Master *Radiation and its Effects on MicroElectronics and Photonics Technologies* (RADMEP)

STUDY OF IR-DROP INDUCED JITTER IN HIGH PRECISION TIMING ASICS

Master Thesis Report

Presented by

David Peninon Herbaut

and defended at

University Jean Monnet

10th of September 2024

Academic Supervisors :	Prof. Jeffrey Prinzie and Prof. Frédéric Saigné
Host Supervisor :	Engineer Gianmario Bergamin, CERN
Jury Committee :	Prof. Arto Javanainen, University of Jyväskylä (JYU)
	Prof. Paul Leroux, Ku Leuven
	Prof. Frédéric Saigné, University of Montpellier (UM)
	Prof. Sylvain Girard, University of Jean-Monnet (UJM)

Study of IR drop-induced jitter in high precision timing ASICs

Abstract

The design, testing and manufacturing of application-specific-integrated circuits (**ASICs**) have become increasingly complex due to large-scale device integration and advancements in technology scaling. Very-large-scale integration (**VLSI**) has remarkably enhanced electronic circuit performance, impacting profoundly on daily life through various applications such as efficient microprocessors and larger memory chips. Despite these improvements, new challenges have arisen, particularly in designing an efficient power delivery network (**PDN**) that ensures a stable and evenly distributed power supply across the chip.

The rising integration of VLSI introduces critical challenges for PDN design, leading to power supply noise, voltage drops, and ground bounces, which themselves causes timing degradation, including jitter. High-energy physics (**HEP**) ASICs are likewise affected by IR drop-induced jitter, significantly limiting the performance of time-critical particle tracking chips. As a result, accurately determining the effects of IR drop-induced jitter through short-length simulations has become essential.

To address these challenges, an innovative approach for precisely predicting IR drop-induced jitter during the ASIC design phase has been developed. This approach is built on a simulation framework initially created by the experimental physics-electronic systems for experiments-microelectronics (**EP-ESE-ME**) group at the European organization for nuclear research (**CERN**). Subsequent development and implementation were conducted under my direct responsibility, focusing on refining, and enhancing its application. The completed framework offers a pre-silicon methodology to estimate IR drop impact on time-interval-error jitter (**TIE**) within a digital-on-top (**DoT**) approach. This technique involves correlating real switching activities with accurate dynamic power results for precise analogue timing simulations.

Additionally, this framework builds upon an existing methodology applicable on the commercial Computer Assisted Design (**CAD**) tools Cadence Innovus, Voltus and Tempus, which is further guided and integrated with Python and Tool Command Language (**TCL**) scripts to overcome existing limitations. It has been applied on two current ASICs developed at CERN and validated against experimental results where possible.

Keywords : Timing detectors, Pixelated detectors and associated VLSI circuits, Digital electronic circuits, IR drop, jitter, CERN, Cadence Voltus, Cadence Tempus, TIE jitter

Acknowledgement

This section will not be exhaustive due to the lengthy list of extraordinary people I would like to deeply thank.

Primarily, I extend my gratitude to my supervisor, M. Gianmario Bergamin, for his commitment and help throughout my master thesis, and furthermore, for his trust in me.

I am deeply grateful to Prof. Jeffrey Prinzie and Prof. Frédéric Saigné for their support, trust, and help during my thesis and my studies and for giving me so many opportunities to follow my dream.

My thanks go to the RADMEP committee for giving me the chance to participate in this master's degree, without which, I would not have had the opportunity to meet so many people, learn through prestigious universities, and travel across Europe.

I am indebted to Mme. Amelie Genvrin, Mme. Leena Mattila, Mme. Patricia Van Genechten and Mme. Inma Dosuna for their assistance with all the administrative processes I had to go through.

I appreciate the support of all the professors and teachers I have met over the last 6 years ; I thank you all for the motivation you gave me and all the useful hints you shared with me.

I would like to thank the EP-ESE-ME department at CERN, especially Dr. Federico Faccio, for the opportunity to complete my master thesis in his section.

Before concluding, I must share my thanks to my RADMEP colleagues and friends, with whom I spent these two years. This time will remain in my memories, and I am grateful to have experienced it with you all.

