIF) file logs the average activity of each signal in a simulation.

Fast Signal Database

The Fast Signal Database (FSDB) is an event-based format, similarly to the VCD, which logs each toggle in every signal. Its representation is binary, while VCD is ASCII making a more compact representation and smaller file sizes.

Synopsys Design Constraint

The Synopsys Design Constraint (SDC) files is used to specify constraints regarding power, timing and area of the design. This includes input transition times, fanout, load capacitance, clock definition. These constraints is used to estimate power, since it tells the synthesis tool how the RTL design is synthesized.

UPF

UPF is a standard set of Tcl-like commands used to specify the power intent of a design. Using the UPF commands, you can specify the supply network, switches, isolation, retention, and other aspects pertinent to the power management of your design. This single set of commands can be used for verification, analysis, and implementation of your design [29] A UPF file example is shown below. The code describes the multi-voltage design in Figure 4, with two voltage supplies, one power switch and three power domains. The main power domain is PD_{CPU} which is a always on domain. The power switch controls the the power gating of the PD_{RAM} domain, with the same power supply as the main power domain. The flash includes a different power domain, and a always on domain at a different voltage level.

```
    create_supply_port VDD_1V2 -direction in
    create_supply_port VDD_1V7 -direction in
    create_supply_port VSS -direction in

    create_power_domain PD_CPU -include_scope
    create_supply_net VDD_CPU_1V2 -domain PD_CPU -resolve parallel
    create_supply_net VDD_FLASH_1V7 -domain PD_CPU -resolve parallel
    create_supply_net VDD_RAM_1V2 -domain PD_CPU -resolve parallel
    create_supply_net VSS -domain PD_CPU
    create_supply_net VDD_CPU_1V2 -ports VDD_1V2
    create_supply_net VDD_FLASH_1V7 -ports VDD_1V7
    create_supply_net VSS -ports VSS
    create_power_switch PSW_RAM \
        -domain PD_CPU \
        -output_supply_port {Vout VDD_RAM_1V2} \
        -input_supply_port {Vin VDD_CPU_1V2} \
        -control_port {CTRL u_TopLevel/powerOnRAM} \
        -on_state {ON Vin {CTRL}}
    set_domain_supply_net PD_CPU -primary_power_net VDD_CPU_1V2 -primary_ground_net VSS

    create_power_domain PD_FLASH -elements u_TopLevel/u_Flash
    create_supply_net VDD_FLASH_1V7 -domain PD_FLASH -reuse -resolve parallel
    create_supply_net VSS -domain PD_FLASH -reuse
    set_domain_supply_net PD_FLASH -primary_power_net VDD_FLASH_1V7 -primary_ground_net VSS

    create_power_domain PD_RAM -elements u_TopLevel/u_RAM
    create_supply_net VDD_RAM_1V2 -domain PD_RAM -reuse -resolve parallel
    create_supply_net VSS -domain PD_RAM -reuse
    set_domain_supply_net PD_RAM -primary_power_net VDD_RAM_1V2 -primary_ground_net VSS
```

3.8 Power scenarios and power cases

To obtain a power profile of the design three different scenarios are used for all power estimations. These scenarios contain different firmware to utilize different sub circuitry and both high and low activity in the design. Each scenario contain a start up sequence, including starting the clocks, setting some registers before the scenario starts and some printing in the end of the scenario.

Different power scenarios are often used to cover multiple use cases and make a power profile of a design. In multivoltage designs large parts of the chip may be powered down, dark silicon, analyzing power in different scenarios covers the consumption in different power states. Different sections and sub-circuits of the design will be clock gated in different scenarios, and contribute to different dynamic consumption determent by the given scenario. By estimating the power consumption for a set of scenarios which together cover each possible power state, the power profile of the circuit is obtained.

System Idle
This is a low activity scenario where most of the circuit is inactive. The power consumption should be low, with dissipation only from leakage and dynamic consumption from always on circuits in the design.

Core Mark The Coremark is a industry standard benchmark test to measure the performance of the CPU. This high activity scenario will give estimations on high CPU activity.

AES The AES scenario utilizes the on-chip cryptography accelerator performing an AES encryption and comparing the result against the NIST standard, to verify correct behavior. The CPU will have low activity, and the main dynamic consumption comes from the cryptographic accelerator.

3.9 Previous work

In the project work [19] a big design with analog macros and multivoltage was investigated. The correlation was inaccurate, with large deviations on analog macros. No calibration in Spyglass was performed in that work which gave an accuracy of 78% between lab measurements and RTL estimation in Spyglass. Due to limited information on the RTL level estimations, estimations in Spyglass will not be as accurate as Primetime. This includes lack of information regarding interconnect capacitance between gates, a simplified wireload model and a fast and simple clock tree synthesis. The calibration performed in Spyglass will provide more information on these parameters related to technology and that specific design, but this is statistics for the whole design. Not on gate level, which the Netlist estimation will have based on the synthesis and layout files. The estimation will be tool specific and the Spyglass Power Estimation and Power Reduce Methodology [28] present the basis for power correlation against a Netlist reference. Figure 15 describes the high level flow of correlation recommended by Synopsys.

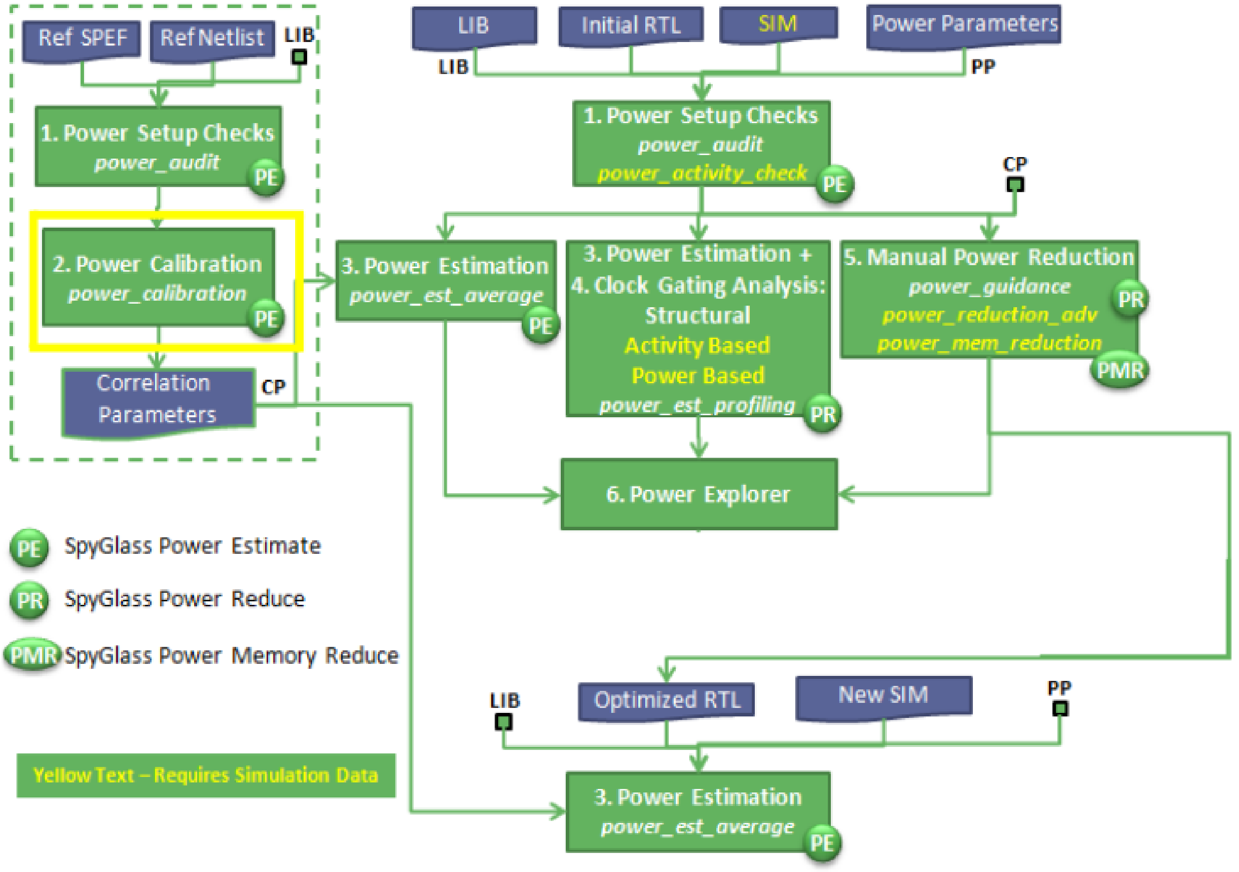

Figure 15: Spyglass correlation flow [5]

This figure shows the setup required for average power estimation and calibration in Spyglass. Furthermore the Spyglass Power Estimation and Power Reduce Methodology provides information regarding analyzing multiple power components and debugging steps of setup and input data. Many sources of correlation deviation are due to the setup of Spyglass and the reference power tool. The different input data, files, models and setup controls need to match RTL and Netlist for efficient correlation. The methodology provide a guide to quickly find the root cause of mismatch between setup and input data files. In general the accepted divergence between RTL estimation and Netlist reference is 15% [28], also the fidelity between different design is crucial for RTL estimation. Synopsys provides a power correlation view in the Spyglass GUI the help determining which components are divergent and requires further analysis and action. The methodology suggested by Synopsys is to use component deviation over total power, find the most dominant component and perform action as described in the High-level Guides for Root Cause Analysis, presented below.

