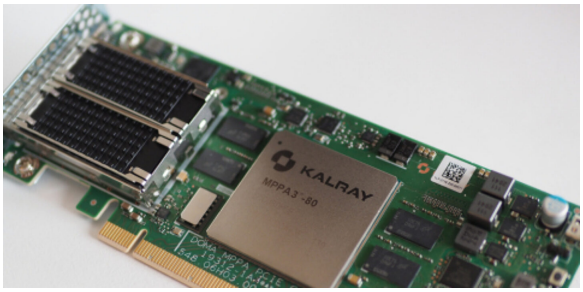
Osama Mowafy

MasterNanotech
Promotion 2022
Kalray, 180 avenue de l'Europe 38330 Montbonnot

RTL Power Estimation & Optimization of Kalray's MPPA using PowerArtist & Joules

from 15/02/2022 to 15/08/2022



Under the supervision of :
-Company Supervisor : Laurent Faniel, Kalray, lfaniel@kalrayinc.com
-Phelma Tutor : Morin Katell, PHELMA, katell.morin-allory@grenoble-inp.fr

Confidentiality : no

Ecole nationale supérieure de physique, électronique, matériaux
Phelma
Bât Grenoble INP - Minatec
3 Parvis Louis Néel - CS 50257
F-38016 Grenoble Cedex 01
Tél +33 (0)4 56 52 91 00
Fax +33 (0)4 56 52 91 03
http://phelma.grenoble-inp.fr

Problem description

The continuous miniaturization and integration of transistors over small area of the chip increase the demand of low power chip in order to be able to process huge amount of data and instructions. Low power ICs have become widely spread in wearables and IOT. Kalray is one of the companies that leads at the field of data centering and processing applications. The company has its own processors which called Kalray's MPPA (Multi Parallel Processing Application) processors which could handle multiple workloads in parallel with no bottlenecks to enable smarter, more efficient and energy wise data-intensive applications. So, one of the main requirements of such processors is the high performance which requires high speed and high workload. This is necessarily leading to high consumption of power and heat consumed from the chip. The first part of this thesis gives a brief introduction about Kalray's MPPA and the difference between the first release which called MPPA Coolidge V1 and future release MPPA Coolidge V2 which planned to be taped out at the end of this year. The second part of the thesis is to introduce the different tools that we used in order to monitor the RTL power consumption and different power optimization techniques that could reduce the power wastage. The last part of the thesis will be dedicated to the measurement of power obtained by different tools, the amount of power saved at the design and the correlation between the RTL estimation and the gate level implementation in order to improve the accuracy of the results.

Abstract

The increased complexity of integrated circuits in addition to the demand for low power ICs made accurate and reliable power estimation of the RTL a high priority task for the designer. The early power estimation allows the designers to take an early decision that makes them meet the specifications and requirements dedicated to their design. There are many tools for RTL power estimation that could compute efficiently the power consumption at the RTL level. Kalray is one of the leading companies that introduces technology IP, processors and solutions for intelligent Data Processing from Cloud to Edge. Intelligent data processing requires processors with very different architectures from traditional ones. The current release of Kalray's processors is two times the previous one in terms of the performance and the storage capabilities which will lead to more power consumed by the new release. In order to meet the power specifications to cope with wide range of applications more than data centering, the power consumption by the design should be monitored and minimized to certain level that could be desirable and competitive with other processors in the market. In this work, RTL power estimation tools from different providers are introduced in order to obtain a power profile for the design at different scenarios. RTL power estimation tool PowerArtist provided by Ansys is one of the most effective tools to reduce power of the design with a lot of reduction techniques introduced by the tool. On the other hand, the Cadence Joules RTL power solution delivers more accurate data in terms of power estimation at the RTL stage as the tool incorporates rapid prototype technology from Cadence Genus Synthesis solution. Faster analysis of RTL power could be a great demand for complex design which takes much time to elaborate, PowerPro tool supported by Siemens could be a potential candidate that combines between rapid RTL power estimation and power reduction techniques.

Power optimization of Kalray's processors which will be taped out at the end of this year is a high priority task for the hardware team in order to determine the power density for the design. The design has 16 processors that are the computing engines in addition to 8 Megabyte memory as the most dominant blocks. Several optimization techniques are implemented at the RTL level in order to fix the power bugs and reduce the power consumption for different power scenarios. The modifications implemented to the design reduced the power consumption by 18% to 37%. Thanks to these power reductions, the power consumed by the current design is the same as the old release in spite of the increased complexity, performance and memory size. At this work, there are multiple steps in order to perform power analysis and optimization such as the development of scenarios or testcases that will dump the modules in order to compute the power, elaboration of the design and estimation of the average power. The PnR step for any design is mainly affecting the net capacitance which reflected to the dynamic power, this step should be monitored and conducted perfectly in order to reduce the glitches and power of the nets by accurate netlist power estimation. Design verification also is executed to validate the modification implemented at the design to optimize the power. Extraction of calibration data from the netlist improves the accuracy of the RTL power estimation

which achieve reliable power data for the RTL at any stage. We will investigate the methodology for the extraction of some data such as cell distribution and wireload model from the netlist.

Preface

This report is a result of the master thesis in the second year of the Master of Nanotechnology for integrated circuit technology (ICT) program at Ecole nationale supérieure de physique, électronique, matériaux, Phelma school at Grenoble institute. The company named Kalray in Grenoble, France posted a project title “power optimization of Kalray’s MPPA”. The aim of the project was to monitor the power consumption of their design and optimize the power consumption. Also, it involved the improvement of power of the design at the RTL level by Cadence & Ansys power’s engine tools. I was provided by suitable workplace and all the power tools with access to all the design tools. This project gave me insight in the low-power digital design flow and the Back-End flow by working with engineers at different aspects of the design. I would like to thank my supervisors professor Katell Morin from Phelma and Marc Schmitz and Laurent Faniel from Kalray for their support and guidance during my work at this project.

Grenoble August 14, 2022
Osama Mowafy

Contents

List of Figures

List of Aconyms

CV1 Kalray Coolidge V1

CV2 Kalray Coolidge V2

MPPA Multi Parallel Processing Application

FSDB Fast Signal Database

PA PowerArtist tool, Ansys

DC Synopsys Design Compiler

HDL Hardware Description Language

EDA Electronic Design Automation

RTL Register Transfer Level

LNR Low Activity non-Enabled Register

LER Low Activity Enabled Register

CGE Clock Gating Efficiency

SDC Synopsys Design Constraint

SPEF Standard Parasitic Exchange Format

PnR Place & Route

TCA Task Control Area

1 Introduction

1.1 Motivation

Power consumption is increasing concern in today's high performance low power designs. Increasing device densities and the performance make power consumption considerations more important than ever. Today's larger devices could implement much more functions including high speed applications that were unheard of only few years ago. As devices running at high speeds and performing these more advanced functions more power is required and consumed. There are many challenges in regard to power when creating such designs. It's necessary to estimate system power usage at the design flow to determine power supply requirements which is difficult to do especially with multi applications which do many different functions and it's hard to get absolute numbers about power. The components should be kept within a fixed power budget which depends on the power supplies chosen which is not option to get outside this power budget because it requires redesigning the power supplies and cooling solutions. At the same time, the cost of the system must be met by keeping the power budget and the cooling system at the allowed range. The system also should be reliable as more cooling hardware and more fans means more moving parts which increases the opportunity for failure. Finally, the design cycle is so short and the thermal and power constrains must be met. That's why the power budget should be meet at every step of the design flow and the focusing on power supplies and cooling systems should be as early as possible in the design cycle which will help to speed the overall system closure. In order to understand how the power analysis tools you need to understand first the power usage enough at the electronic devices when a device is first powered up the current usage ramps up until it reaches its static level of power usage. Static power is also known as standby power which is the current driven by the device resources that are not clocked at the design this includes leakage from the functional block resources. So, static power shows the power consumption of the device if no clock signal is applied. Dynamic power is the main component of power consumption as it's the power that used to perform the main functions with specific speed and performance. At the next section, we will explore the design of Kalray's MPPA in order to know more about the design and understand how the design is working.

As the world is witnessing a large amount of data which the current technologies are not capable of handling and processing efficiently, there is a demand for a new generation of highly and massively parallel processor which could be considered as a new generation of intelligent processors. Kalray's MPPA manycores is one of these intelligent systems that have the capability to handle, process and store huge amount of data. In order to cope with the rapid changes of the requirements and demands of the market and current applications such as IOT and autonomous cars, Kalray decided to improve their processor by doubling the performance and the memory size so that they could target wide range of customers at different fields not only data centering from cloud to edge. Power management has become more crucial issue at the current applications, and it became essential to give much care to this parameter which affects the performance and reliability of the final products. To minimize the power consumption at the RTL level, many tools from different CAD providers are introduced in order to reach the optimal power consumption that meets the new generation of Kalray's MPPA processor. Power estimation could be performed at different stages from the design phases, netlist estimation gives the most accurate results as it will represent the real design after all the synthesis steps, on the other hand, it requires more time and effort. RTL power estimation could be a middle solution for the designers to get power figures faster than the netlist power estimation, however, the result will be less accurate. The gap between the RTL & netlist power estimation could be decreased by extracting some parameters and data from the netlist which could improve the accuracy of the RTL as we will investigate at this thesis. PowerArtist helps the designer to change the RTL design description, coding style, clock gating style and specifications in order to see the impact of these modifications on the power of the design. Fast RTL power estimation could evaluate the power effectiveness at early stage of the design such that different architectures and styles could be evaluated early without the need to perform all the steps of synthesis and PnR to reach the final netlist which takes hours to get the final layout. Figure 1 shows the trade off between RTL & Gate-level power estimation in terms of time & effort which gives a great advantage for the RTL power estimation to assess the improvements of the power and the area tradeoffs.

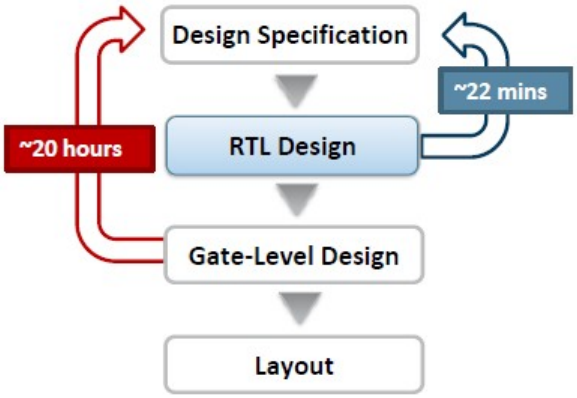


Figure 1: RTL power estimation vs Gate-level estimation [1]

1.2 Objectives, limitations and main contributions

This project mainly aims to explore RTL power estimation in PA, Joules & Powerpro. The goal is to monitor the power analysis of Kalray's chip in order to meet the specifications and requirements of the design. The objective also is to fully describe the methodology for RTL power estimation and how to extract calibrated parameters from the layout to the RTL power estimation in order to have accurate results. This effective methodology could be referenced to any design for reliable RTL power estimation for different tools. The results obtained from each tool could be a perfect indication for the strength points for the proposed tools which could be a perfect reference for designers to choose the optimal tool that matches with their design. The second part of this project is to optimize the power for Kalray's design by introducing some optimization techniques, illustrating some power reduction cases for each tool and developing an effective way to look at the proposed modifications and implementation. This part is very critical for this project because the main goal of this project is to optimize the power consumption to meet the specifications of the new version of processors. We could list the main contributors in the following parts:

1. Exploration of different RTL power estimation tools.
2. Investigation of the power results and how to get more accurate data.
3. Discussion of different optimization techniques and evaluation of the final reductions.