A big thanks to my friends in Montpellier and to all the coffee we shared back there ; you have made this university journey truly pleasant.

Finally, I extend my loving thanks and love to my family and Déborah for their support and help throughout this journey.

And to my best buddy, Yannick Noé. You rock man !

Table of Contents

List of Abbreviation

.lib	Liberty
.pgv	Power Grid View
.spef	Standard Parasitic Exchange Format
.vcd	Value Change Dump
AJAM-DOT	Advanced Jitter Analysis Methodology for Digital-On-Top
AoT	Analogue-on-Top
ASIC	Application Specific Integrated Circuit
CAD	Computer Aided Design
Cadence	Cadence Design Systems
CDR	Clock and Data Recovery
CERN	European Organization for Nuclear Research
CMOS	Complementary Metal-Oxide-Semiconductor
CTS	Clock Tree Synthesis
DoT	Digital-on-Top
DRC	Design Rules Checks
ECFA	European Committee for Future Accelerators
EIV	Effective Instance Voltage
EP-ESE-ME	Experimental Physics - Electronic Systems for Experiments - Microelectronics
GBA	Graph-Based Analysis
HDL	Hardware Description Language
HEP	High-Energy Physics
IO	Input-Output
IP	Intellectual Property
LGAD	Low Gain Avalanche Detector
LHC	Large Hadron Collider
lpGBT	Low Power GigaBit Transceiver
LVS	Layout Versus Schematic
MIP	Minimum Ionizing Particle
NIST	National Institute of Standards and Technology
PBA	Path-Based Analysis
PCB	Printed Circuit Board
PDN	Power Delivery Network
PLL	Phase-Locked Loop
PPA	Power Performance and Area
RTL	Register-Transfer Level
SEE	Single Event Effect
SPICE	Simulation Program with Integrated Circuit Emphasis
STA	Static Timing Analysis
TCL	Tool Command Language
TDC	Time-to-Digital Converter
TIE	Time-Interval-Error
TOA	Time Of Arrival

TOT	Time Over Threshold
UVM	Universal Verification Methodology
VLSI	Very-Large-Scale-Integration

List of Figures

List of Tables

Chapter 1 - Introduction

The High-Energy Physics (**HEP**) community plays a leading role in the quest for the origins of the Universe and the understanding of the fundamental constituents of matter. Thanks to a multidisciplinary research framework and global organization, significant advances have been made not only in particle physics, but also in areas such as medicine, space exploration and the development of the internet, to name three examples.

Among the research centers dedicated to particles physics using particles accelerators, the European Organization for Nuclear Research (**CERN**) stands out as the most important one. Founded in 1954 in Meyrin, Switzerland, CERN is home to several particle accelerators, including the world's most powerful : the Large Hadron Collider (**LHC**). The LHC is designed to generate collisions between packets of particles travelling at 99.9997828% of the speed of light, with an energy of 450 GeV, with a frequency of 40 MHz, and in a controlled environment, enabling scientists to evaluate and complete the Standard Model of physics, which falls short to explain certain phenomena, such as dark matter [1].

In greater depth, packets of particles, such as protons or heavy ions, are accelerated in successive stages in a complex network of accelerators, before being injected into the LHC as illustrated in Figure 1. The aim is to increase the probability of collisions in a short space of time. These collisions are then analyzed by the LHC's four major experiments : ATLAS, CMS, ALICE and LHCb. Each of these experiments has a specific mission.