3.9.1 High-Level Guides for Root Cause Analysis [27]

Synopsys provides a High level guide for determining the root cause of power deviations. This guide is used as a starting point for correlation between RTL and Netlist.

Sequential leakage root analysis:
Compare the number of registers and latches with reference. Ensure that the design and the reference use the same type of flip-flops, check SDC if the design uses scan flops. In a multi-threshold voltage design, consider changing the vt mix extracted by calibration. Detailed debugging on the hierarchies with largest deviations, locate set of registers at the root cause, and identify mismatch. (Vt, library, corner)

Combinatorial Leakage:
Compare the area and number of instances in the RTL and reference run. Debug on hierarchical level to find largest mismatch. Analyze cell type, vt, library and library corner. Debug Spyglass and Primetime optimization settings.

Dynamic Power:
Ensure that activity file annotation is as close to 100 % as possible. To improve annotation, find which signals is not annotated and change the simulation settings. Ensure that the reference and rtl estimation are given the same vectors with the same time window on the scenarios. Inspect fsdb file, time based, activity graph. Ensure that RTL simulation does not have a lot of Xs, especially clock, reset, enables.

Internal power:
Change the slew parameter to the average pin slew of the cells in the design. Ensure that clock gating threshold is similar.

Switching power:
Check the wireload models from calibration SPEF files.

Clock power:
Adjust clock tree from calibration. Ensure same frequency of the clock signal, ensure that sequential power matches, otherwise fix sequential. Adjust capacitance model from calibration.

Memory power:
Start with leakage, this should be almost identical check libraries and instances provided. If dynamic are divergent, check vectors and their annotation for the memories. Ensure that cycle-based propagation on memory ports. Check memory access rates.

This guide by Synopsys is a good starting point to verify and correct the setup, input data and files between the RTL estimation tool and the Netlist reference. This is mostly limited to the setup of the tools, to further improve correlation and verify fidelity and a methodology that can be trusted on different designs, without a reference a more specific and detailed methodology is required.

4 Methodology

This work aims to improve the correlation of RTL and Netlist estimation and implement a methodology which will give good correlation on other designs as well. The purpose of RTL estimation is to get fast estimation with high fidelity and good accuracy. This requires calibration data from a netlist and thorough debugging against a reference estimation on the netlist level. In a design process the designer might need power numbers early in the process and need accurate RTL estimation before the netlist is finished. This work will develop a methodology that ensures good correlation, which can be used on same technology and different designs. This is done by first following the default correlation flow by Synopsys described in Section 3.9, then do multiple iterations and debugging for good correlation to make a more specific methodology. Furthermore different low power techniques are implemented in the design, then performed synthesis, layout and estimation in both RTL and netlist. This will give different designs to verify and improve the methodology and investigate how accurate the calibration data is. The goal of the methodology is to make a fast and simple way of correlating a design against a reference, and use these correlation data as a reference for similar design in the same technology. Resulting in a general methodology for RTL estimation which could be used in other designs as well. The work will consist of:

- Create synthesis and layout for default TC1 design
- Establish reference in Primetime PX
- Estimate power in Spyglass Power with Synopsys standard flow
- Improve correlation between RTL and Netlist
- Develop better methodology for correlation
- Implement low power RTL techniques
- Synthesis, layout, Primetime, Spyglass estimations of these designs
- Verify correlation for these design

4.1 Assumptions

For the power estimation some assumptions has to be taken in to account. Temperature and voltage has an impact on both performance and power consumption for a CMOS circuit, as described in Section 2.5. These parameters effect both the dynamic consumption based on the activity files and the leakage power given by the library files for the cells. In this work a typical corner is chosen to model the circuit. With a temperature of 25 degrees Celsius and 1 V supply. Since a chip should meet the specification in a given temperature range, different voltage supplies and process variations power estimation on different corners should be investigated. Due to limited time this is left for feature work and discussed in the discussion section.

4.2 Approach

The first part of the project is to simulate the different scenarios described in Section 3.8 and generate switching activity data for both gate level and RTL level. Then create a reference power estimation in Primetime PX. To ensure that the tools estimates in the right window a time

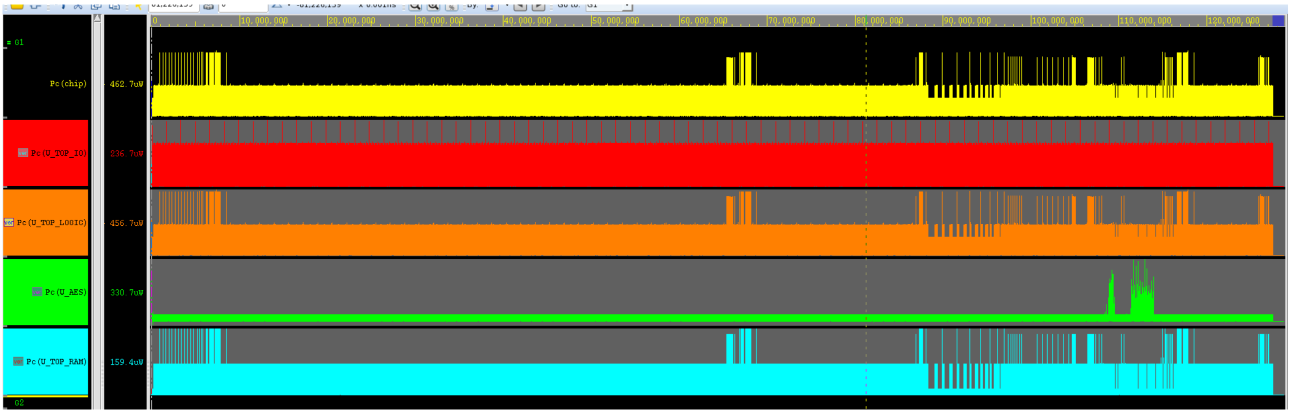

Figure 16: Time based analysis, AES scenario

based power analysis is performed in Primetime to get an overview of the power consumption with regards to time. This could have been done by checking registers and debugging in a simulation tool. The time based analysis is chosen since it will give a good overview of the power consumption over time in each scenario. This feature is also used to debug and verify behavior and power consumption within the scenarios to analyze power consumption on different modules, regarding clock gating, activity and leakage power. Figure 16 shows the time based analysis from the AES scenario.

As the time based analysis show there is an initialization sequence in the start, and some spikes and drops in power depending power consumption in RAM resulted from the number of memory accesses. The AES cryptography operations is performed in the time window between 105 000 ns and 115 000 ns, and is chosen as the time window for this scenario. The time window of the other scenarios were also chosen based on the time based analysis in Primetime. The RTL estimation in Spyglass Power does not support time based power analysis. In the further power estimation, average power is considered in the chosen time window for both RTL and Netlist analysis.

As described in Section 3.5 the switching activity from each scenario needs to be extracted from a simulation tool. In this work the default tool flow from Nordic is used. Running the simulation in Questasim, and exporting the switching activity with a FSDB dumper. Codelink and exploration of the nets in the RTL simulation was used for debugging, and to verify the right timing to start the estimation, based on the time window set in the Primetime time based power analysis. The firmware of all the scenarios start with a system reset, and initialization of the chip. This could include setting registers, activate peripherals and clock gating. The initialization sequence generates activity not relevant for the scenario and should be excluded from the estimation.

When the FSDB files for both RTL and Netlist was created, power analysis in Primetime was performed as a reference for correlation. To correlate estimation in RTL and Netlist a iterative process is followed. The three scenarios described is used for estimation with the default Spyglass setup as a starting point. Estimation with the default Spyglass setup gave the following results described in Table 1, 2 and 3. The result tables present the leakage, internal and switching power in mW of the different power components combinatorial, sequential. memory, clock and total power of Primetime and Spyglass estimation, and the deviation in percent.