1.3 Kalray's MPPA

The architecture of the Kalray's MPPA manycores is a multi-core architecture that is based on Massively Parallel Processor Array (MPPA) architecture which consists of multiple clusters that have enormous computing capabilities and connected to each other, I/O peripherals and external memory. The main component of the MPPA are the clusters, there are 5 clusters, each cluster is composed of 16 processing core and one core which called resource management which dedicated to handle the initialization and synchronization between the processors and manage the security and safety functions. The processors are the application zone inside the cluster which execute the instruction and do the required computation, in addition to 8MB local memory called SMEM, each one has 16 memory banks and other peripherals related to the connections and control. Figure 2 shows the architecture of Kalray MPPA manycore which is scalable architecture for high performance and lower cost system. Every cluster has 16 processors and the shared memory which are the main building blocks of the cluster, that's why we will give much care to them during the power analysis.

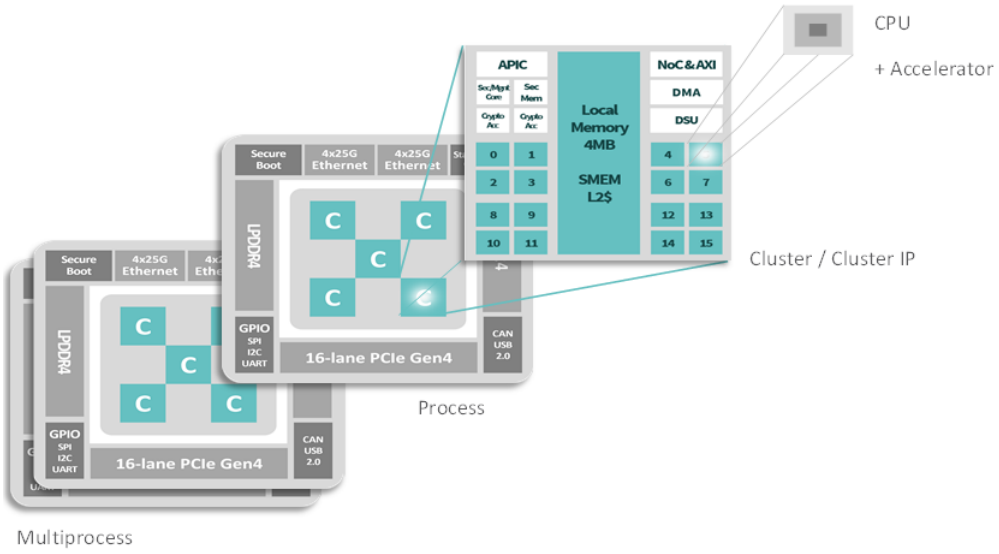


Figure 2: Kalray's MPPA architecture

1.4 Kalray Coolidge V2

Kalray offers technology IP, processors and solutions for intelligent data processing from cloud to edge. Intelligent data processing requires processors with very different architectures from traditional ones. Kalray has launched different generations for their processors. MPPA processors family started with BOSTAN which is 28nm technology processor, then COOLIDGE V1 which is 16nm technology processor and finally COOLIDGE V2 which is same technology and under development to be released at the end of this year. It's important to get knowledge about the difference between CV1 & CV2 which are the same technology in order to get sense of how we improved the power of CV2 such that it became close to the power consumption of CV1. In general, the purpose of CV2 is to have optimized version of CV1, pin to pin and software compatible, CV2 has double performance and memory size compared to CV1. In addition, processors at CV2 have more features such as more instructions supported, less pipelines stages and floating-point data instruction sets which is very large module. Finally, the register widths have also been increased which means more combinational and sequential cells to the design. Figure 3 shows the architecture of processor core for CV2, we should notice the following from the architecture of the processor core: first, if we look at the design of the coprocessor which executes complex and data intensive instructions, we should

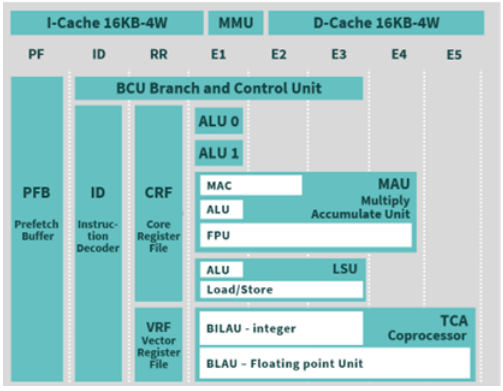


Figure 3: Processor core architecture

observe the difference in the complexity and timing between BILAU module which handles and manipulates the integer data type in 2 cycles and BLAU which is the floating-point unit that handles data in 4 cycles. MAU (multiply accumulate unit) is consisting of two main components that shared the same input flops: ALU unit that performs computations in one cycle and FPU unit which has multipliers and executes instructions in 4 cycles. Finally, TCA or coprocessor is the most significant module inside the processor core which could handle complex and data massive instructions which consumes a lot of power and will be the main focus for the power scenarios as we will explore at the following sections. The coprocessor has a lot of adders, shifters and multipliers which consumes a lot of power at the power scenarios that related to the coprocessor.

2 Power and energy fundamentals

This section describes the concepts of power estimation and provides some details about how the calculation of power is performed during power analysis at the power tools. All the following concepts are applied to all RTL power estimation tools. As we know the energy is defined as the total work needed for an object like a charge to move under the effect of voltage, the power is the time average of energy. Power could be divided into two main categories: dynamic power and static power which will be described at the following sections:

2.1 Static power

Static power is the power consumed by the circuit at the steady state, frequency independent and is one of the characteristics of the cells that could be controlled based on the technology and features of the cell. The static power is always called the leakage power or the leakage current that comes from the subthreshold current, gate leakage current and reverse biased junction leakage. These factors are highly dependent on the technology of fabrication that's why the value of the static power could be found at the liberty files provided by the fabrication suppliers. PowerArtist and other tools are mainly calculating the static power based on the type of cell and current state. The static power is highly dependent on the temperature that's why we could observe that at higher temperature, the static power is increasing. Moreover, the type of Vt cells has a great impact on the leakage power, for example LVT cells are more speed but more leaky as we will investigate at the design with different Vt cells.

2.2 Dynamic power

Dynamic Power is the most dominant factor of power consumption. Dynamic power which also called switching power is mainly dependent on the frequency of the operation and the power supply. Moreover, one of the basic contributors of the dynamic power is the net capacitance which could be defined as the amount of power to charge and discharge the load capacitance driven by specific cells and this capacitance is wire capacitance and fanout load input pin capacitance. The last factor is called activity factor which represents the probability of specific input or net to toggle between zero and one. The following formula shows all the factors that contribute to the dynamic power:

$$P_{Dynamic} = C_L * V_{DD}^2 * f_{clk} * \alpha$$

Where:
$V_{DD} = supply\ voltage$
$f_{clk} = clock\ frequency$
$\alpha = activity\ factor$
$C_L = external\ load\ capacitance$

There is another factor that contributes to the dynamic power which is the internal power of the cells. This factor is calculated based on the transition time and the load capacitance of the cells. The liberty files have matrices that give the amount of power consumed by the internal circuitry of the cells based on the given transition time and load capacitance.

3 Literature Review

This chapter describes the tools and techniques introduced for the purpose of power analysis and some techniques for power optimization. The tools introduced by this work are highly expensive in terms of the cost of license which makes it widely used only at the industry field but not at the research domain. For example, the cost of PA tool license is 53k $/year which made it used only for industry domain not a research purpose. Power and CGE Analysis using PowerArtist [**4**]: this paper is presented the way to setup the flow for PA and the automation of the PA tool for early RTL power and clock gate efficiency estimation by a group of researchers at California State University. The paper introduces all the features and command lines used at PA and the way to look at the power reports and graphical user interface. The flow of PA and the inputs of the tool are described in details at this project in addition to how to change the commands in order to setup the flow and generate all kinds of power reports like clock gating efficiency and hierarchical power modules. However, the researchers did not run the tool on real design which will be big advantage of our work on this project. The results that we will mention at this thesis will be so valuable to judge and evaluate this tool as there are no other papers or work do such idea. This paper is more descriptive than the manual for PA which made more helpful to understand how the tool is working.

Power reduction through RTL clock gating [**6**]: this paper describes an effective way to reduce the power consumption for ASIC by clock gating. The integration of the clock gating could be done through the synthesis tool such as DC. The implementation of the clock gating saved about 72% for the design which was three bits counter. This paper describes the importance of clock gating technique to save the power consumption of the design at different scenarios.

RTL Power Estimation Flow and its Use in Power Optimization [**2**]: in this paper, the author presented a typical flow or methodology to measure RTL power in Spyglass Synopsis tool and the netlist estimation in PrimeTime. This is the most relevant work to our project, the implementation of power reduction technique such as clock gating is proposed at this paper. The author estimated the gap between the RTL & Netlist power estimation by 5% for all the scenarios which is completely different from our estimation which could be explained by the difference at complexity of the two designs. The implementation of the clock gating saved about 82% for all the scenarios. Moreover, the netlist estimation shows a power reduction of 42% which correlated with the RTL power reduction. The leakage current is highly influenced by the threshold voltage of transistor, mix of Vt cells are explored in order to investigate the improvement of the leakage power at the design. the deviation between the RTL & Netlist reached about 10% for the worst case scenario which was improved by the work to reach 5%. The author mentioned the typical flow of RTL power estimation for different power scenarios and the implementation of different power reduction techniques. The correlation between the RTL & Netlist is quite good due to the simplicity of the design compared to our design.

4 Power tools

4.1 PowerArtist

Powerartist tool is one of the tools provided by Ansys Inc[**1**] that is dedicated to RTL power estimation which performs power analysis and automatic reductions for RTL design. The tool is not only estimating the power consumed by the designs but also proposing some power enhancement by scanning the whole design searching for any power bugs that could lead to power wastage and gives an idea about the solution it has a set of power reduction techniques that called PowerBots that could improve the power consumed by clock, inputs and memory. At the following sections we will described how the tool is working and the functionality of each step to gain better understanding about the tool before going into our main project.

4.1.1 Power analysis

At this section, we will describe how the tool performs RTL power analysis for any design and the description of each step such as elaboration, average power and time-based power analysis. Figure 4 describes the flow of PA tool for power analysis starting from the HDL sources until the generation of power reports and power database for further analysis.