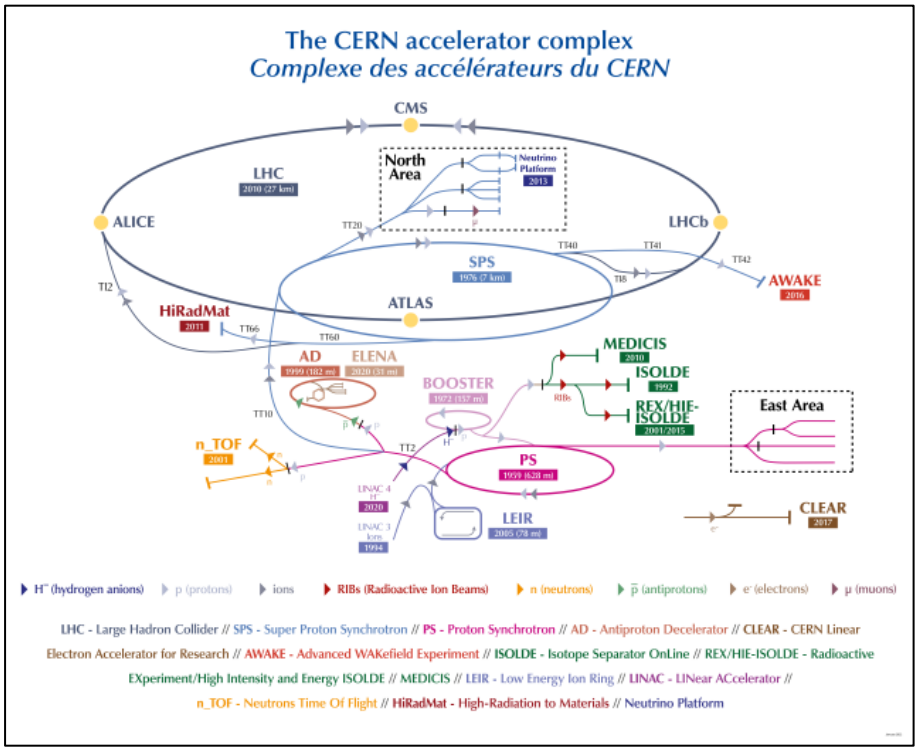

Figure 1 : The CERN accelerator complex [2].

For example, the ATLAS detector focuses on the detection of Higgs bosons, the exploration of new dimensions and the search for particles that might constitute dark matter [3]. Its operating principle relies on the fact that particle beams collide at the center of the ATLAS detector, which results in collisions debris. Those debris form new particles which spread in all directions and are tracked by an ingenious network of detectors arranged in layers. Additionally, powerful magnets are used to bend the trajectory of the particles to measure accurately their energy, speed, and position.

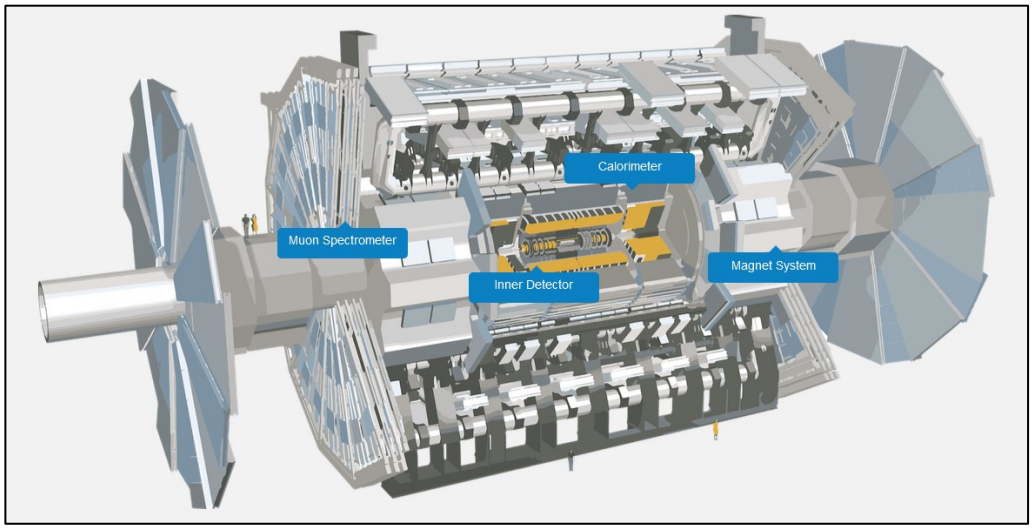

Figure 2 : ATLAS detector [4].