Table 1: System idle no calibration

		Leakage[mW]	Internal[mW]	Switching[mW]	Total[mW]
Total power	Primetime	0.1255	0.7114	0.3546	1.1920
	Spyglass	0.1199	4.5633	0.7102	5.3934
	Diff	-4%	541%	100%	352%
Combinatorial	Primetime	0.0092	0.0060	0.0152	0.0306
	Spyglass	0.0084	0.0087	0.0051	0.0223
	Diff	-8%	43%	-67%	-27%
Sequential	Primetime	0.0076	0.0005	0.0004	0.0085
	Spyglass	0.0093	4.3848	0.0012	4.3953
	Diff	22%	854976%	201%	51424%
Memory	Primetime	0.1018	0.0303	0	0.1320
	Spyglass	0.1017	0.0304	0	0.1322
	Diff	-0%	1%	0%	0%
Clock	Primetime	0.0009	0.5717	0.3131	0.8857
	Spyglass	0.0003	0.1393	0.7038	0.8435
	Diff	-58%	-76%	125%	-5%

Table 2: AES no calibration

		Leakage[mW]	Internal[mW]	Switching[mW]	Total[mW]
Total power	Primetime	0.1256	1.4430	0.8870	2.4550
	Spyglass	0.1201	5.6538	0.8788	6.6526
	Diff	-4%	292%	-1%	171%
Combinatorial	Primetime	0.0093	0.2331	0.4770	0.7194
	Spyglass	0.0085	0.1940	0.1221	0.3246
	Diff	-8%	-17%	-74%	-55%
Sequential	Primetime	0.0076	0.0187	0.0255	0.0518
	Spyglass	0.0094	4.8924	0.0091	4.9108
	Diff	23%	26090%	-64%	9387%
Memory	Primetime	0.1018	0.4136	0.0119	0.5272
	Spyglass	0.1017	0.4250	9.8182E-5	0.5269
	Diff	-0%	3%	-99%	-0%
Clock	Primetime	0.0009	0.6744	0.3467	1.0220
	Spyglass	0.00034	0.1424	0.7475	0.8902
	Diff	-58%	-79%	116%	-13%

Table 3: Coremark no calibration

		Leakage[mW]	Internal[mW]	Switching[mW]	Total[mW]
Total power	Primetime	0.1255	1.5340	0.6628	2.3220
	Spyglass	0.1200	5.7494	0.8252	6.6946
	Diff	-4%	275%	24%	188%
Combinatorial	Primetime	0.0093	0.1504	0.2717	0.4313
	Spyglass	0.0085	0.1227	0.0735	0.2047
	Diff	-8%	-18%	-73%	-53%
Sequential	Primetime	0.0076	0.0067	0.0085	0.0228
	Spyglass	0.0093	4.8815	0.0049	4.8957
	Diff	23%	73218%	-43%	21390%
Memory	Primetime	0.1018	0.6070	0.0141	0.7228
	Spyglass	0.1017	0.6031	0.0001	0.7049
	Diff	-0%	-1%	-99%	-2%
Clock	Primetime	0.0009	0.6672	0.3427	1.0110
	Spyglass	0.0003	0.1419	0.7467	0.8891
	Diff	-58%	-79%	118%	-12%

The results shows a high overestimation on total power consumption and sequential, and an underestimation on combinatorial and clock power. This is due to important parameters used in the Spyglass estimation, like clock tree, drive strength, cell size and Vt mix is based on default numbers from Synopsys, independent of technology.

Spyglass has a feature to extract calibration data based on the layout for the design. With the calibration Spyglass is able to get statistics on cell size and drive strength, Vt mix and clock tree properties like fanout, icgc fanout and wireload as described in Section 3.6 A calibration of TC1 layout was performed and then gave the following results.

Table 4: System Idle default calibration

		Leakage[mW]	Internal[mW]	Switching[mW]	Total[mW]
Total power	Primetime	0.1255	0.7114	0.3546	1.1920
	Spyglass	0.1163	0.8729	0.3605	1.3498
	Diff	-7	23	2	13
Combinatorial	Primetime	0.0092	0.0061	0.0153	0.0306
	Spyglass	0.0104	0.0099	0.0054	0.0257
	Diff	12	63	-65	-16
Sequential	Primetime	0.0076	0.0005	0.0004	0.0085
	Spyglass	0.0087	0.4357	0.0014	0.4458
	Diff	15	84873	237	5127
Memory	Primetime	0.1018	0.0302	0	0.1320
	Spyglass	0.0961	0.0302	0	0.1263
	Diff	-6	0	0	-4
Clock	Primetime	0.0009	0.5717	0.3131	0.8857
	Spyglass	0.0010	0.3969	0.3537	0.7517
	Diff	13	-31	13	-15

Table 5: AES default calibration

		Leakage[mW]	Internal[mW]	Switching[mW]	Total[mW]
Total power	Primetime	0.1256	1.4430	0.8870	2.4550
	Spyglass	0.1163	1.6064	0.5227	2.2455
	Diff	-7	11	-41	-9
Combinatorial	Primetime	0.0093	0.2331	0.4770	0.7194
	Spyglass	0.0104	0.2187	0.1450	0.3742
	Diff	12	-6	-70	-48
Sequential	Primetime	0.0075	0.0186	0.0254	0.0517
	Spyglass	0.0087	0.5534	0.0107	0.5729
	Diff	16	2863	-58	1007
Memory	Primetime	0.1018	0.4136	0.0119	0.5272
	Spyglass	0.0961	0.4201	9.81E-5	0.5163
	Diff	-6	2	-99	-2
Clock	Primetime	0.0009	0.6744	0.3467	1.0220
	Spyglass	0.0010	0.4142	0.3667	0.7820
	Diff	13	-39	6	-23

Table 6: Coremark default calibration

		Leakage[mW]	Internal[mW]	Switching[mW]	Total[mW]
Total power	Primetime	0.1255	1.5340	0.6628	2.3220
	Spyglass	0.1163	1.7249	0.4721	2.3135
	Diff	-7	12	-29	-0
Combinatorial	Primetime	0.0092	0.1504	0.2717	0.4313
	Spyglass	0.0104	0.1894	0.1033	0.3032
	Diff	12	26	-62	-30
Sequential	Primetime	0.0075	0.0066	0.0085	0.02278
	Spyglass	0.0087	0.5210	0.0056	0.5354
	Diff	15	7726	-34	2251
Memory	Primetime	0.1018	0.6070	0.0140	0.7228
	Spyglass	0.0961	0.6028	0.0001	0.6990
	Diff	-6	-1	-99	-3
Clock	Primetime	0.0009	0.6672	0.3427	1.0110
	Spyglass	0.0010	0.4115	0.3630	0.7756
	Diff	13	-38	6	-23

The results of Synopsys default setup correlates better with calibration, but debugging on the different power classes shows overestimation on sequential power and underestimation on combinatorial and clock power. The inaccuracy of sequential power is caused by the fact that Spyglass estimates the clock pin of sequential cells as sequential power, while Primetime as clock power. In further analysis Spyglass Design Constraints are set to estimate sequential power as clock power. These results are about 10 % from Netlist estimations, still the results on the different power classes deviates and needs to be further improved.

For the next section debugging of the power numbers is performed on both power classes and on hierarchical levels to find the largest deviations and improve correlation. The exploration on a hierarchical level is useful to find inaccuracy in RTL estimation and with Spyglass Design Constraints, calibration data should be changed to improve correlation. This debugging shows inaccuracy in the IO PAD power which is computed as clock power in Spyglass. Also all analog macros shows inaccuracy [19] shows that analog macros are highly inaccurate between RTL and Netlist estimation, and left to future work. For further analysis this work isolates MCU0 for estimation to eliminate inaccurate IO PAD estimation, MCU0 consist of only digital circuitry and will eliminate the analog domain. Table 7, 8, and 9 shows the results of estimation on MCU0.

Table 7: System Idle MCU0

		Leakage[mW]	Internal[mW]	Switching[mW]	Total[mW]
Total power	Primetime	0.0568	0.3753	0.2496	0.6817
	Spyglass	0.0577	0.4453	0.1250	0.6281
	Diff	2	19	-50	-8
Combinatorial	Primetime	0.0054	0.0005	0.0001	0.0061
	Spyglass	0.0061	0	0	0.0061
	Diff	13	-100	-100	1
Sequential	Primetime	0.0056	0	0	0.0056
	Spyglass	0.0058	0	0	0.0058
	Diff	4	0	0	4
Memory	Primetime	0.0452	0.0302	0	0.0755
	Spyglass	0.0452	0.0302	0	0.0755
	Diff	-0	0	0	0
Clock	Primetime	0.0006	0.3445	0.2494	0.5945
	Spyglass	0.0005	0.4151	0.1250	0.5406
	Diff	-13	20	-50	-9

Table 8: AES MCU0

		Leakage[mW]	Internal[mW]	Switching[mW]	Total[mW]
Total power	Primetime	0.0569	1.1060	0.7804	1.9430
	Spyglass	0.0578	1.1779	0.2865	1.5222
	Diff	2	7	-63	-22
Combinatorial	Primetime	0.0054	0.2275	0.4619	0.6949
	Spyglass	0.0061	0.2086	0.1395	0.3544
	Diff	13	-8	-70	-49
Sequential	Primetime	0.0056	0.0181	0.0237	0.0474
	Spyglass	0.0058	0.0193	0.0093	0.0345
	Diff	5	7	-61	-27
Memory	Primetime	0.0452	0.4136	0.0119	0.4707
	Spyglass	0.0452	0.4201	9.81E-5	0.4654
	Diff	-0	2	-99	-1
Clock	Primetime	0.0006	0.4464	0.2829	0.7299
	Spyglass	0.0005	0.5297	0.1374	0.6677
	Diff	-13	19	-51	-9

Table 9: Coremark MCU0

		Leakage[mW]	Internal[mW]	Switching[mW]	Total[mW]
Total power	Primetime	0.0568	1.1980	0.5566	1.8110
	Spyglass	0.0577	1.2999	0.2373	1.5951
	Diff	2	9	-57	-12
Combinatorial	Primetime	0.0054	0.1448	0.2566	0.4068
	Spyglass	0.0061	0.1795	0.0979	0.2836
	Diff	13	24	-62	-30
Sequential	Primetime	0.0056	0.0061	0.0069	0.0187
	Spyglass	0.0058	0.0125	0.0042	0.0226
	Diff	5	103	-39	21
Memory	Primetime	0.0452	0.6070	0.0140	0.6663
	Spyglass	0.0452	0.6028	0.0001	0.6482
	Diff	-0	-1	-99	-3
Clock	Primetime	0.0006	0.4397	0.2790	0.7192
	Spyglass	0.0005	0.5050	0.1350	0.6406
	Diff	-13	15	-52	-11

These results shows an improvement in correlation, but some deviation in combinatorial power. The layout from the Netlist includes spef data which contains information about capacitance in the design to create more technology and design specific wireload models. By including these files to the Spyglass calibration wire load tables are created and selected by Spyglass for the standard cells and clock tree for more accurate estimations. Table 10, 11, 12 shows the results with included wireload data in the RTL estimation.