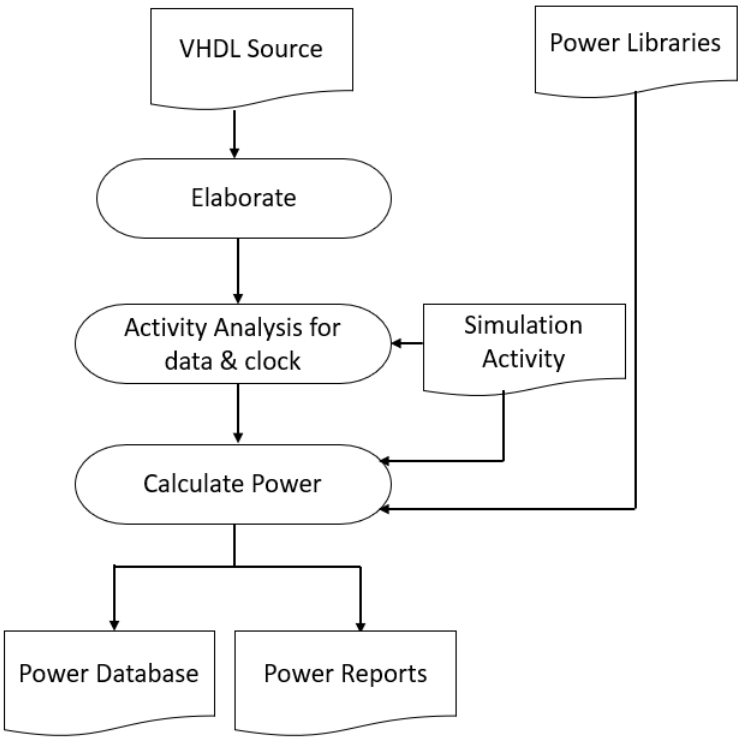


Figure 4: Power analysis flow

There are two types of dynamic power; the switching power which is related to the nets of the design and depend on the capacitance that calculated from the wireload model at SPEF file or liberty file, the supply voltage, the operating frequency and the switching activity, the internal power is related to the power of cells which depend on the input transition and the output capacitance and is mainly library based. Furthermore, we have two types of matrices; the first one is calculating the internal power from the load capacitance and the input transition time and the other matrix gives values for the transition time or the slew time for the upstream cells from the input transition time and the load capacitance. This is the mechanism of how PA is calculating the transition time for every cell and and giving number for the internal power from the first matrix. We will explore all the steps at RTL power estimation as following:

Inputs PowerArtist has some inputs like any EDA tools in order to perform the function that is designed to perform. The first input is the RTL or gate level design files, the tool could support multiple types of HDL sources or mix of them. The second input is the dumping file or the simulation file, this file is representing the activity of each net, cell and pin inside the design and how the value of this element is changing and toggling over the time, the tool could support multiple types of activity files such as FSDB, VCD and SAIF. The rest of the inputs are the power libraries, liberty files which define the power characteristic for each cell, the technology used, clock gating style, the top module, the top instance, clock and data buffering style, the net capacitance, clock frequency and clock definition.

Elaboration The first step of power analysis is design elaboration. PA tool could elaborate any type of HDL source files (Verilog, VHDL, System Verilog,....) to generate an internal binary format which called the scenario file (.scn). This step is creating the schematic for the design in order to do all the power calculation adding all the elements and parameters after later. The efficiency of this step is moderate, it's not as accurate as the normal synthesis tools like DC as it elaborates the design with only cells that have two inputs, however it maps the real design with a good approach to the real netlist. Design compilation, optimization and technology mapping are executed at this step.

Generate activity waveforms The second step is analyzing the activity file which generated from the simulation process or the given scenario to produce a file called GAF file which describes the activity of each net and cell inside the design. At this step, the activity factor is set for all the nets and you should check that there are no simulation data missing for some nets especially the power critical nets and the time of the signal when it was in the X or undefined state from the simulation file from the log files generated by the tool.

Calculate the flop clock activity At this step, the tool creates the clock tree for the design and estimates the activity of the clock pins inside the registers in order to identify the register and clock activity.

Calculate power The final step of power analysis which could be categorized into two types: the first one is to perform average calculation for the total power and the second one is to perform time based power analysis through picking up certain window and observing the power consumed at this interval. We usually care about the average power as we could pick up the desired window either by looking at the trace file that has the all the instruction executed at the simulation or perform activity analysis at PA to identify the window which has very high peak activity or power consumption.

Output One of the main advantages of PA tool is the variety of the information and reports we could obtain. For example, power summary report could be generated to identify the detailed power categories such as leakage, internal and switching power for all the instances up to any level of hierarchy. Moreover, average power report summarizes all the design's power of all components, gates, memories, buffers and clock tree. The data about the elaboration and optimization for the design could be also reported by the tool along with the simulation data and clock gating signals. The most important output is the power database which could be loaded on the tool and viewed with the GUI for power debugging and investigation which made it more usable and flexible to access all the data which is one of the most important key features of PA. PowerCanavas is the graphical user interface for PA, the power database created by the tool in a binary file could be viewed which contains a hierarchal netlist for the design, power analysis and reduction information.

4.1.2 PACE model

Power estimation at the early stages of the design is very bottleneck and could be important factor at the design decisions. However, low technology process increases the complexity of power analysis and widen the gap between the RTL power estimation and the netlist, in another meaning, it enhances the influence of physical design on the power especially the clock power. This led to a critical issue that should be taken into consideration such as the variation between the power of the RTL stage versus the gate level netlist, in addition the power optimizations at the RTL level which could be vanished at the physical implementation due to the growing gap between the two power perspectives.

PA introduces one of the advance technologies that takes into consideration the power estimated for clocks, glitches and wire capacitance. This technology is called PowerArtist Calibration and Estimation (PACE) which provides mapping between the RTL power and gate level sign-off power by extracting some parameters which included at the power estimation of the RTL. PACE model creates feedback loop between the RTL design and physical implementation to add more accuracy to the power estimation of the RTL. The calibrated data from the synthesis and back-end characteristics includes the following: percentage of different types of Vt cells which impact

the leakage power, the clock tree distribution and clock buffers. Moreover, the wireload model is extracted from parasitic file SPEF is added to the model. Figure 5 shows the flow of PACE model; the flow is similar to the normal flow of RTL power estimation, the only difference is that PA reads post-layout to estimate clock tree models and capacitance. PA elaborates the gate level netlist and adds some parasitic from SPEF file in order to obtain net capacitances for different type of cells, the cell distribution and the technology used to generate Pace model file which will be used on RTL design for power analysis for accurate power numbers. The pace file is containing data about the types and number of cells such as memories, registers, clock tree, buffers and macros. We will see later all types of information we could extract from the PACE model file.

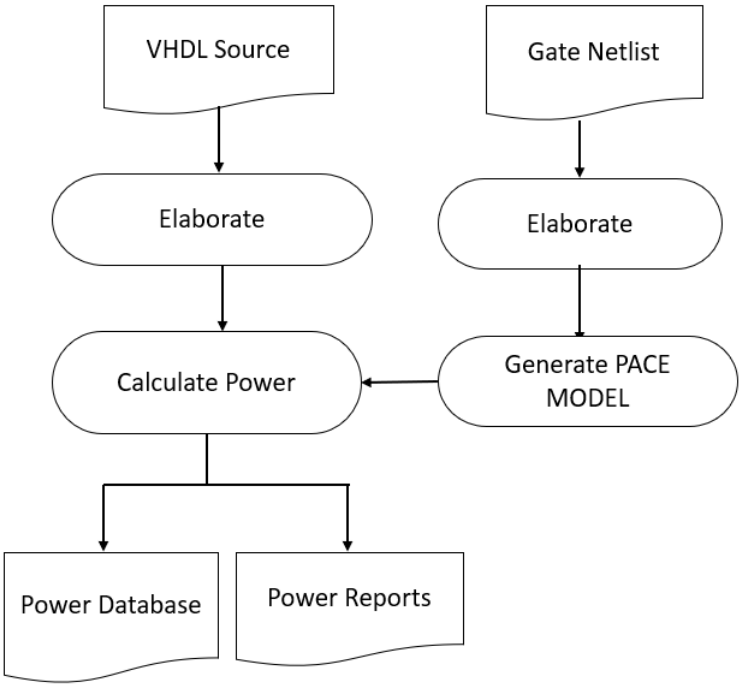


Figure 5: PACE model flow

4.2 Joules

Joules RTL power solutions and estimation[**3**] is one of the tools supported by Cadence that could estimate the power at the RTL level providing high estimation of the gates and wires power that arise from the strong capabilities of Cadence backend tools which gives real figures for the power on more well elaborated designs. The basic flow of Joules tool is close to PA, so the main steps are the same except that as we mentioned before, Joules has the capacity to elaborate the design with more accurate way that PA which gives more accurate power estimation as we will investigate at the measurements of power with both tools. The tool incorporates the prototype of the design from Cadence Genus synthesis solution that could elaborate and analyze designs up to 20 million instances with accuracy close to signoff. The setup of the tool is simple due to the fact that you don't have to put all the parameters manually, it uses the SDC files which is already exit for normal synthesis tools such as DC which saves the effort to setup the flow manually.

4.3 PowerPro

PowerPro[**5**] is a power analysis tool for RTL power estimation and optimization that provided by SIEMENS company. The basic flow for the tool is like PA except that there are few optimization techniques introduced by

the tool such as observability and stability techniques. The power reduction technique which is observability is similar to ODC technique in PA, on the other hand stability is another power reduction technique which is based on scanning the design searching for conditions at which the data is stable when the clock is still toggling or the reverse situation, So we could say that it combines LNR & LER power reduction techniques under the same category. The tool is replacing the elaboration step performed at PA & Joules tools with a step called design prototyping which affected the results for power estimation as we will investigate. PowerPro provides information about what's called sequential and combinational redundancy highlighting any sources of power wastage due to unnecessarily toggling of any signals. The accuracy of power estimation for this tool is highly affected by how the design is built which shows how accurate the combinational part is elaborated. The graphical user interface of the tool is quite helpful for investigation and debugging.

5 Power reduction techniques

Power reduction for the RTL design could be necessary at the cases when the power budget of the design exceeds the specifications of the design. PA tool is one of the tools that performs power optimization at the RTL level by different power reduction techniques that scan the whole design and identify some power bugs in order to reduce your chip's power. These techniques are based on modules called PowerBots which scan the design and searching for some features such as non-enable registers which caused wasted toggling of clock or clock toggling with unobserved data as we will see at the following sections. PowerBots analyze the design and the simulation activities on different nets and propose modifications that make your design consumes less power. It locates the power bug, gives number for the power wastage and power saving and finally reports recommendations that could save power. There are many PowerBots at PA tool that are available on the tool. We will investigate some of the power reduction techniques introduced by PA related to different categories of the design such as registers, memories and combinational part as it will be discussed at the following sections:

5.1 Low-Activity Non-Enabled Register (LNR)

LNR PowerBot is one of the reduction techniques that are available on the PA tool that finds all the registers which are not enabled and their inputs are low activity and not change frequently. It estimates the amount of power saved by introducing an enable signal that could detect the change of the inputs of the registers. The following schematic at Figure 6 shows the unnecessary toggling of the clock while the data is inactive for a long period of time and this leads to unnecessarily recirculation of the clock signal inside the sequential elements and the upstream logic while the data input remains stable. The power is wasted due to the clock signal that access the register although the data input is not toggling. So, driving the clock for this register causes power wastage while the circuit's behavior is not changing. The LNR PowerBot spots these low activity non enabled registers by analyzing the simulation activity of these nets and the probability of changing the data. It reports the power saved by inserting an enable signal for these registers and the area overhead.