However, the collisions generate more than sixty million megabytes per second [5], hence, a clever trigger system is used to prompt which event to record and which to ignore. Additionally, the ATLAS detector operating conditions presents challenges for electronic systems such as the -40°C temperature at the detector's level [6] for example. This emphasizes the necessity of upgrading the electronics chips that compose ATLAS to keep on overseeing the harsh environment and improve the detection accuracy.

In a global context, state-of-the-art electronics are essential to support CERN's objectives. The experiments conducted require ever more precise electronic systems, capable of processing massive volumes of data while meeting tight performance requirements in harsh radiative and constraining environments. For this reason, several CERN teams, in collaboration with international research institutes, are working to develop high-performance detectors and electronic circuits.

1.1. Section EP-ESE-ME at CERN

The Experimental Physics - Electronic Systems for Experiments - Microelectronics (**EP-ESE-ME**) section is one of these teams. Its main activity is to develop Application Specific Integrated Circuits (**ASIC**) of extreme precision and reliability for future LHC upgrades. This section focuses primarily on the design of ASICs capable of processing the very weak signals generated by particle detectors, ensuring that designs meet the constraints imposed by the LHC, such as efficient power consumption, radiation tolerance and acceptable electronic noise for accurate analysis of results [7].

Considering the complexity of the digital logic to be implemented, the preferred design approach is the Digital-on-Top (**DoT**) methodology, which is particularly suited to the design of large-scale digital logic circuits. This approach operates at a high level of abstraction, making it possible to efficiently manage designs containing tens to hundreds of millions of transistors. DoT relies on the use of netlists and scripts to automate the circuit development process, in particular design implementation and optimization. In addition, this methodology uses Register-Transfer Level (**RTL**) scripts to accelerate development.

1.2. Future accelerators upgrades and current limitations

However, newly developed ASICs or those under development must meet strict specifications set by the European Committee for Future Accelerators (**ECFA**), which imposes demanding technical limits in terms of timing and associated uncertainties. According to the ECFA's roadmap for future accelerators, high-precision timing capabilities, specifically achieving a timing resolution of less than fifty ps, have become a standard for devices currently in development [8]. More specifically, the performance targets for detectors include the following goal :

"achieve a timing performance below the 10 ps level" [9]

Importantly, while this reflects the trend that designers are aiming for in future accelerators, it does not mean that all current ASIC designs are targeting a 10ps timing resolution already.

These increased requirements also highlight the importance of power integrity analyses and jitter control. ASIC Back-end development must consider power consumption, including dynamic power, which is becoming a limiting factor for future LHC upgrades. In addition, jitter control is crucial, with a limit of 25ps imposed to support the development of future accelerators.

This limitation is particularly relevant as original approaches to detector design increasingly rely on timing as a figure of merit. For example, in the context of the ATLAS and CMS detectors, an innovative technology known as "5D" has been created to use delay lines to calculate the velocity, position and energy of particles passing through the detector. By integrating such techniques for pixel-based detector, it is possible to improve the detection accuracy of CERN's experiments.

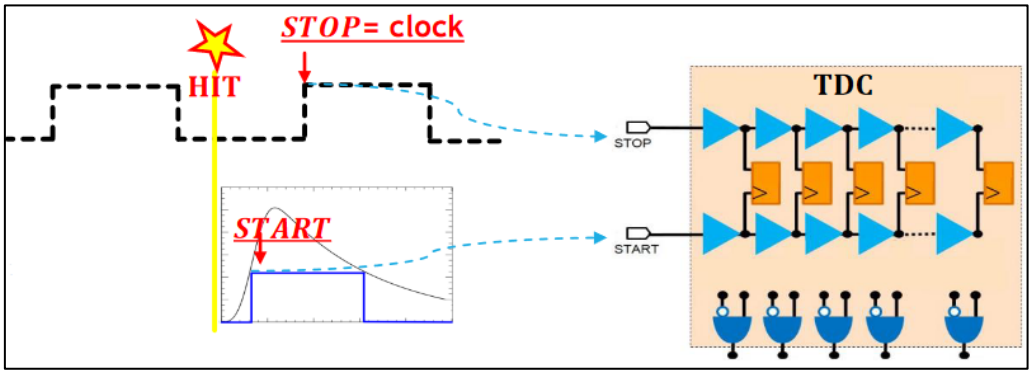

Figure 3 : 5D measurement principle [10].