Table 10: System Idle with spef

		Leakage[mW]	Internal[mW]	Switching[mW]	Total[mW]
Total power	Primetime	0.0568	0.3753	0.2496	0.6817
	Spyglass	0.0578	0.4458	0.2930	0.7967
	Diff	2	19	17	17
Combinatorial	Primetime	0.0054	0.0004	0.0001	0.0060
	Spyglass	0.0062	0	0	0.0062
	Diff	15	-100	-100	3
Sequential	Primetime	0.0056	0	0	0.0056
	Spyglass	0.00585	0	0	0.0058
	Diff	4	0	0	4
Memory	Primetime	0.0452	0.0302	0	0.0755
	Spyglass	0.0452	0.0302	0	0.0755
	Diff	-0	0	0	0
Clock	Primetime	0.0006	0.3445	0.2494	0.5945
	Spyglass	0.0005	0.4156	0.2930	0.7091
	Diff	-13	21	17	19

Table 11: AES with spef

		Leakage[mW]	Internal[mW]	Switching[mW]	Total[mW]
Total power	Primetime	0.0569	1.1060	0.7804	1.943
	Spyglass	0.0579	1.1815	0.7933	2.0328
	Diff	2	7	2	5
Combinatorial	Primetime	0.0054	0.2275	0.4619	0.6949
	Spyglass	0.0062	0.2116	0.4271	0.6450
	Diff	14	-7	-8	-7
Sequential	Primetime	0.0056	0.0181	0.0237	0.0474
	Spyglass	0.0058	0.01940	0.0276	0.0530
	Diff	5	7	17	12
Memory	Primetime	0.0452	0.4136	0.0119	0.4707
	Spyglass	0.0452	0.4201	0.0044	0.4698
	Diff	-0	2	-62	-0
Clock	Primetime	0.0006	0.4464	0.2829	0.7299
	Spyglass	0.0005	0.5303	0.3340	0.8649
	Diff	-13	19	18	19

Table 12: Coremark with spef

		Leakage[mW]	Internal[mW]	Switching[mW]	Total[mW]
Total power	Primetime	0.0568	1.1980	0.5566	1.8110
	Spyglass	0.0578	1.3025	0.6449	2.0053
	Diff	2	9	16	11
Combinatorial	Primetime	0.0054	0.1448	0.2566	0.4068
	Spyglass	0.0062	0.1814	0.3010	0.4887
	Diff	15	25	17	20
Sequential	Primetime	0.0056	0.0061	0.0069	0.0187
	Spyglass	0.0058	0.0125	0.0139	0.0323
	Diff	5	104	100	73
Memory	Primetime	0.0452	0.6070	0.0140	0.6663
	Spyglass	0.0452	0.6028	0.0045	0.6526
	Diff	-0	-1	-68	-2
Clock	Primetime	0.0006	0.4397	0.2790	0.7192
	Spyglass	0.0005	0.5056	0.3254	0.8316
	Diff	-13	15	17	16

The spef information gave an increased correlation on combinatorial power, but decrease of clock power correlation. The spef file create wireload tables based on capacitance data from the layout, and is generated by Spyglass calibration. Each table represent different components in the circuit, presented by the capacitance dependent of the number of fanout cells of the gate. Spyglass chooses a table from the calibration for standard cells, which effects combinatorial and sequential power, and tables for clock power.

The wireload model used by Spyglass for the clock power shows a linear capacitance dependent on number of cells. This calibration data is wrong and might be an error in Spyglass. These results in overestimated clock power for high number of fanout cells. To get better correlation of clock power, a wireload model with less capacitance was used. This model gives lower capacitance and should decrease and correlate clock power better. The results with changed wireload table are presented in Table 13, 14 and 15.

Table 13: Final results System Idle

		Leakage[mW]	Internal[mW]	Switching[mW]	Total[mW]
Total power	Primetime	0.0568	0.3753	0.2496	0.6817
	Spyglass	0.0578	0.4457	0.1973	0.7009
	Diff	2	19	-21	3
Combinatorial	Primetime	0.0054	0.0004	0.0001	0.0060
	Spyglass	0.0062	0	0	0.0062
	Diff	15	-100	-100	3
Sequential	Primetime	0.0056	0	0	0.0056
	Spyglass	0.0058	0	0	0.0058
	Diff	4	0	0	4
Memory	Primetime	0.0452	0.0302	0	0.0755
	Spyglass	0.0452	0.0302	0	0.0755
	Diff	-0	0	0	0
Clock	Primetime	0.0006	0.3445	0.2494	0.5945
	Spyglass	0.0005	0.4154	0.1973	0.6133
	Diff	-13	21	-21	3

Table 14: Final results AES

		Leakage[mW]	Internal[mW]	Switching[mW]	Total[mW]
Total power	Primetime	0.0569	1.1060	0.7804	1.9430
	Spyglass	0.0579	1.1814	0.6925	1.9319
	Diff	2	7	-11	-1
Combinatorial	Primetime	0.0054	0.2275	0.4619	0.6949
	Spyglass	0.0062	0.2116	0.4271	0.6450
	Diff	14	-7	-8	-7
Sequential	Primetime	0.0056	0.0181	0.0237	0.0474
	Spyglass	0.0058	0.01940	0.0276	0.0530
	Diff	5	7	17	12
Memory	Primetime	0.0452	0.4136	0.0119	0.4707
	Spyglass	0.0452	0.4201	0.0044	0.4698
	Diff	-0	2	-62	-0
Clock	Primetime	0.0006	0.4464	0.2829	0.7299
	Spyglass	0.0005	0.5302	0.2333	0.7640
	Diff	-13	19	-18	5

Table 15: Final results Coremark

		Leakage[mW]	Internal[mW]	Switching[mW]	Total[mW]
Total power	Primetime	0.0568	1.1980	0.5566	1.8110
	Spyglass	0.0578	1.3023	0.5441	1.9044
	Diff	2	9	-2	5
Combinatorial	Primetime	0.0054	0.1448	0.2566	0.4068
	Spyglass	0.0062	0.1814	0.3010	0.4887
	Diff	15	25	17	20
Sequential	Primetime	0.0056	0.0061	0.0069	0.0187
	Spyglass	0.0058	0.0125	0.0139	0.0323
	Diff	5	104	100	73
Memory	Primetime	0.0452	0.6070	0.0140	0.6663
	Spyglass	0.0452	0.6028	0.0045	0.6526
	Diff	-0	-1	-68	-2
Clock	Primetime	0.0006	0.4397	0.2790	0.7192
	Spyglass	0.0005	0.5055	0.2246	0.7306
	Diff	-13	15	-19	2

The final results gives a correlation of 5 % or better which is good for RTL estimations. There are still some small deviations in sequential and combinatorial power, but the total power and main contributors to power yields great correlation and the deviations are small in relative power numbers.

4.3 RTL estimation methodology

This iterative process leads to a guide for improving the correlation in RTL power estimation listed below.

1. Establish reference results. Run power estimation on gate level, Primetime, on different scenarios.
   Use different power scenarios to establish a power profile of the power consumption and create reference power numbers in a gate level estimation. If a reference is established on same technology in a similar design, the calibration data from this layout could be used for reliable early stage RTL estimations.

2. Check PVT corners and library list against reference
   To get accurate power estimations it is important that the setup in Spyglass and the input files are correct. The RTL files, PVT corners, libraries and SDC settings in Spyglass has a big impact on the power estimation and the correlation of component deviation.

3. Select time window of activity file based on time based analysis and or simulation. Ensure high annotation on the activity files, at least 95 %. Otherwise fix simulation
   The reports from Spyglass contains information about the annotation of the activity files,

high annotation is crucial for accurate estimation.

4. Isolate the design, since analog and PAD IO estimation is inaccurate. Select only digital parts of the design, improve correlation on subcircuit.

5. Debug component deviation over total power. Find the most dominant power contribution, and the sub-component with highest deviations in correlation. Debugging on the circuitry hierarchy level to find the worst correlation and root cause for further analysis.

5 Implementation of low power techniques

In this section different low power RTL techniques are implemented in TC1, creation of logic synthesis, place and route layout and estimation in Netlist and RTL. The main purpose in this Chapter is to investigate fidelity and accuracy of power correlation on different design alternatives. This is to verify and further develop the methodology. The purpose of RTL estimation is to make tradeoffs between different design alternatives on a early stage in the design process. This requires high fidelity and accuracy to make the designer certain that the correct choice can be made. The results from Chapter 4 shows an accuracy about 5% on the default design, which is a good starting point. This Chapter will also investigate accuracy with the calibration data as the default design and explore efficiency of the low power techniques. Furthermore improvement of the correlation with new calibration data and debugging of power numbers will be performed to see how much the correlation alters with design changes. To verify correlation on the different designs netlist synthesis and layout is performed. The tool and implementation flow is described in Figure 17. This flow is done for each technique where the synthesis is done in Synopsys Design Compiler, place and route in Synopsys IC Compiler.