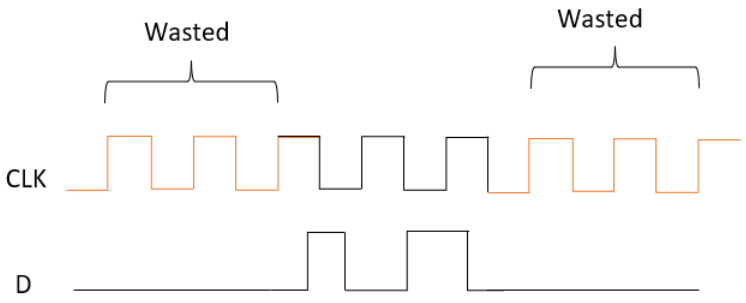


Figure 6: Timing Diagram - LNR

One way of constructing an enable signal is by using XOR gate which detects the change of the inputs of the register. The inputs of the XOR gate will be the current state and the next state of the register, if the next state is the same as the current state, there is no need to change or toggle the clock. If the bits of the state are changing between the two states, then the register is enabled, the clock signal passes the data between the two states. This driven enable signal is clock gated in order to have a free glitch clock signal that access the register only in case of changing of the data input. The following circuit at figure 7 shows the proposed implementation of the LNR techniques by XORing the next state and current state of the flipflops to generate an enable signal for the clock gating cell to detect if any bit is changed.

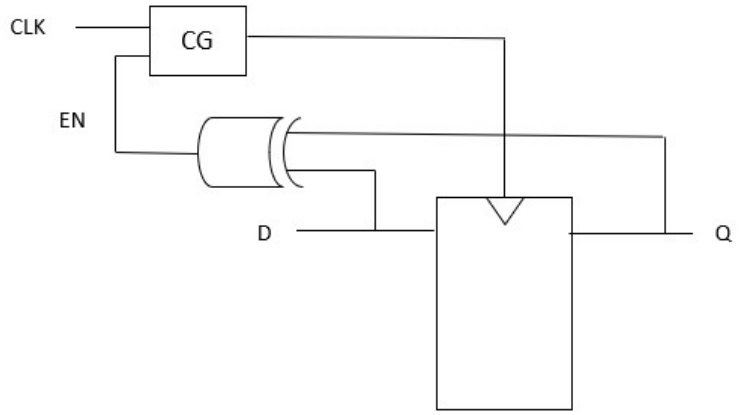


Figure 7: LNR - Circuit Diagram

This technique has a significant trade off which is the large area of adding XORs and ORs gates. Moreover, adding some logic between the clock bus and the registers could have a timing impact especially if there is a change of the data input that comes later and requires some time to generate the enable signal before the clock arrival. These impacts could be overcome by using human knowledge by locating an existing signal for the candidate registers, for example if we talk about a design that has multiple modes or shared modules, we could use the signal that controls each mode or chooses specific modules to enable the register which is only active at this time or mode and no need to construct non smart enable signal with a lot of timing and area cost. This technique is not recommended at the complicated design that has inputs with large bit width because it's a scenario or test dependent and is depending on the current activity of the bus which may change with different scenarios.

5.2 Low-Activity Enabled Register (LER)

This type of reduction technique is like LNR PowerBot. However, LER registers have an enable condition which is already exist. The techniques tried to improve the existing enable signal by generating extra enable signal like LNR which improves the clock gating efficiency and saves some power. The low frequency registers is identified as the registers that have less than 5% activity which is set as a default from the tool and could be changed by the user. This technique has the same implementation, idea and impacts of LNR technique, the only difference as we mentioned above is that the candidate register has an existing enable condition. The tool estimates the efficiency of the clock gating signal by measuring the CGE which is the ratio between the number of clock cycles when the register is at idle state over the total number of clock cycles at the simulation window.

5.3 Observability Don't Care (ODC)

The ODC PowerBot is one of the most effective power reduction techniques. The basic idea of this technique is that the output of some registers is not observed at some conditions for the upstream registers. The PowerBot

is scanning the topology of the circuit in order to generate an enable signal for the downstream registers under conditions which the output of these registers is not observed by the upstream registers. The ODC PowerBot is going through the design to locate the registers that are not clock gated and examine the candidate registers in order to verify the conditions under which the output of these registers is not observed due to the enable condition of the upstream registers that blocks the output from being observed. If the condition of the technique is met, the candidate registers have the ability to be enabled under specific conditions which the output is meaningless to the upstream. The technique should be implemented well as the area and the delay of the topology may be affected, also the correct timing of the driven enable signal should be taken into consideration in order to sample the correct data from the candidate register. Moreover, when the data width of the candidate register is too large, it happens that this technique become less effect in terms of area because it requires more clock gating cell that added more clock power to the circuit.

5.4 MUX Power Linter (MUX)

The MUX PowerBot is one of the power reduction techniques introduced by PA that scans the design to locate all the multiplexers and follows the inputs of these multiplexers. It may happen that one of the inputs of the multiplexers is toggling or performing some computations through the logic and finally, the output of this signal is not selected by the multiplexers which is considered as a power wastage. The technique estimates the amount of power that wasted due to the unnecessary toggling of one of the inputs and the amount of power consumed at the upstream for this input. Figure 8 shows the signals for a multiplexer which the selection signal is choosing input B while input A is consuming power by transitioning and not selected by the selector signal, the power consumed when the input is toggling while not propagated to the output of the mux is considered as wasted power.

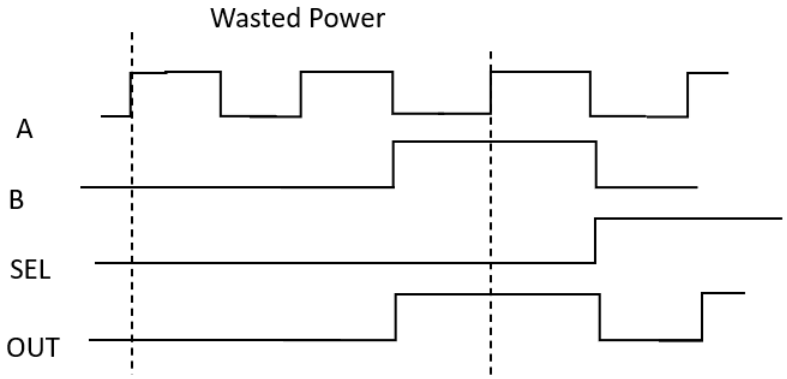


Figure 8: Timing Diagram Mux

The unnecessary toggling of one of the inputs could be desirable or undesirable, it depends on the design and the required functions. The implementation of this technique is quite simple, however it requires better understanding of the data flow, the functionality and the timing requirements. MUX PowerBot points out the part of the design that could be candidate for this technique and gives information about the amount of potential saved power. The implementation of this technique is design and activity dependent such that if one of the inputs has high activity without selected by the upstream design, it has been considered as a potential power bug and could save power.

5.5 Gate Memory Clock (GMC)

The GMC technique is one of the techniques introduced by PA tool that identifies the power bugs for memories such as RAMs, ROM, single port register file or a two-port register file with memory enable and write enables. This technique identifies the redundant read or write memory accesses and memory accessing while the enable or

accesses enable pins are not activated which result at wasted power due to redundant read or write the same address and same data while the clock of the memories are toggling and consuming power. The technique is searching for a way to disable the memory clock for redundant read or write accesses by two ways: the first way is to implement a circuitry that detect the change of the data that comes out or into the memory which is similar to LNR & LER implementation, the second way is to look at the downstream cone of the design in order to search for observability don't care (ODC) opportunity for the read write signal. This technique is also scenario and design dependent because the redundant read or write accesses could be scenario dependent and if the data is changing, this issue is no longer considered as a bug and we will add area and complexity to the design while we don't save power at the realistic or different situation, so it should be handled carefully with such a way that different scenarios and test cases are used to evaluate the effectiveness of this techniques. Finally, the tool should be provided by memory script that define all the memory types inside your design with all the information about all ports such as memory enable, access enable, read or write address and the data ports, so that the tool could identify the memory cuts and propose an idea about the power bugs inside the memory.

5.6 PRISM

Prism PowerBot is scanning the design looking for specific pattern or scenario at which there is a chain of registers that are dependent on each other with a way that the early register of the chain is enabled while the other registers in the chain are not enabled. The early register of the chain could be used for clock gating the following registers at the chain. The Prism PowerBot is trying to find candidate registers in the design that are clock gated and determine if the upstream registers could be clock gated by the candidate register. This technique is very efficient at the designs that have long and large pipeline stages with such way that every pipeline stage is depending on the previous stage so that we could use the output of each pipeline stage to clock gate the following stage as we will see later at the thesis with a real application. The technique is evaluating the power saved and drawbacks of this technique such as the effectiveness of the clock gating.

5.7 Clock Enable Condition (CEC)

The clock enable condition linter PowerBot is the type of the power reduction techniques that is looking for registers and multiplexers in a feedback loop at figure 9 in order to introduce a clock gating opportunity where the input of the flip flop is not changing while the clock is enabled. This technique is basically aiming to identify enable conditions that are not optimally designed which leads to power wasted due to poor enables. This technique is very effective at the mux-flop feedback loop topology which will be a candidate for clock gating opportunity such that the clock gating condition is driven from the mux selection and if the output of the registers is not changing as it has been recirculating, the clock will not toggle and therefore will not contribute to wasted power. The implementation of this technique is depending on the design and simulation such that the choice of the enable signal will not affect the function of the circuitry and add more power to the clock while this signal has high activity and inserting clock gating condition will not improve the power.

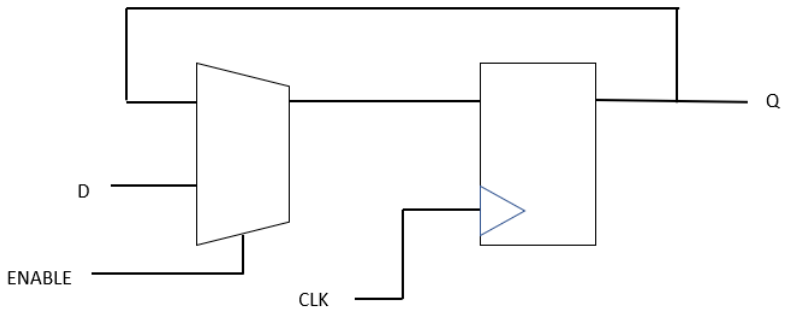


Figure 9: multiplexer in a feedback loop diagram

6 Methodology

6.1 Assumptions

This work aims to estimate the RTL power consumption of Kalray's MPPA cluster for CV1 & CV2. Early RTL power estimation will be conducted by multiple tools from different providers. CV1 was taped out two years ago which means that the final layout and the real power estimation on Silicium is already exit. On the other hand, we should keep in mind that CV1 & CV2 are the same technology, however we mentioned before that there are a lot of differences between the two releases in terms of the performance and memory size which makes CV2 more power and area. The backend tools for the two releases are different; the backend team used Nitro tool provided by Mentor for PnR of CV1 in addition to more LVT cells at the design to meet the timing of the design. However, the PnR of CV2 is conducted by Innovus Cadence which is more accurate regarding the timing and density of the design such that the designers will not have to use more LVT cells to meet the timing of the design and the tool will do the job accurately and precisely. This difference at the Backend side for the different versions will be more visible at the leakage current of the design and the cell selection. The final layout of CV2 is not ready yet which will be taped out at the end of this year, however the final layout for the processors of CV2 is almost finished with a good percentage of annotated information. Furthermore, the implementation of different power reduction techniques is introduced in the design of CV2 to reduce the power consumption for different scenarios. The goal of this methodology is to investigate the methodology for early RTL power estimation, give the way to improve the results by extracting some parameters from the netlist and implement different optimization techniques on CV2. The work will consist of:

1. Explain the RTL power estimation flow.
2. Estimate the RTL power consumption for CV1 & PACE model.
3. Compare between different power tools on the processors of CV2.
4. Implement power reduction techniques on CV2
5. Improve correlation between RTL & Netlist

6.2 Approach

The first part of this project is to estimate the RTL power consumption of CV2 & CV1. As we mentioned before, the design is mainly consisting of five clusters; every cluster has 16 processors which are the computing engines. First, we will calculate the power at different scenarios. There are many scenarios or test cases that could be executed on the cluster or processor level which represented different kinds of instructions that could be executed on the design at real applications. The first scenario is called TCA_FP16 scenario, is a coprocessor scenario which represent the multiplication of two matrices each one is 4 by 16 and could be represented by the mathematical equation: $M_A x M_B += M_C$. This test case was not supported at CV1 and is one of the new features of CV2 and the coprocessor is the part of the processor core which executes this instruction. We need to know that this scenario is using 4 pipeline stages to be fully executed and a lot of multipliers, shifters and adders which make it the most power consumable scenario as we will see at the results. The second scenario is TCA_UINT8 scenario, also coprocessor scenario which responsible for the multiplications of two matrices each one is 4 by 8 and could be represented by the mathematical equation: $M_A x M_B += M_C$. This test case was supported at CV1 & CV2 which will help us to have a clear comparison between the two versions of the design under the same test. This scenario is mainly executed at 2 pipeline stages with less amount of combinational compared to FP16 scenario. The two remaining scenario are mainly developed by the software team in order to evaluate the power reductions at the RTL level; the first scenario is called Random_Bundle which is random instructions executed by the processors and the second scenario is called Tiny ALU scenario which is addition instructions performed by the ALU inside the processors, these two scenarios are mainly dumping or activating the processors only not the whole cluster so that we will judge the power optimization for these two scenarios at the processor level only as we will Investigate later. RTL power estimation is mainly two types of calculation: time-based power analysis which estimate the

power consumption of the design at different time which allows you to obtain power waveforms as a function of time for RTL and netlist, the other type is average power analysis over specific time interval. The selection of time window could be through the trace file generated from the simulation which has the list of instructions executed at the simulation time, you should understand the scenario in order to pick up the right time window when the highly power consumed instructions such as the multiplication ones are performed. Time based power analysis could be also suitable choice to specify the highest power peaks over the simulation time to calculate the average power for such peaks. The start and end time of average power calculation is very crucial issue to get accurate power estimation. The iterative process for accurate RTL power estimation is listed below:

1. Specify the top module and instance inside the design with all the files to be compiled.
2. Identify the clock signal, frequency and clock gating style.
3. Elaborate the design and provide the tool with activity file.
4. Analyze the power scenario to pick up the interested time window.
5. Run the average power analysis to get RTL power estimation.
6. Generate PACE model from the gate level netlist and SPEF.
7. Use PACE technology file to add calibrated data to get accurate power results.

7 Implementation of power reductions

In this section, we will investigate the implementation of different RTL power reduction techniques on CV2. The main objective for the company was to optimize the power consumed by the clusters at different scenarios by located the power bugs inside the design by the aid of different tools and fix theses bugs at the RTL level. at the previous sections, we identify different types of power reduction techniques on PA. The power optimization of Joules cadence was validated previously by the designer; however they didn't get positive feedback from the power optimization of this tool which failed to identify and locate expected power optimization opportunities which were expected to be discovered, at the meantime we will not try to setup the power reduction flow for Joules Cadence. PowerPro which is not an accurate tool at the power estimation at the RTL shows a great potential for power reduction techniques as we will show. In general, PA showed a great performance and productive results in terms of power reduction techniques compared to Joules and Powerpro. In the following section, we will explore different cases for power bugs inside CV2 which reported by the tools or discovered by the investigation of power figures for different hierarchical levels of the design at different power scenarios.

7.1 Clock Gating

Clock power is one of the main contributors to the total dynamic power consumption of the design. Clock gating is the most efficient power reduction techniques that decreases the dynamic power consumption. This technique is implemented automatically with all the synthesis tools. Occasionally, some synthesis tools are not able to insert or detect the clock gating conditions for some registers due to the complexity of the code or optimization algorithms conducted by the tool. In this section, we will explore the importance of clock gating and how we could implement clock gating for some registers in order to save power and cut down the unwanted toggling of some inputs of unused registers.

Due to the complexity of the HDL codes some synthesis tools couldn't insert automatically clock gating for some registers during the elaboration due to unclear conditions, there are two cases that happen at CV2 when the power tools couldn't detect a clock gating condition for the registers during the elaboration before power computation: the first case was due to variable data type which could result in a complex code and the second case was due to a clock gating condition that was a For loop statement. The insertion of For loop and variable inside the If statement

at VHDL makes difficult for some synthesis tool to detect the clock gating condition, so it's always recommended to look at the netlist to see if the tool could insert the condition or misunderstand the inserted condition.

One way to insert a clock gating for some registers is by ODC, which mainly based on looking at the upstream of the logic to understand when the output of some registers is not observed due to a specific condition that applies to the output. For example, assuming that the output of a register is propagating to one input of a multiplexer which has a selector signal, this signal could be a candidate condition for the clock gating of the register because if the signal does not select the output of the register, it will be not observed and propagated to the following logic, so we could save some power by using the selection signal as a clock gating condition for this register. This is an important technique which should be implemented to reduce power based on good understanding on the hierarchy of the design and data flow. We could eliminate the propagation of inputs and the power consumed for calculation if we know that the output observability is based on some conditions which could be used as a clock gating conditions for the input registers.

Clock gating cells are added to the clock signal in order to save the power consumed by the registers under some conditions. However, we should know that these cells have input capacitance, so they increase the power needed from the clock signal in order to drive theses cells. The clock gating style inside the design is very crucial issue as it should be handled carefully in order not to increase the power of the design. The estimation of the optimal number of clock gating cells could be conducted by the power tools in order to know the threshold of clock gating insertion. In the following section, we will see how the power tools calculate the limit of clock gating insertion and determine the limit at which adding more clock gating cells will not improve the power anymore and it will add more driving power to the clock tree.

7.2 Mux Technique

One of the most effective optimization techniques is MUX power optimization technique. As we mentioned before, this technique scans the design to locates all the multiplexers and follow the inputs of these multiplexers, it might happen that one of the inputs of the multiplexer is toggling or performing some computations by a specific module and at the end the output of this module is not selected by the multiplexer which is consider as a waste of power. In this section we will investigate one of the most significant power bugs that was found at the design of CV2 which great amount of power is consumed at the coprocessor inside the processor core. The coprocessor consists of many building blocks which execute different instructions on integer and floating points data type; the first block is BILAU which executes instructions of integer data type at two pipeline cycle, the second module is BLAU which executes instructions on floating point data. Figure 10 shows the architecture of the coprocessor; the inputs are propagated to both modules from the same registers at the same time to the two modules. We talked before about TCA_UINT8 scenario which perform integer data instructions and TCA_FP16 scenario which perform instructions on floating point. If we are going to use one scenario, the corresponding block will operate normally while the other will be recirculated as it's useless to operate. For example, if we are going to use BLAU module, E2 flops at BILAU will be recirculated and no data will be propagated to this module and vice versa. The power bug was discovered when we simulate TCA_UINT8 scenario which should dump BILAU module only, the other module which is BLAU is consuming more power, the reverse situation also happened for TCA_FP16 scenario which should dump BLAU module only, however the module that dedicated to integer data type is also consuming power.

This power bug should be discovered by PA by MUX technique which scans the design to search for specific modules which consume power and the output of these modules are not selected at the output of multiplexer. After all, PA failed to highlight this obvious power bug by any other techniques although the conditions to locate this power bug should be clear for the mux power reduction algorithm, moreover there are a lot of multiplexers between E1 & E2 stages which could be a good indication for the optimization techniques. This case was reported to the R&D of the tool in order to debug this issue. On the other hand, PowerPro tool which was not accurate at RTL power estimation managed to point to this significant power bug by stability technique.

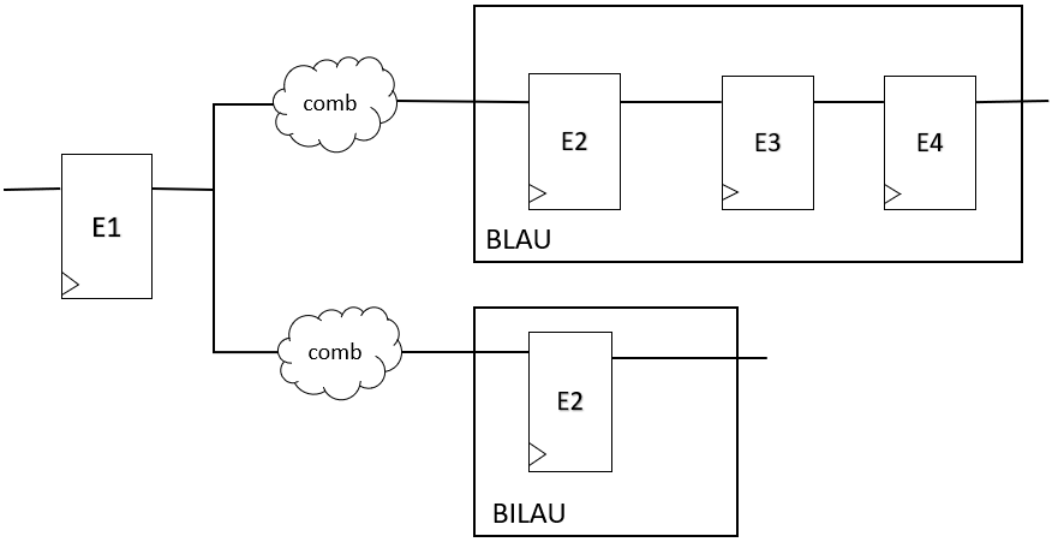


Figure 10: Coprocessor architecture

The modification for this power bug was implemented by identifying and understanding how the instructions are executed and the number of cycles at which every module is used and if these cycles are regular or irregular in terms of continuity. There are two ways to solve this bug: the first solution is to duplicate the data flipflop at stage E1 which is responsible for the data propagation and connect each flops separately to every block such that we could guarantee there is no datapath to the block when the block is not used and there is no power consumed at all, however this solution will add a significant area overhead as it will add 2048 input flops and the utilization and density factor during PnR will be a big issue. The second solution to insert multiplexers at the datapath for these two main blocks so that we could select the inputs to pass or zero to pass when the block is not needed. The mux insertion could be a perfect solution that perfectly matches this case as we know that the execution of any instruction will last for more than 100 cycle, so the input will not switch between zero and real data every clock cycle otherwise we will add more power due to this toggling or switching. By solving this issue, we could save power about 33% from TCA_UINT8 scenario and 18% from TCA_FP16 scenario. It was the most effective modification for power optimization at this project for the most power consumed scenarios.