Measurements in this technology include Time Of Arrival (**TOA**) to determine the time difference between the arrival of the detected signal and a reference. As well, Time Over Threshold (**TOT**) measurements are performed to quantify the particle energy. The TOT is then used to correct the "time walk" effect perceived in the TOA, enabling an accurate measurement to be obtained. In detail, “time walk” is a phenomenon that happens when different particles energy leads to different TOA. In the context of the ATLAS detector mentioned earlier, this implies that the triggers applied may miss particles as they would be considered in another bunch crossing due to the time walk, if not corrected. This principle, illustrated in Figure 3, highlights the crucial importance of jitter monitoring, as it limits the resolution of the detector [11].

In addition to these requirements, there is the transversal project DRD7-WG7.7, which aims to develop CERN's internal skills and tools to standardize and streamline the creation of ASICs. Moreover, the EP-ESE-ME section has identified a critical gap in the current methods for assessing the effects of voltage drops, also known as IR drop, induced by current spike on the power delivery network grid. These methods lack precision and key features for timing uncertainties simulations, making it necessary to develop a more accurate methodology. Until now, the limitations specified for jitter have been sufficiently broad for Static Timing Analyses (**STA**) in DoT to be sufficient to verify the timing criteria imposed. However, as performance requirements evolve, the current method is no longer sufficient to accurately verify the effects of IR drop on timing.

1.3. State-of-the-art : jitter simulating tools

The current DoT method, guided by recommendations from Cadence Design Systems (**Cadence**), proposes a methodology for simulating the effects of IR drop on timing uncertainties. This approach, detailed in section 2.2, is only applicable to late-stage circuits, and relies on accurate simulation of the circuit's logic activity to derive a simulation of IR drop.

The operating principle is to use a precise logic activity to derive an accurate circuit’s aware IR drop analysis. Then, IR drop results are used as inputs for simulating accurate timing simulations using a precise analogue software. In detail, the timing simulations consists of Simulation Program with Integrated Circuit Emphasis (**SPICE**) decks generated from the circuit’s information. In step by step, an “input-event”, which is the voltage waveform input sent on the first element in the circuit, is defined. Then, in separate stages, each element of the circuit is rigorously analyzed, including its parasitic elements and connected logic. Finally, for each clock cycle simulated with IR drop, a simulation is run, and timing reports are drawn.

Nevertheless, this method has critical shortcomings, notably the absence of explicit correlation between activity reports and power analyses, which can compromise the reliability of jitter analysis. In addition, this method lacks transparency for the user, and has significant limitations, such as the inability to analyze multiple clock domains simultaneously, or to correctly simulate D flip-flops [12], [13]. Additionally, triplicated paths are not addressed, leading to essential features being unavailable for the user.

Other approaches available in scientific literature [14], [15], also share similar limitations, notably the lack of explicit edges correlation between recorded activities, power analyses and timing analyses, which limits the accuracy of analog simulations. Due to the relative novelty of the DoT approach and its constant evolution, few articles address this subject in detail.

Thus, it has become crucial to fill these methodological gaps to meet the growing demands for precision in the design of new detectors and electronic circuits. To complement the existing methods and effectively integrate the proposed improvements, it was decided to use the Cadence software suite, already mastered by the EP-ESE-ME section for the creation of ASICs. The existing methodology provides a suitable basis for rapid implementation, while allowing for targeted improvements. On top of that, the work of M. Gianmario and M. Soulier [10], [16], has been used as the foundation for the later discussed improvements.

In this context, this master thesis aims to detail the current DoT ASIC design method, examine the effects of IR drop-induced jitter, and propose improvements for a reliable and accurate methodology. Additionally, two applications of the methodology on ASICs developed at CERN will be discussed to confront the proposed approach. Finally, a conclusion will summarize the main results and introduce future improvement.