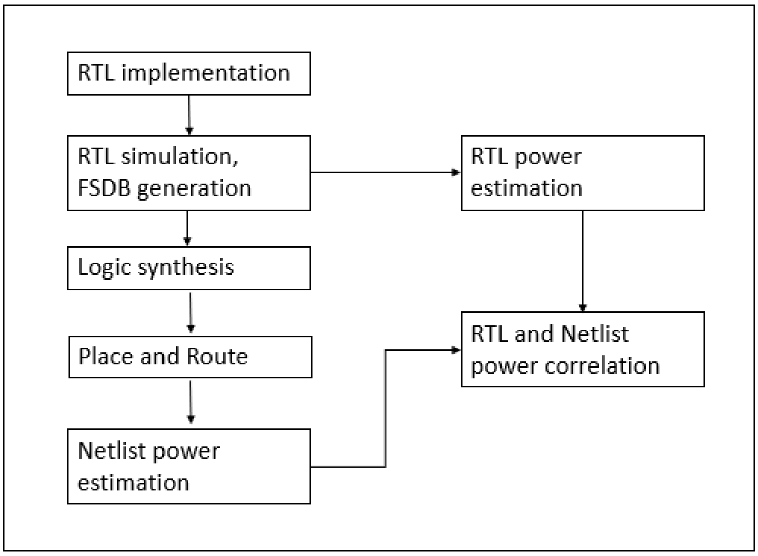

Figure 17: Implementation flow

Multiple rounds of implementation, simulation and power estimation in Spyglass is done to find a more power efficient design. The final design is then synthesized and performed a place and route, before estimation in both Spyglass and Primetime

5.1 Clock gating

As the final results shows in Chapter 4, the clock power is the largest contributor to the total power consumption in the design. Clock gating is a efficient technique to lower the dynamic power consumption. Clock gates could be applied manually at the HDL code or automatically interfered by the synthesis tool. Automated clock gates could be easily applied with few or no modification in HDL coding and applied directly at gate-level. In this work both clock gates implemented at RTL and automated synthesis with Synopsys Power Compiler [4] and Synopsys Spyglass [5] is explored.

5.1.1 Clock gate implementation in RTL

The results shows a high power consumption in the clock tree. The inferred clock gates from Synopsys synthesis inserts the clock gates close to the registers, and toggling in the clock tree contribute to most of the power consumption. By inserting icgc in the RTL on multiple hierarchical levels on parts of the design, the clock will be gated higher in the design and decrease the clock power. In this design the icgc enable signal is implemented in a register, so the clock gating could be controlled by software written in each scenario. The main focus has been on the largest contributors to clock power. The crypto accelerator and parts of the CPU is not active in the whole time window of each scenario and is explored for power reduction by inserting icgc. Figure 18 compare the results of the original design, clock gated design in Primetime, and estimation in Spyglass.

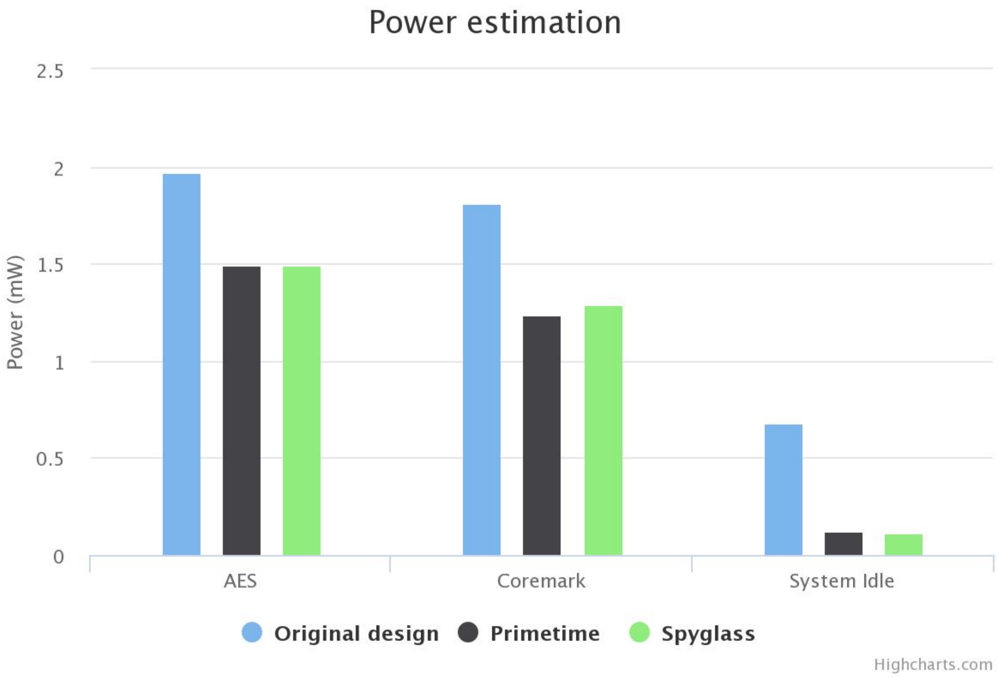

Figure 18: Estimation clock gated design

The results show a decrease of total power of 24.4 % for AES, 31.7 % for Coremark and 82.4 % for System Idle. The calibration from the default design gave good correlation between Netlist and RTL estimation. A recalibration is done on the new layout shown in Figure 19

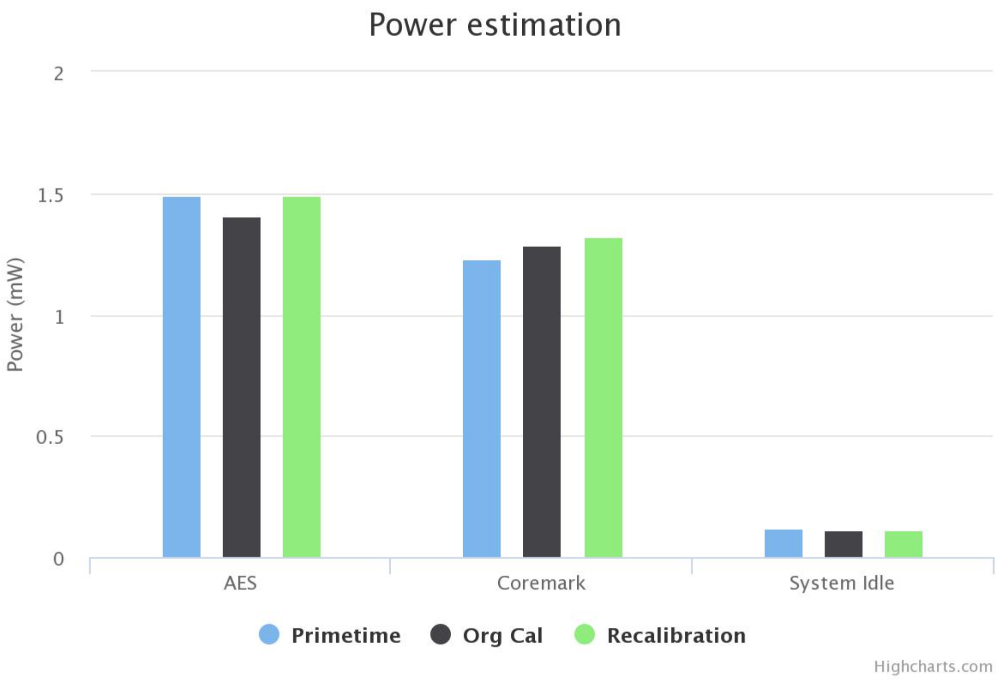

Figure 19: Recalibration clock gated design

As the results show, a recalibration gives high correlation on the AES scenario, and a decrease on the Coremark benchmark. From debugging of the power numbers a new calibration gave better correlation on combinatorial power, due to a updated SPEF file. In the Coremark scenario a small overestimation of clock power with new calibration data decreases the correlation slightly. These changes in the calibration and correlation are small and shows that some changes in the design will not effect the calibration significant.

5.1.2 Clock gating threshold experiment

The Synopsys synthesis tools automatically interferes clock gates in the design for power reduction. This is done where the synthesis tool sees registers with a bit width higher then a gating threshold, with a common enable signal and the synthesis tool will insert a icgc in the design. This spares the designer time in implementation, but may lead to ineffective clock gates. Also a interfered icgc in the design gives another AND gate in the clock tree which will impact the performance, area and power overhead. As the AND gate will give more delay and leakage in the clock tree. In this experiment clock gating threshold bit width from 1-6, 8, 16, 32 and 64 is explored in Spyglass. The most interesting results is then synthesized, performed place and route and netlist estimation to verify power correlation.

Figure 20 shows the results in Spyglass for different clock gating threshold bit widths for the original design and clock gated design, and both full chip estimation and estimation isolated to MCU0.