7.3 PRSIM

Prism PowerBot is scanning the design looking for specific pattern or scenario at which there are chain of registers that are dependent on each other with a way that the early register of the chain is enabled while the other registers are not enabled. This type of technique is mainly useful for the design which has multiple stages of pipelines, the technique is trying to locate the chain of registers and identify possible clock gating condition driven from the previous stages. Some tool cannot propose modifications for this complicated situation, however it refers to the part of the circuit that is a potential candidate for such thing, so that the designer could redesign or think how to modify the code. The designer could drive an enable condition or strength the existing enable condition. The following case from CV2 is a module which is 14 pipeline stages was highlighted from PA tool as a potential candidate for PRSIM power reduction technique, however the tool has not proposed any idea how to fix this power bug. Figure 11 shows a simplified piece of code of how the pipeline stages are coded. The left code shows a power bug at which will have these situations: when there are valid inputs at the first stage the data will propagate to the second stage which will be turned on by the validation condition that propagates from the first stage, then the data will be propagated to the third stage and so on. This style of coded put a specific condition to turn on the following stages, however it didn't take into considerations the case when the pipelines stages will be switched off due to the invalidation of new operands, this style will turn on all the pipeline registers once they operated and they will be kept active as

long as the power supply is on. This power bug is mainly due to the register that keeps the clock gating conditions which is e_invalid_op variable register at which if there are no inputs valid coming from the first stage, the second stage will be switched off, however the following stages will operate normally as the condition of invalid operands didn't propagate to the upstream registers due to the clock gating condition. This power bug is solved as shown at the right piece of code by putting the invalidation operands condition outside the clock gating condition so that the new condition will be free to propagate to the upstream registers in two cases of existing or non-existing of valid operands which will update the status of the pipeline stages every clock cycle.

```
entity sfu is
port (
  e1_invalid_op : in std_logic;
  Q4 : out std_logic_vector(63 downto 0);
  clk : in std_logic;
);
end sfu;
architecture rtl of k1fu is
---E2
  process(clk)
  begin
    if ((clk'event ) and (clk = '1')) then
    if (e1_invalid_op = '0') then
      Q2 <= Q1;
      e2_invalid_op <= e1_invalid_op;
    end if;
  end if;
end process;
---E3
 process(clk)
  begin
    if ((clk'event ) and (clk = '1')) then
    if (e2_invalid_op = '0') then
      Q3 <= Q2;
      e3_invalid_op <= e2_invalid_op;
    end if;
  end if;
 end process;
---E4
process(clk)
  begin
    if ((clk'event ) and (clk = '1')) then
    if (e1_invalid_op = '0') then
      Q4 <= Q3;
      e3_invalid_op <= e4_invalid_op;
    end if;
  end if;
end process;
```

(a) before

```
entity sfu is
port (
  e1_invalid_op : in std_logic;
  Q4 : out std_logic_vector(63 downto 0);
  clk : in std_logic;
);
end k1fu;
architecture rtl of k1fu is
---E2
  process(clk)
  begin
    if ((clk'event ) and (clk = '1')) then
    if (e1_invalid_op = '0') then
      Q2 <= Q1;
    end if;
    e2_invalid_op <= e1_invalid_op;
  end if;
end process;
---E3
 process(clk)
  begin
    if ((clk'event ) and (clk = '1')) then
    if (e2_invalid_op = '0') then
      Q3 <= Q2;
    end if;
    e3_invalid_op <= e2_invalid_op;
  end if;
end process;
---E4
  process(clk)
  begin
    if ((clk'event ) and (clk = '1')) then
    if (e3_invalid_op = '0') then
      Q4 <= Q3;
    end if;
    e4_invalid_op <= e3_invalid_op;
  end if;
end process;
```

(b) after

Figure 11: modification of RTL to optimize power consumption

8 Results

The following section will describe the results obtained for the power analysis of Kalray's MPPA CV1 & CV2, also the calibration between the netlist & RTL in order to improve the accuracy of the results. The performance of the three power tools has been investigated on the processor level to evaluate the accuracy of the estimation for each tool. Furthermore, the power optimization and clock gating efficiency of CV2 are explored at different scenarios. Finally, the deviation between the power tools and the Back-end side is discussed to understand the parameters that contribute at increasing the gap between the RTL & netlist power estimation.

8.1 Power Estimation of COOLIDGE V1

The first part of this project is to measure RTL power estimation for CV1 on PA. In the previous sections, we mentioned the difference between CV1 & CV2 in terms of performance and memory size, this version of cluster could only support TCA_UINT8 scenario which will be used to evaluate the power consumption. CV1 was taped out and released in 2019 which means that we could have actual power estimation for the final product on Silicium. At this section we will measure RTL power estimation of CV1 on PA and how to improve the power estimation with the PACE model at PA. At this part of the project, we will have the advantage of comparing the measurements from the tool with actual power estimation as this version of the chip is already at the market.

The RTL power estimation of CV1 is described at figure 12. The total RTL power estimation is 4,61 W for each cluster, PA shows the power consumption for all the components and gates at the design to give an indication which part of the design is consuming most of the power. As described in section 4.1.2 the PACE model is extracting some parameters from the netlist such as clock tree, cell distribution, the technology parameters and wireload model . The power measured at PA with PACE model was 12.04 W for each cluster. The power measurement of CV1 on the Silicium has shown that each cluster is consuming 14.10 W. The power estimated with the PACE model for CV1 is about 17% different from the real results obtained on Silicium which means that PACE model increases the accuracy of measurements of the RTL power and improves the estimation close to the reality as expected. The large gap between the RTL & Silicium measurements is due to the complexity of the design which increases the differences between the physical implementation of the power tools and real implementation at the final layout.

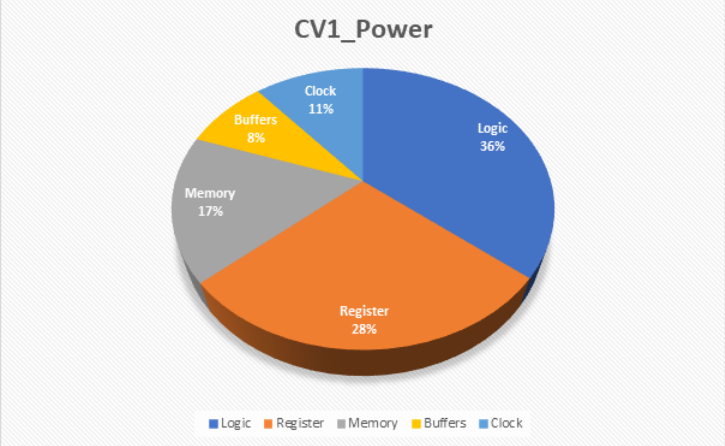


Figure 12: CV1_Power

8.2 Power Estimation of COOLIDGE V2

The next part of the results is mainly concentrating on the RTL power estimation of the processors inside CV2 which is currently at the stage of development and will tapped out at the end of the year. We managed to estimate the RTL power consumption of cluster level at different scenarios on PA, however we couldn't do the same on Joules & Powerpro due to the lack of time. Before that, we should take into consideration many factors; first, the final netlist of the cluster is not ready in terms of the PnR step which means that we cannot do PACE model with the current netlist as the percentage of annotated nets from the netlist is very low however, the netlist of the processor inside the cluster is well done until the post route step which means that we could extract the needed parameters for the PACE model for only the processor not the whole cluster. Finally, we should keep into mind that each cluster has 16 processors which consumes about 82% from the power consumed by the cluster. So, in our analysis we will measure the power of the processors at different scenarios with different tools in order to evaluate the difference between each tool and how they correlate to each other. Unlike CV1, the current version CV2 is supporting TCA_FP16 scenario which will be measured with the three tools: PA, Joules Cadence and PowerPro.

The measurements of power for the three power tools were conducted under the same condition such as the technology files, environment, simulation data and list of HDL files. Figure 13 shows the RTL power estimation of one processor at the cluster of CV2 for TCA_UINT8 scenario and figure 14 for TCA_FP16 scenario. The results show the different power components; combinational, sequential, memories, clock, and total power for the different tools. The results shows that Joules is giving the most accurate result which could be explained by the accurate elaboration step that performed through Genus synthesis tool which is more close to the real design. Powerpro has a low estimation of power results compared to the other tools due to the inefficient elaboration step which is replaced by rapid prototyping of the design which is not good representative for the real design. This explanation is supported by the estimation of combinational power which shows how accurate the elaboration step for each tool. PA shows a good power estimation close to the power obtained by Joules in spite of the difference at the elaboration steps at the two tools.

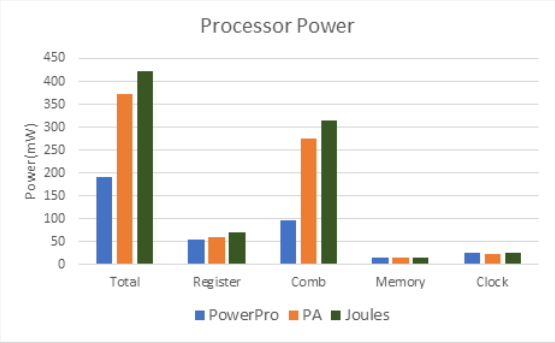


Figure 13: Processor's power at TCA_UINT8 scenario

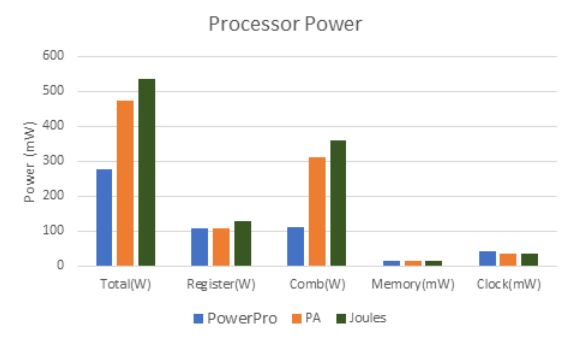


Figure 14: Processor's power at TCA_FP16 scenario

8.3 Clock Gating Efficiency (CGE)

Clock gating is the most effective way to reduce the power consumption by disabling the clock signal inside the sequential elements by inserting clock gating conditions. Clock tree is one of the most significant contributors for the total power consumptions due to the accuracy and low skew required for the clock signal. Inserting clock gating cells will reduce the power consumption of the register and upstream cones of these sequential elements, however, it will increase the load capacitance of the clock tree and effort to charge and discharge these nodes which will impact the clock power. PA estimates the power savings gained by the insertion of the clock gating and analyze the optimal number of clock gating cells to be inserted into the design. The number of clock gating cells could be specified by the minimum and maximum bit width for the sequential elements to be clock gated, in other words, we cannot insert clock gating for one bit register because the amount of power saved from clock gating the register

will be less than the power consumed by the register when it's free toggling. These parameters are specified at the synthesis tool like Design Compiler. At this part of the results, we will perform Local Explicit Clock Enable analysis on PA which guide the synthesis tool to include the optimal number of clock gating cells contribute to save power and minimize the clock skew.