Chapter 2 - ASIC Digital-On-Top

Modern ASICs design flows rely on Computer-Aided Design (**CAD**) tools and methodologies to derive the ASIC layout. Among the various methodologies, two major approaches are Analogue-on-Top (**AoT**) and DoT. The choice of approach is dependent on the nature of the ASIC.

For global digital logic circuits, the DoT approach, which operates at a higher level of description, is more adapted. This method is netlists and scripts based to automate the chip design process, especially the layout implementation. The DoT approach makes use of RTL scripts to enable rapid development, offering better scalability and speed for complex digital systems.

On the other hand, the AoT approach is used for precise analogue designs. It consists in creating schematics where components are placed individually or in minor groups, with the layout being drawn manually. This technique focuses on detailed design processes to maximize the Power, Performance, and Area (**PPA**) of the ASIC. Although the schematic approach of this method increases the difficulty yield by Very-Large-Scale-Integrated (**VLSI**) chips.

In this study, the DoT methodology will initially be presented since the ASICs being analyzed were designed by using this method. Following this, a comprehensive discussion on the power analysis and IR drop current framework will be provided. Finally, the impact of IR drop-induced timing degradation on electronic circuits will be discussed.

2.1. Digital on Top

In general, to simplify multi-million gate chip creation, ASICs designed using DoT are analyzed at three levels, each corresponding to a phase of the methodology. The top level of description, known as the system level, involves defining the ASIC specifications. Next is the logical level, which bound the RTL description of the microarchitecture, Intellectual Property (**IP**), testbenches for functional verification, and logic synthesis. Lastly, the physical level provides a detailed view of the ASIC through physical layout implementation and technological mapping.

According to the three analysis scales mentioned above, the DoT design methodology is structured in four major phases : defining the ASIC specifications, synthesizing the design, implementing the layout, and performing the sign-offs verifications. Each phase will be elaborated in the order presented. At all levels, the design is implemented to convert the initial ideas or specifications into logical or hardware designs. Also, verifications are performed at every step of the process to ensure functionality, timing, and design integrity.

For clarity, the entire section will reference Figure 4, which illustrate the simplified DoT flow from the ASIC specifications to the sign-offs, as provided by Cadence.