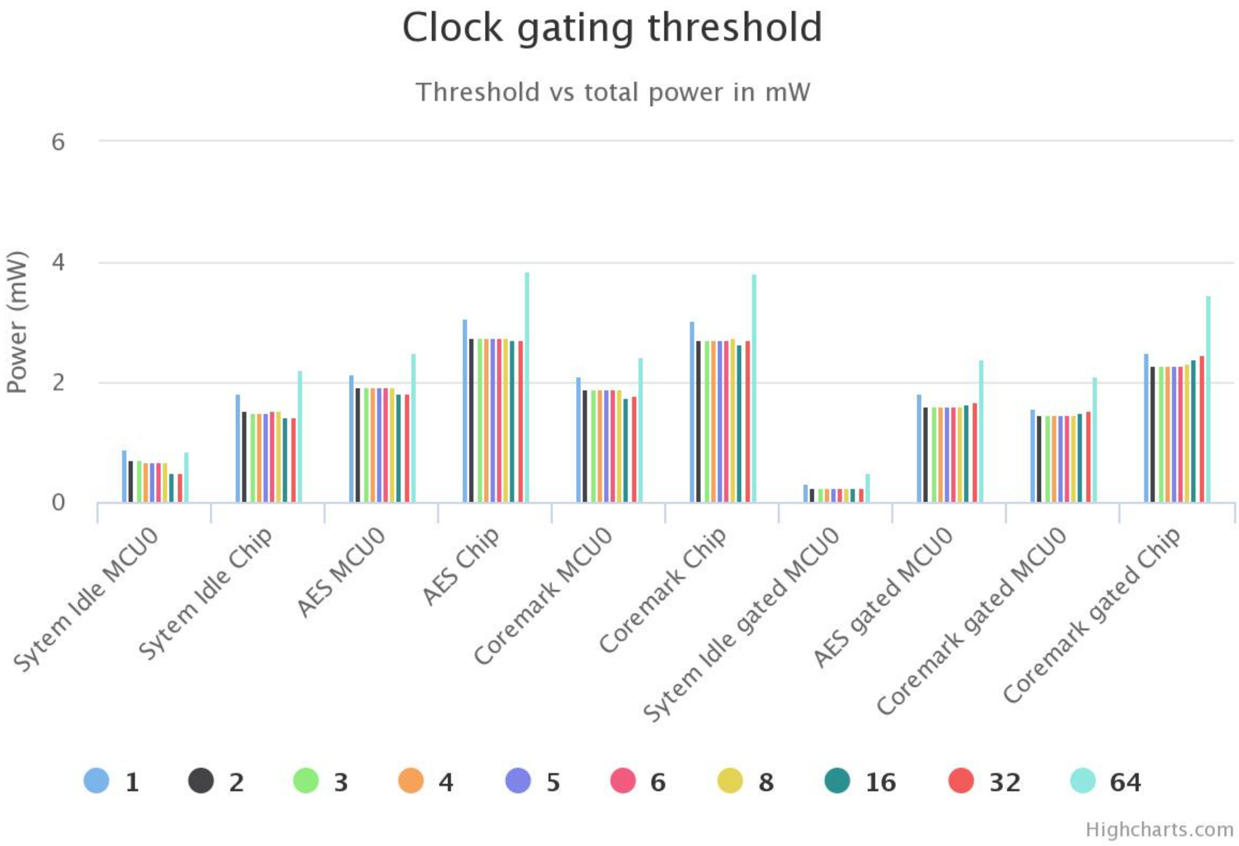

Figure 20: Results clock gating threshold

As the results show a bit width of 1 and 64 gives significantly worst power numbers, the default bit width of 3 set by the Synopsys tools is similar to rest of the results. A bit width of 16 and 32 gives better power results in all scenarios. Therefore a bit width of 3, 8 and 16 is investigated in netlist estimation. Figure 21 shows the correlation between Netlist and RTL estimation.

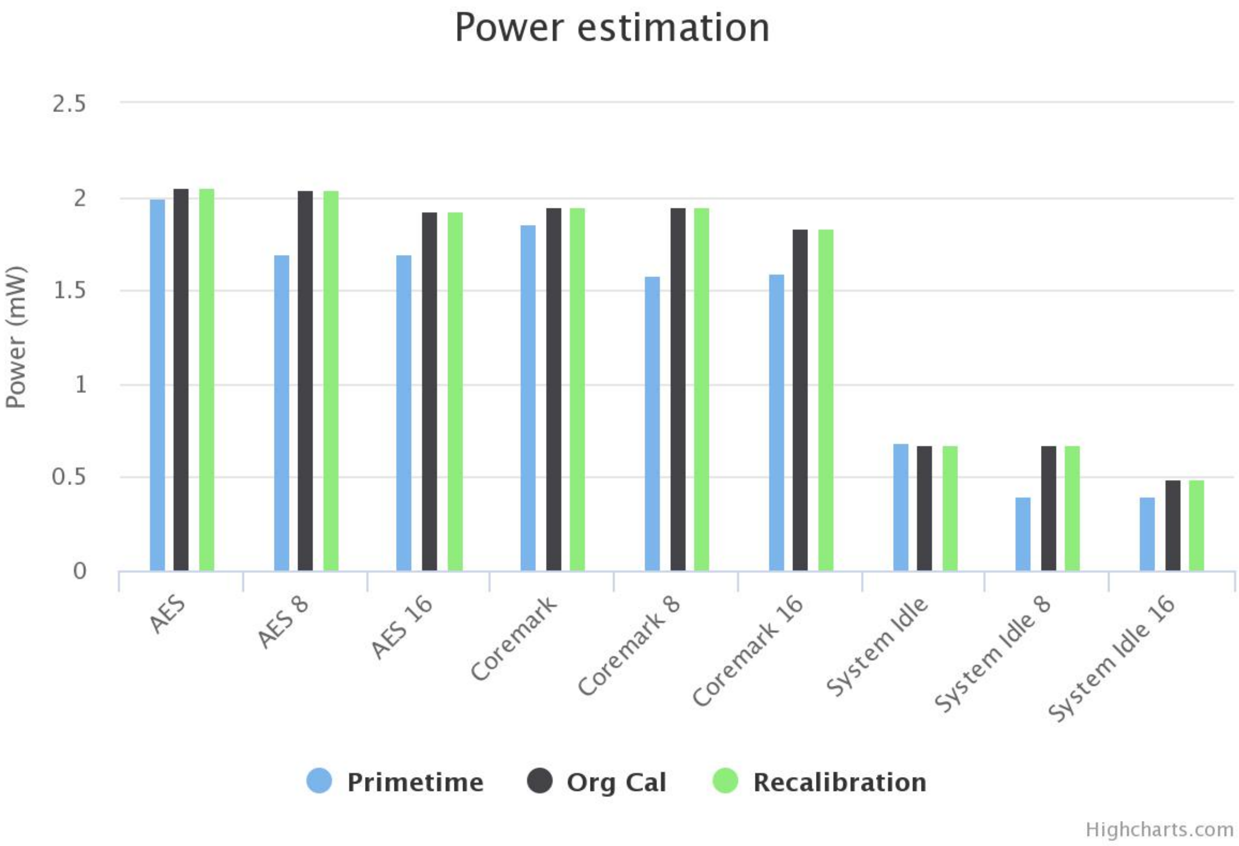

Figure 21: Results clock gating threshold

The netlist estimation gives higher power reduction with a bit width of 8 and 16 then RTL estimation. A new calibration in Spyglass based on the new layout gives small changes in the estimate. Spyglass overestimates the power in designs with higher bit width threshold, with a deviation of about 20% for bit widths of 8 and 16. These inaccuracies could be caused by a simplified clock tree synthesis in Spyglass, different number of icgc in Primetime and Spyglass, or inaccurate calibration data extracted from Spyglass calibration.

5.2 Bus encoding

Bus encoding is a technique to reduce the dynamic power consumption by reducing the transitions in the circuit. Statistics shows that typically, in a execution of a program 85% of the instructions is sequential accesses [3]. Which generates a high number of transactions on the bus, in buses with high capacitance this will generate a high dynamic power consumption. Bus encoding tries to address this problem by minimizing the number of transactions on the high capacitance bus, by introducing encoding schemes that make the transactions closer to memory. Literature [15] show that the encoding scheme T0 coding of the address bus, can achieve the biggest power savings. T0 coding [11] uses an extra bit line along with the address bus. This bit is set when the addresses on the bus is sequential, where the data on the bus is not altered but handled in the decoder. In TC1 the most activity on the bus is done between the CPU and the RAM. The bus encoding is implemented by inserting an encoder after the prefetcher in the CPU, a extra bit on the bus and a decoder before the RAM. Figure 22 shows the implementation in TC1.