The tool is performing some iterations by inserting clock gating cells inside the design and drawing a relation between the number of clock gating cells and the amount of power saving. The number of clock gating cells inside the design could be controlled by the min & max bit width for the register to be clock gating. Figure 15 shows the efficiency of clock gating for different scenarios which shows that there is a threshold for the clock gating cells insertion which there will be no improvements for the power after this threshold. Th most optimal number for the clock gating cells is from 70k to 80 k for the cluster at different scenarios. Moreover, the tool showed the efficiency of clock gating technique to save power consumption for the design. The amount of power saved by clock gating technique is about 44% from the total power consumed by each cluster.

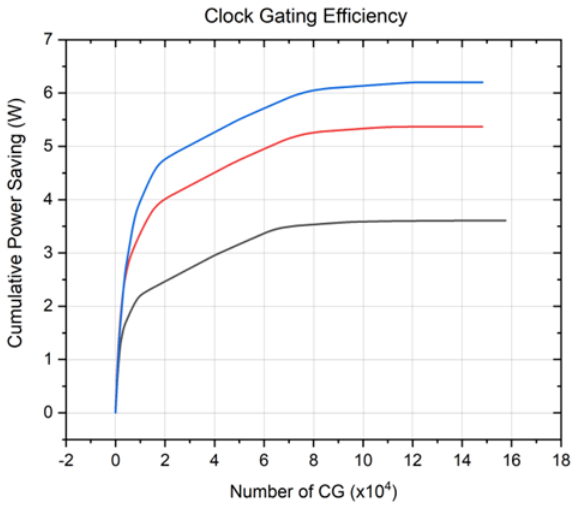
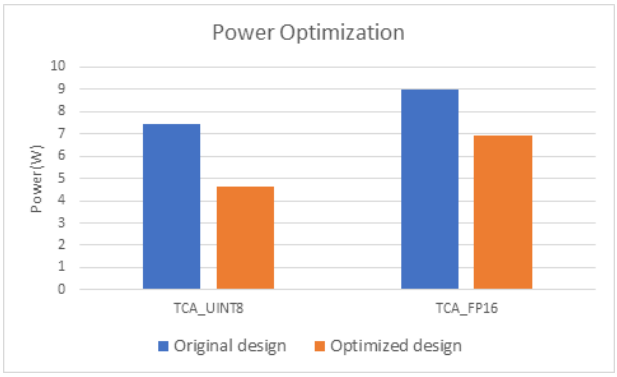


Figure 15: Clock Gating Efficiency for the cluster

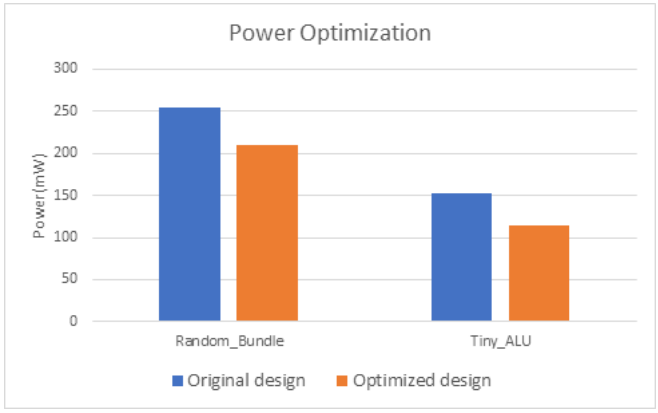
8.4 Power optimization

One of the main objectives of this project is to optimize the power consumption of CV2 especially for the two most power consuming scenarios at the coprocessor. The methodology was to measure the RTL power consumption before any modifications implemented to the RTL at different scenarios and compare the results with the RTL power estimation after the implementation of the HDL modification described at section 7. There are two scenarios which run on the processor level: Random_bundle test case and Tiny_ALU scenario, we will evaluate the power improvement at the processors' power consumption. We will explore the power optimization of the cluster power after modification for the two scenarios: TCA_FP16 and TCA_UINT8 which means that the modification of the HDL for shared memory(SMEM) will be evaluated at these two scenarios. The power improvement is estimated at the RTL level as we don't have final netlist in order to evaluate the final optimization at the layout.

Figure 16 shows the results for the power optimization for cluster of CV2 for two different scenarios, the results show a decrease of the total power by 37.7% for TCA_UINT8 scenario and 23.1% for TCA_FP16 scenario. The RTL power consumption of the cluster is reduced from 7,44 W to 4,63 W for UINT8 scenario and from 9,02 W to 7,2 W for FP16 scenario. The power consumed for the cluster of CV2 at TCA_UINT8 is equivalent to CV1 despite of the difference in area and performance for the two releases. the left figure also shows the RTL power optimization for the Random_Bundle and Tiny Alu scenario for one processor.



(a) Cluster



(b) Processor

Figure 16: Power Optimization of CV2

The MUX power optimization technique saved about 70% of the total power saved at the coprocessor scenarios, PRISM technique saved about 8% while the Clock Gating technique by insert new clock gating, driving new clock gating or fixing the clock gating condition saved about 22%. GMC techniques which is relevant to the power of the memory was introduced by the tool to save some power at the shared memory part due to the redundant read of the same address for more than one clock cycle, however the two scenarios are designed by mean to make all the processors are reading the same address at the same time to get maximum power consumption. So, it's not reasonable to modify the code based on scenario which is not possible to occur at the real scenarios especially for accessing the memory cuts.

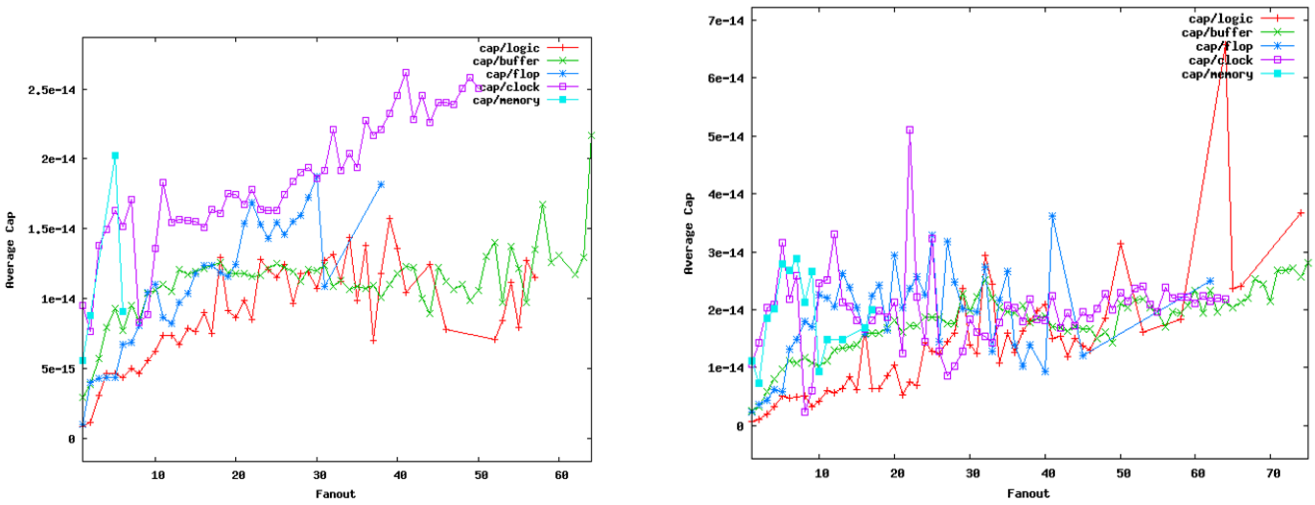
8.5 Netlist power estimation

Netlist power estimation produced the most accurate power results which conducted at the last development stage with real clock tree and cell distribution. However, Due to the lack of simulation platform on the netlist and the time consumed to get the final netlist, we cannot perform netlist power estimation at the early stages of the design. One of the main Inputs for any power estimation tools is the simulation activity file which should provide the activity factors and the operating frequency. The dumping file provides information about the activity for all types of nets and wires for the elaborated design, so the simulation should be run on the design which will be elaborated otherwise we will have a mismatch between the activity file and the elaborated design, as a result we cannot annotate the activity factor for some nets and the power consumption will be not accurate. In summary, we cannot run netlist power estimation with an activity file run on the RTL design and vice versa. So, we should ensure that activity file annotation is close to 100% as possible.

There are some factors that contribute at the deviation between the RTL & Netlist power estimation and widen the gap between them. The First factor is the number of data buffers, at the PnR step, designers have to add and insert some buffers at the path of data in order to meet the requirements of timing by slowing down or accelerating the driving capabilities of high fanout cell and reduce the load capacitance. The number of inserted buffers at this step is very large as the design gets more complicated, for instance, the number of buffers inserted by the power tool at the cluster level is about 1.6M buffers consuming about 800 mW, on the other side, this number is about 2.8M after PnR step which shows a great deviation in terms of power estimated for the data buffers at the cluster level. The second factor is the number of clock gating cells which is about 65K at the power tools and 77K at the postroute netlist. The third factor that has high impact at the inconsistency between the RTL and netlist is the type of cells, there are many cells inside the libraries that has different driving capabilities, logic cells with higher driving load consume more power due to the large load capacitance. The variation of cell characteristics could have significant effect on the total power consumed by millions of cells. one of the main components of dynamic power is

the switching power which is affected by the wireload model. The wireload model gives an estimation for the load capacitance versus the fanout which impacts the power required to charge or discharge these nodes, the wirload model for the RTL estimation is library based, while for the netlist power estimation is calibrated from the SPEF files that reflects the actual implementation. The diversity of the wireload model is the most important factor that wide the gap between the RTL & Netlist. Power tools are reducing this gap by extracting the wireload model from the SPEF files to be added at the RTL power estimation to improve the accuracy of the results. As we discussed before, PACE model is extracting some parameters which could be used to increase the accuracy for RTL power estimation. We are going to compare between the netlist for processors of CV1 & CV2. CV1 netlist was developed by Nitro Mentor and more LVT cells, while CV2 was developed by Innovus Cadence with less LVT cells, moreover both netlists are the same technology which is 16nm. The types of parameters we are going to extract are the leakage power and how the designers managed to improve the efficiency of the netlist's power by perfect PnR to reduce the capacitance for the netlist.

The amount of leakage power for CV1 netlist was 55.3 mW while CV2 netlist shows a leakage power of 32.6 mw. This result could be explained or justified by two reasons: as we know, LVT cells are more speed cells, however they consume more static or leakage power due to the low threshold voltage which means that more LVT cells in CV1 leads to more static or leakage power compared to CV2 as indicated from the netlist. Wireload model is used to describe the effect of fan out on the net capacitance, this model is either extracted from the netlist or the libraries. The wireload model extracted from the SPEF for CV2 shows a reduction of the load capacitance for all the nets by 25% compared to CV1. Figure 17 shows the relation between average net capacitance and fan out for all types of cells for CV1 & CV2 at processor level. The tool is performing some extrapolation to get the average value for the capacitance for every net to be added at the dynamic power calculation. Sometimes, this average value could be less than one which shows a great improvement to the dynamic power at PnR step.



(a) CV1 (b) CV2

Figure 17: Capacitance model for CV1 & CV2

9 Discussion

In this section, we will discuss the limitations, future work, results and the improvements of our flow.

9.1 Assumption

In this thesis, the methodology and power analysis and optimization for Kalray's CV2 design are investigated. The analysis is conducted under specific parameters such as the operating voltage, temperature, technology node and library files. The results obtained could be changed according to the design and parameters. The design is very complicated and large in terms of number of modules, combinational and sequential cells. Moreover, the final layout of the design is not ready to validate the accuracy of the power results obtained by the tools. The complexity of the design reduces the accuracy of the elaboration steps for the power tools such as Powerpro tool which failed to estimate accurately the RTL power for such big design. The efficiency of each tool is relevant to the complexity of the design, so it should be verified also to relatively smaller design with different parameters and technology.

9.2 PowerArtist tool

PowerArtist tool is the most wide spread and used tools at the Digital design teams that concern about power analysis and optimization. However, there are some shortages at the tool that were encountered during this project. Some proposed LNR modifications lead to an increase of the power consumed by the design which is mainly due to the large bit width of some signals, the tool introduced an idea to reduce the power by implementing the proposed circuit at figure 7 for some signals that are 512 to 1024 bit width. The implementation of this circuit is adding more XORs & ORs gates which leads to more power consumption not power reduction. The threshold number of bits for this technique is mainly depending on the design and the simulation activity, however the tool proposed modifications to reduce power that increase the consumption. The second issue was the power bug at the coprocessor that described at section 7.2 which should be detected by the tool due to the multiplexers that are located between E1 & E2 stages, this case is reported to the R&D of Ansys inc. This power bug is not identified by PowerArtist power reduction techniques especially MUX technique.

9.3 Joules tool

As we discussed before, Joules incorporates the prototype of the design from Cadence Genus Synthesis solution that can elaborate and analyze designs up to 20 million instances with accuracy close to signoff. So, the tool elaborates the design through Genus tool and the created netlist could be debugged and viewed only on this tool. Unfortunately, we don't have a license for Genus as the hardware team is doing synthesis step on DC. It was hard to view or debug the netlist created by the power tool before power computation without this tool. This was a very critical issue especially, when we got a power result that are not reasonable or meet the expectation, we need to look at the netlist to debug. CV2 is very large and complicated design and the synthesis tools may not perform the optimal design, so it's highly recommended to get license for Genus, if you are going to use Joules for the power estimation of complex designs In order to view and debug the netlist generated by the tool. We couldn't estimate the RTL power for the cluster due to the long runtime and the issue we described before to compare with RTL power estimation for the cluster with PA.

9.4 PACE model of CV2

The PACE model generates accurate results for RTL power estimation which reached to 17% of signoff power for CV1. The final layout for CV2 is under development and planned to be taped out at the end of year. So, we cannot perform PACE model for CV2 to get accurate results and validate the accuracy of our results, as well as the netlist of CV1 is developed by different PnR tools and different Vt cells so that it's not accurate and reasonable to use this netlist to do PACE model for CV2, this is left for future work.

9.5 Power scenarios

One of the main requirements for any RTL power estimation tool is the testcases or power scenarios. The power scenario represent the design activity and the state of every node inside the design. The scenario is mainly depending on the type of data, type of instructions and the time of execution. For the power optimization, there are many techniques called stability or data stable clock toggling which detect the low activity register that are not frequently

changing their values and could be a good candidate to be gated to reduce the power consumed by the sequential elements and the propagation of inputs. The lack of power scenarios for this project made it difficult to implement these technique as the inputs are dependent on the scenario which could be changed according to the testcase. I imagined if we have more scenarios, we could specify accurately the activity for all the registers and implement these techniques. Moreover, more power scenarios increase the opportunity to discover more power bugs at the circuit as they will dump different modules inside the design which highlight any power issues.

9.6 Netlist estimation

Netlist power estimation gives accurate results that is close to the sign off power as there is a real implementation about clock tree, cell distribution and cell capacitance. The front-end netlist which produced from synthesis tool such as DC shows more power consumption than the RTL with about 15 to 25%, however, the netlist from DC has not included the actual clock tree, actual number of data and clock buffers and the DFT part. The measurement of post route netlist power consumption has many requirements such as simulation platform to run the power scenario on this netlist in order to avoid any missing simulation data for some nets. This is left for future work to develop simulation platform for the postroute netlist.

Conclusion

The methodology and results for RTL power estimation and power reduction of CV2 using PowerArtist & Joules Cadence have been investigated at this thesis. The increased complexity of modern VLSI and the high demand of low power applications show the importance of early power estimation for the designers. The RTL power estimation tools reduce the time to investigate the power budget and make the designer able to take an early decision at the development cycle. The proposed design at this thesis is the current release of Kalray's processors which is more area, power, efficiency and memory size than the previous release. The power consumption for multiple scenarios has been investigated and optimized with PA to achieve power reduction of 25% in average which was a great achievement in terms of power consumption of the new version. The methodology of correlation between the RTL & netlist has been proposed at this work for CV1 to achieve accurate RTL power results. The evaluation and performance for three different power tools were explored on the processor to identify the drawbacks and advantages of each tool, as well as the steps of RTL power estimation. The RTL power estimation methodology has been introduced to give a reference for any RTL power estimation and optimization for any design at different scenarios.

Multiple RTL power optimization techniques have been implemented on CV2 and several power issues have been discussed to in order to explore different type of situations inside the design that cause power wastage and evaluate the efficiency of each power technique to save power. The power consumed of CV2 at TCA_UINT8 scenario has become the same as CV1 although the differences between the two versions in terms of area and efficiency which was a perfect achievement for the power of the new release. Multiple issues at the RTL power were explored, different clock gating styles have been proposed to reduce the unnecessarily toggling of some register and logics, the issues of coprocessors were investigated to get a better view about how to look at the power figures and explore the power bug without the aid of power tools.

The clock gating threshold is explored for automatically interference with the synthesis tools to better estimate the optimal number of the clock gating cells inside the design. The thesis investigates the power analysis of Kalray's design which is very large and complicated design, the evaluation and conclusions reached at this thesis could be totally different at small designs such as some optimization techniques which were not efficient for input signals that are 512 to 1024 bit width. The evaluation of different power tools is not absolute and could be relevant to the complexity of the design and the scenarios. PACE model improves the accuracy of the results by extracting some calibrated data from the netlist and SPEF such as wireload model, clock tree and cell distribution. The variation between the capacitance model for CV1 & CV2 has been investigated to show the improvement of the dynamic power at the netlist due to the PnR tool and Vt cells changes. At the end, the thesis described and explored efficient and practical methodlogy for RTL power estimation with improved accuracy with the PACE model, in addition to

different optimization techniques that are proposed and implemented to reduce the power consumption of CV2 at different scenarios to fulfill the power requirements for some applications.

References

[1] Ansys Inc. PowerArtist User Guide. Feb 2022.
[2] S. Nesset. RTL Power Estimation Flow and Its Use in Power Optimization. Department of Electronic Systems, Norwegian University of Science and Technology. Jun 2018.
[3] Cadence Inc. Joules User Guide. Apr 2022.
[4] P. Dagli. Power and CGE (CLOCK GATE EFFICIENCY) analysis using POWERARTIST tool. Department of Electrical and Electronic Engineering, California State University, Sacramento. 2016.
[5] SIEMENS EDA. PowerPro User Guide. Mar 2022.
[6] F. Emnett, M. Biegel. Power Reduction Through RTL Clock Gating. SNUG San Jose 2000 .

Annexes

Gantt chart

The Gantt Chart concerning the evolution of my internship is presented in Figure 18, several parts can be identified: the documentation on the operation of the tools used by the company, the recovery and finalization of the current generator prototype, the setup for each tool, the power analysis and optimization, the writing of this report and the internship defense and the addition final of some features on the power tools.

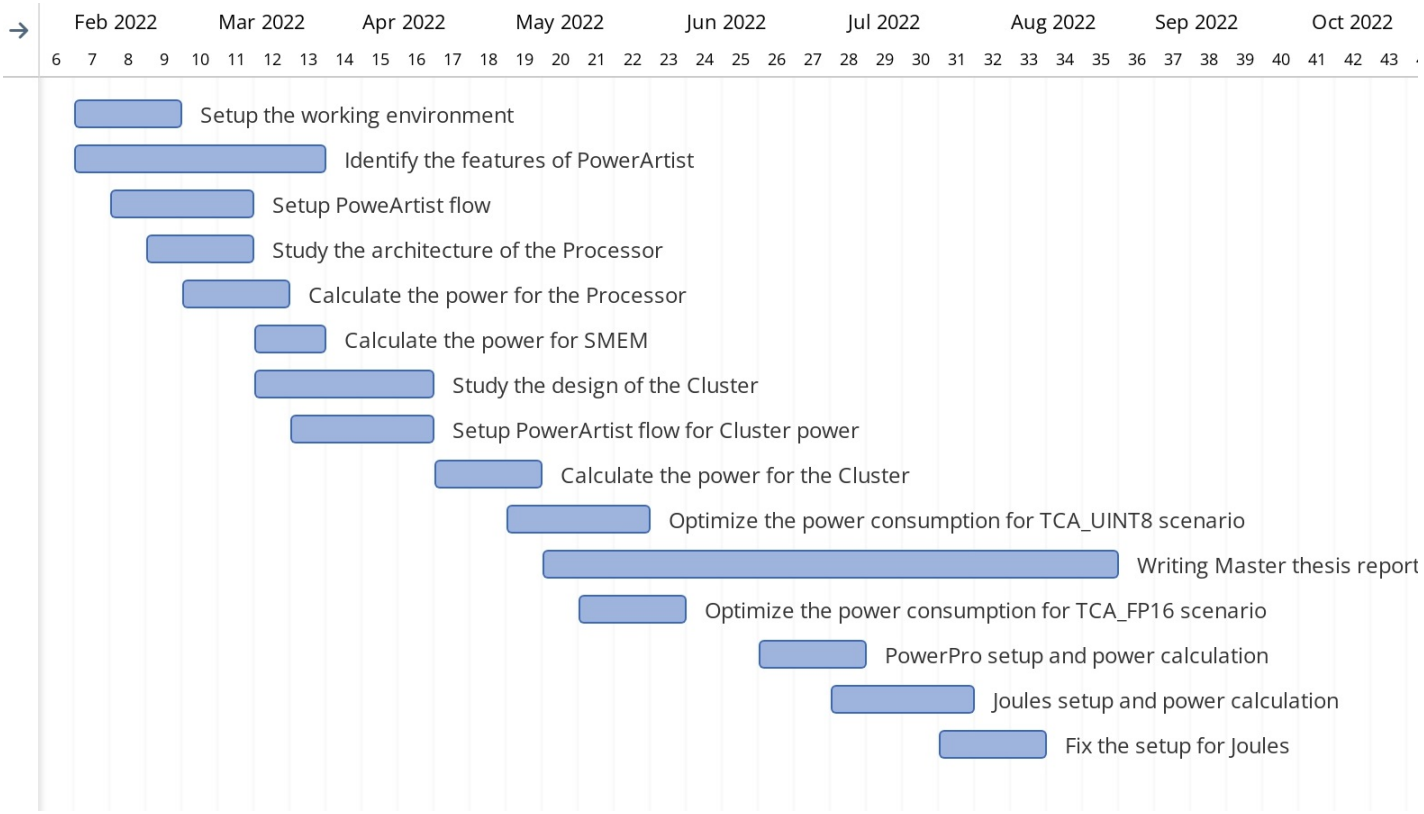


Figure 18: Timeline of the project