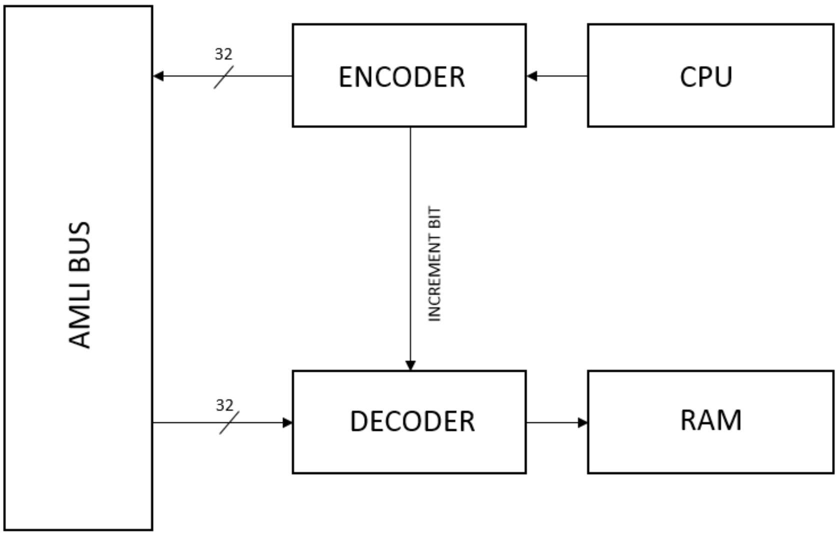

Figure 22: Bus encoding architecture

Figure 23 24 shows a block diagram of the implemented encoder and decoder.

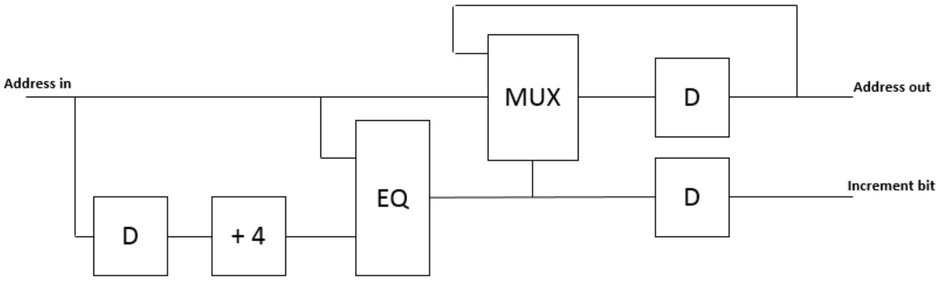

Figure 23: Encoder

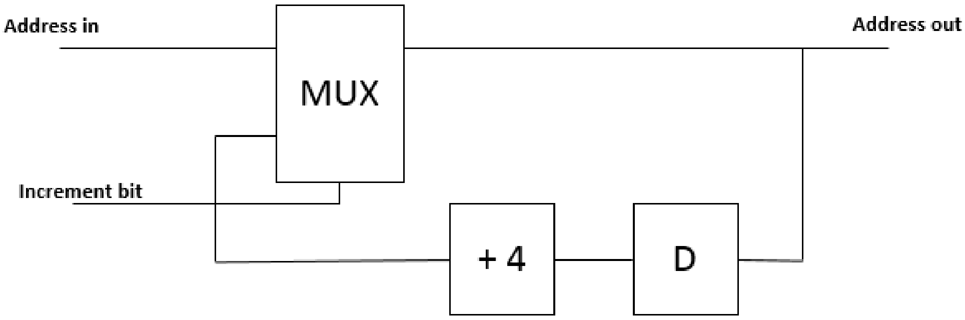

Figure 24: Decoder

In the encoder the address from the CPU is compared with the previous address plus 4, if these are the same the addresses are sequential, the increment bit is set and the address bus is not altered. The decoder adds 4 to the previous address if the increment bit is set, if not the address on the bus is used. Figure 25 shows signals from simulation in Questasim of the

Coremark benchmark with the bus increment signal, the clock, the address received on the bus for the decoder and the actual address for memory access.

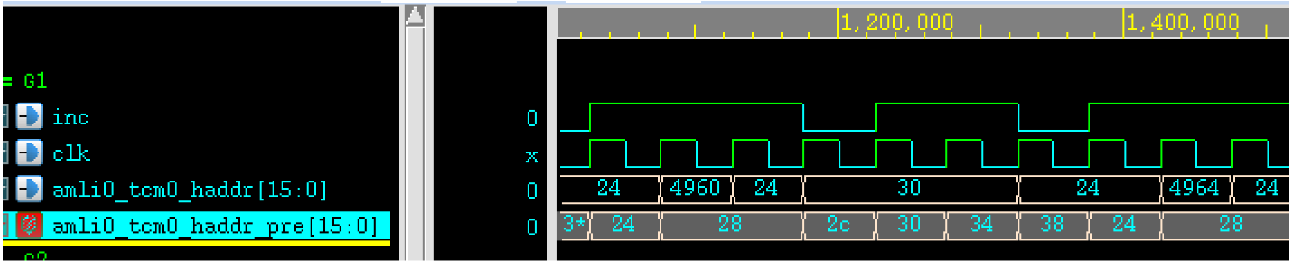

Figure 25: Bus encoding simulation with Coremark benchmark

As the Figure shows the addresses is sequential and the increment signal high about 75% of the time of the Coremark scenario. This decreases the number of transitions on the bus and has the potential for power reduction. Figure 26 shows the power estimation of the original design, the bus encoded estimation in both Spyglass and Primetime of the different scenarios.

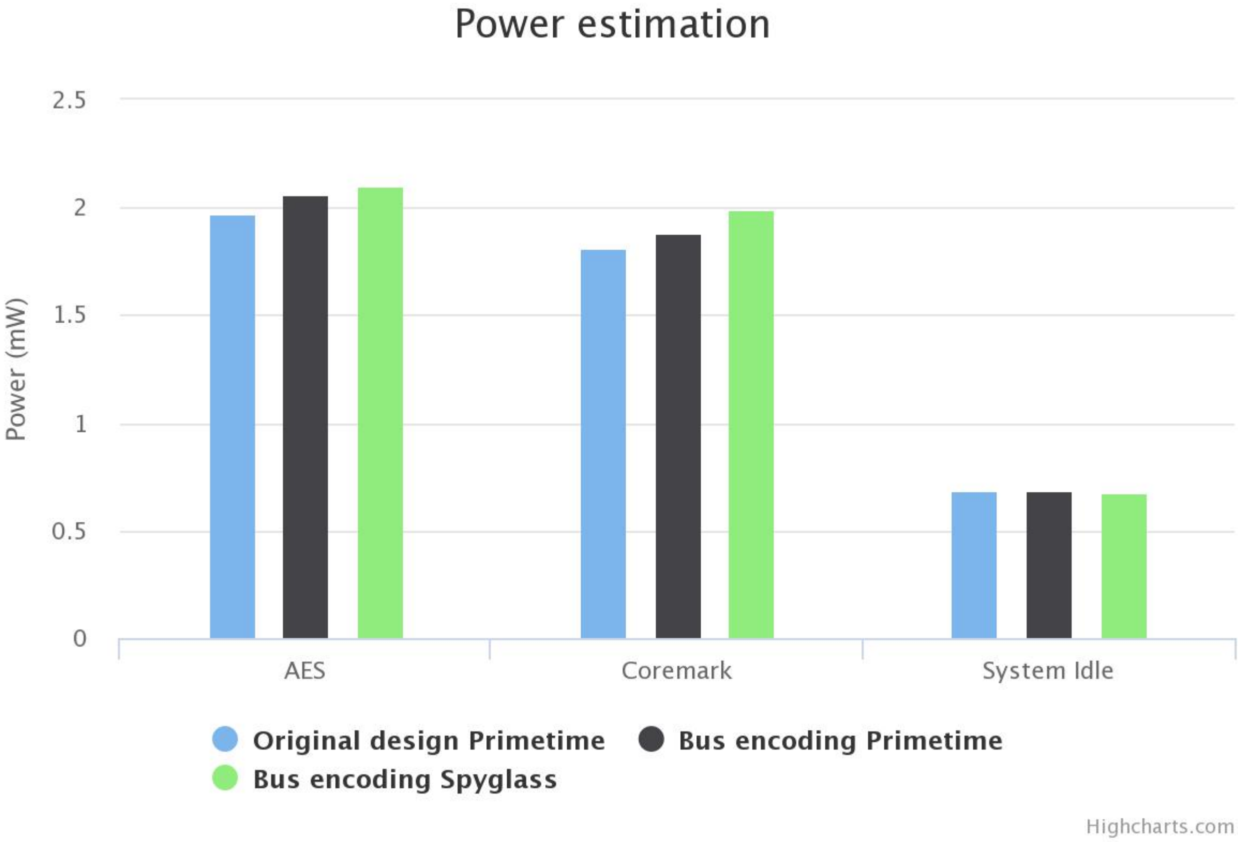

Figure 26: Bus encoding results

The results show an increase in power consumption of the bus encoded design, the power estimation gives reduction in dynamic power in the bus, but increased clock power due to overhead in the design. The encoder and decoder requires some extra registers and logic which introduces both area and power overhead. In this design the capacitance in the bus is too small to give a significant decrease of dynamic power. In other design with off-chip buses or high capacitance on-chip buses the T0 encoded bus could give better power results. The correlation between Primetime and Spyglass are good and within 5%, therefore no recalibration is done to the Spyglass setup.

5.3 Vt mix

The TC1 chip is synthesized for standard Vt cells. High Vt cells will decrease leakage power, but decrease the performance. To optimize the Vt mix, a more complex layout needs to be performed this is to verify that the timing constraints are met in all corners. As the setup and hold time for the transistors vary depending on temperature. This type of layout is not done, due to limited licences on Synopsys IC compiler for multiple corners and limited time. In this design the dynamic power is dominating, so using much time on reducing the leakage is not a priority. In table 16 the results of a Vt mix of 100% HVT cells is estimated in Spyglass to see the potential for reduction.

Table 16: System Idle VT mix, HVT SVT

		Leakage	Internal	Switching	Total
Total power	SVT	0.0578	0.4457	0.1973	0.7009
	HVT	0.0467	0.4462	0.1974	0.6903
	Diff	-19	0	0	-2
Combinatorial	SVT	0.0062	0	0	0.0062
	HVT	0.0008	0	0	0.0008
	Diff	-86	0	0	-86
Sequential	SVT	0.0058	0	0	0.0058
	HVT	0.0006	0	0	0.0006
	Diff	-89	0	0	-89
Memory	SVT	0.0452	0.0302	0	0.0755
	HVT	0.0452	0.0302	0	0.0755
	Diff	0	0	0	0
Clock	SVT	0.0005	0.4154	0.1973	0.6133
	HVT	5.42E-5	0.4159	0.1974	0.6133
	Diff	-90	0	0	0

As the table shows the leakage is reduced on both sequential, combinatorial and clock power, but the total power is only decreased by 2%. A vt mix of 100% HVT cells would probably not hold the timing constraints for all corners. And since the power savings is not largest the layout and Netlist estimation is left for further work.

6 Discussion

In this Chapter the results, limitations and future work is discussed.

6.1 Assumptions

In this thesis only typical process, voltage and temperature corner is investigated. These parameters effects power significantly as described in Section 2.5. Investigation of correlation in different corners should be performed in future work. This parameters can be set by using technology library files, LIB, for other corners.

6.2 Clock gating threshold

The clock gating threshold experiment gave some power savings in Primetime with higher threshold then 3, this should be further investigated in other designs.

The correlation between RTL and Netlist power estimation with a bit width of 8 and 16 has deviation higher then other designs and need thorough debugging and investigation. Possible sources of error could be different numbers of icgc in both total and on subcircuit level in Netlist synthesis and the simplified RTL synthesis performed in Spyglass. Investigate and debug the calibration data for the clock tree. A high bit width will decrease the number of icgcs, therefore the number of AND gates in the tree, which decreases the fanout capacitance. A different source could be a simplified clock tree synthesis in RTL, which does not cover the complexity in clock tree generation and parameters to the power analysis.

6.3 Validate methodology on other design

This methodology is only performed on one design and should be verified on a different and larger design. It would be interesting to test a design with the same technology and see the correlation with the calibration data from TC1, and how much changes are required to achieve good fidelity and accuracy. And if these changes could be anticipated by the designer. This way a accurate RTL estimation could be performed early in the design phase, before any synthesis or layout is available.

6.4 Vt mix

The Vt mix is only performed estimation in RTL in this work. A more complex place and route setup is required to verify timing in all corners to generate a layout. Therefore no correlation is performed, this is left for further work. But is an important artifact that should be investigated since bigger design and projects will have Vt mix optimization.

6.5 Other low power techniques

As the results show, different low power techniques yields great correlation and high fidelity in most RTL optimization. The clock gating threshold gave small changes in RTL estimation between different bit width, while good power savings in Netlist estimations. Other techniques should be investigated to see which gives good correlation with no or small changes to the calibration data.

7 Conclusion

Improvement and development of methodology for correlation between RTL and Netlist power estimation has been explored in this thesis, with the tools Spyglass Power and Primetime PX by Synopsys. The design explored is a small micro controller design, Test-Chip 1, by Nordic Semiconductor. Multiple iterations of estimation, changes in the Spyglass setup and configuration of the calibration data and input files was performed to achieve good correlation. In general the acceptable divergence of RTL estimation is 15%. The final results of the default design gave an divergence between netlist and RTL estimations below 5% with small component deviation between different power classes.

Different RTL power reduction techniques was implemented in the design to verify the methodology on different designs, and explore the efficiency of the techniques. Multiple options were explored in RTL estimation, and the most interesting results were synthesized and performed netlist estimations. Calibration data from the default design to explore how much the calibration data effects the correlation on similar design with same technology.

The implemented bus encoding gave no power savings, due to power overhead and a low capacitance bus. The manually inserted clock gates gave power savings of 82.4% in low activity scenarios. For both clock gating manually inserted in the RTL and bus encoding the correlation was below 6%, with small changes in accuracy with new calibration from netlist. Hence this methodology could be used for reliable estimation in the RTL phase. The clock gating threshold experiment, where different bit widths is explored for automatically interference of clock gates by the synthesis tool, shows potential for power savings in netlist estimations. In this designs the correlation between RTL and netlist estimation is decreased to 20%, and needs investigation and is left for future work.

This thesis explores a relatively small design, and should be further verified in a bigger design. Also this work is limited to a single voltage design and only typical process corners are explored.

References

[1] Robert W. Brodersen Anantha P. Chandrakasan. *Minimizing power consumption in digital CMOS circuits*. Proceedings of the IEEE, pp. 498-523. Volume 83, Issue 4, 1995.

[2] Synopsys ARM. *Low Power Methodology Manual, For system on chip*. 2007 by Synopsys, Inc ARM Limited.

[3] J. L. Hennessy and D.A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann Publishers, Inc. San Mateo, CA. Second edition, 1995.

[4] Synopsys Inc. *Power Compiler User Guide*. Y-2006.06, June 2006.

[5] Synopsys Inc. *Spyglass Explorer User Guide*. Version N-2017.12, Dec 2017.

[6] Texas Instruments. *CMOS Power Consumtion an C_{pd}Calculations*. URL: `http://www.ti.com/lit/an/scaa035b/scaa035b.pdf`.

[7] J. Monteiro J. B. Martins R. Reis. *Capacitance and Power Modeling at Logic-Level*.

[8] B. Taskin J. Lu. *Post-CTS Delay Insertion*. Department of Electrical and Computer Engineering, Drexel University, Philadelphia. Nov 2009.

[9] K. Kirsch K. Muller-Glaser and K. Neusinger. *Estimating Essential Design Characteristics to Support Project Planning for ASIC Design Management*. IEEE International Conference on Computer-Aided Design‘91, Los Alamitos, CA, pp. 148-151, November 1991.

[10] M. Hamid K. Roy. *Leakage Current Mechanisms and Leakage Reduction Techniques in Deep-Submicrometer CMOS circuits*. Proceedings of the IEEE, Vol 91, NO 2, Feb 2003.

[11] E. Macii L. Benini G. De Micheli. *Asymptotic zero-transition activity encoding for address buses in low-power microprocessor-based systems*. Great Lakes VLSI Symposium, March 13 1997.

[12] P. Landman. *Low-Power Architectural Design Methodologies*. Ph.D Dissertation, US Berkeley, August 1994.

[13] D. Liu and C. Svensson. *Power Consumption Estimation in CMOS VLSI Chips*. IEEE Journal of Solic State Ciruicts, pp. 663-670, June 1994.

[14] M. Poncino M. Bruno A. Macii. *RTL power estimation in an HDL-based design flow*. IEEE Proceedings - Computer and Digital Techniques, pp. 723-730. Vol 152, 2005.

[15] N. Dutt M. N. Mahesh D. S. Hirschberg. *Encoding Techniques for Low Power Address Buses*. Center for Embedded Computer Systems Department of Information and Computer Science University of California.

[16] W. P. Burleson M. R. Stan. *Bus-invert coding for low-power I/O*. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol.3, pp. 49-58 March 1995.

[17] F. Najm. *Towards a High-Level Power Estimation Capability*. International Symposium on Low- Power Design, April 1995.

[18] David Harris Neil H.E. Weste. *CMOS VLSI Design: A Circuits and Systems Perspective*. 3rd Ed. Addison-Wesley.

[19] Sondre Rennan Nesset. *RTL power analysis and development of MEMA tool flow on a full chip design*. Student Project.

[20] J. Rabaey P. Landman. *Architectural Power Analysis: The Dual Bit Type Method*. IEEE Transactions on VLSI Systems, pp. 173-187, June 1995.

[21] J. Rapaey P. Landman. *Activity-Sensitive Achitectural Power Analysis*. IEEE Transactions on CAD, June 1996.

[22] Y. Manoli R. Bhutada. *Complex Clock Gating with Integrated Clock Gating Logic Cell*. Design Technology of Integrated Systems in Nanoscale Era, 2007.

[23] P. Chau S. Powell. *Estimating Power Dissipation of VLSI Signal Processing Chips: The PFA Technique*. VLSI Signal Processing Chips 4, pp. 250-259, 1990.

[24] *Semiconductor Industry Associate*. International Technology Roadmap for Semiconductors, 2007.

[25] Synopsys. *Library Compiler Methodology and Function Modeling User Guide*. J-2014.09 edition, September 2014.

[26] Synopsys. *Library Compiler Timming, Signal Integrity, and Power Modeling User Guide*. J-2014.09 edition, September 2014.

[27] Synopsys. *Power Correlation CookBook*.

[28] Synopsys. *Spyglass Power Estimation and Spyglass Power Reduce Methodology*.

[29] Synopsys. *Synopsys Low Power Verification Tools Suite User Guide*. Version K-2015.09-SP1, December 2015.

[30] J. P. Uyemura. *Introduction to VLSI Circuits and Systems*. John Wiley Sons, 2002. ISBN 9788126509157.