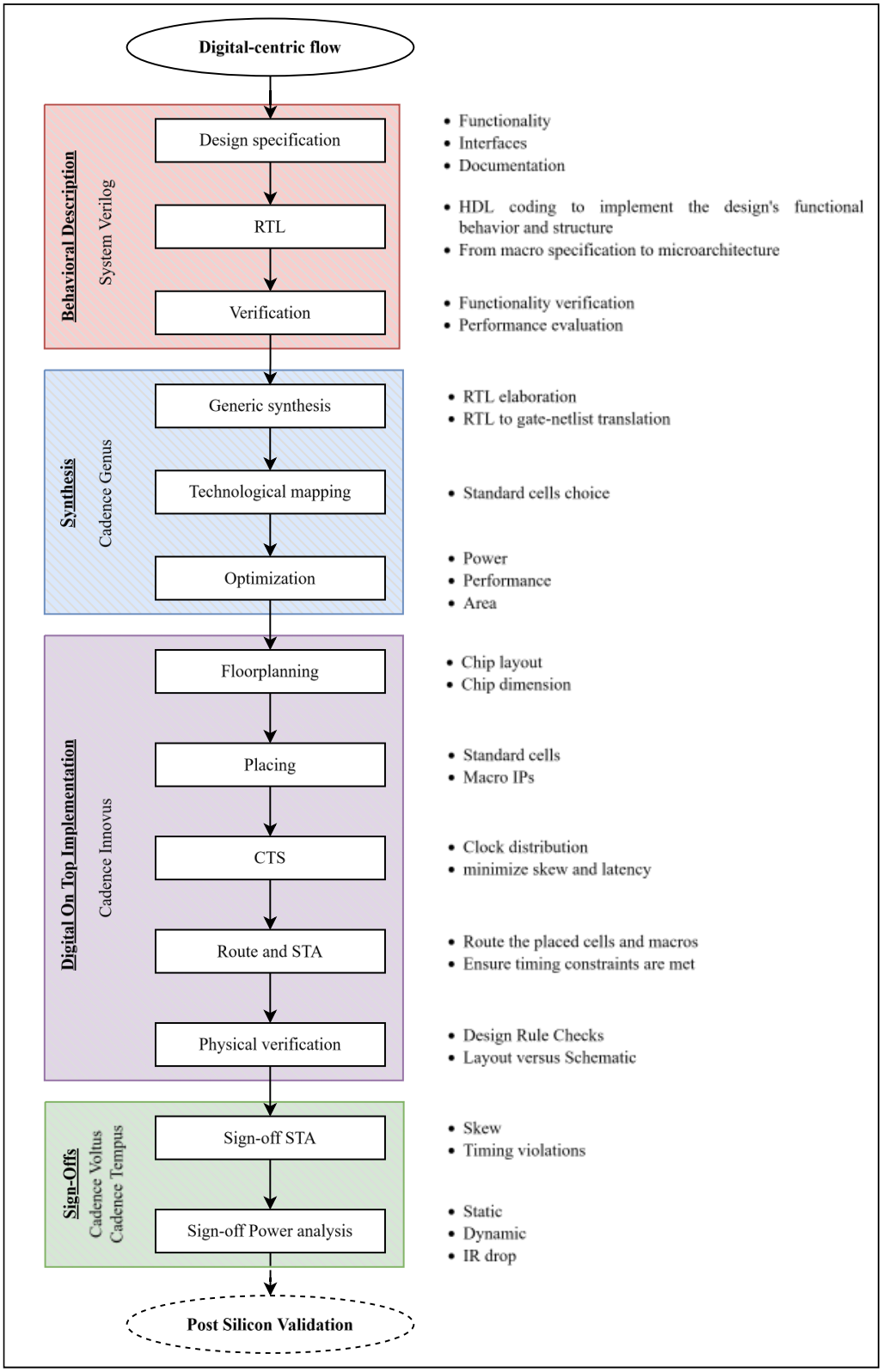

Figure 4 : Digital-On-Top simplified representation.

2.1.1. From specification to microarchitecture definition

The first step in the ASIC design flow is the specification, which is a set of explicit requirements that the ASIC must satisfies. This consists of designing on a high level of description the functionality of a chip, specifying the IPs and macro blocks to develop, the digital logic that must be created, the interfaces between the identified blocks and partition the ASIC into smaller scale blocks to be implemented [17].

At this stage, macro blocks functional principles are defined, and IP are carefully selected to anticipate future design challenges. These high-level decisions involve choosing between diverse types of electronic systems, allowing the designers to predict potential changes in the design and software before committing to it.

After, each macro block is detailed individually. This corresponds to the microarchitecture's creation, which implies detailing the composition of each block, the interfaces protocols, the input and output signals, and the internal operating principle of the microarchitecture. Also, mid-level design choices are made within the microarchitecture to determine the best trade-offs between performance and design constraints.

Once the macro blocks and microarchitectures and conceptualized, RTL is used to transform blocks and IPs into behavioral blocks using Hardware Description Language (**HDL**). This enables logic testing and simulation through testbenches. In details, during this stage, the blocks are designed for final implementation, IPs are interfaced and connected, block-level interfaces are created, and blocks are partitioned into sub-blocks. Finally, simulations are computed to verify the performance against the specifications.

Lastly, systems verification of the RTL description is confronted with the ASIC specification through accurate discrete logic simulations. In particular, testbenches are written to validate the performance of the design and make the required adjustments to the RTL, if necessary. However, simulations do not include any timing information at this point due to the purely behavioral nature of the design simulated [18]. As will be shown later, recreating the operating conditions of the ASIC is a fundamental requirement for obtaining accurate jitter results.

In the following section, the synthesis step of the DoT methodology will be presented. This phase involves translating the verified RTL design into a gate-level netlist, mapping the design, and performing various optimizations to balance the PPA specifications.

2.1.2. Synthesis and optimization

The second step of the DoT methodology is the synthesis phase. It constitutes a critical section in the ASIC design flow, where the high-level RTL descriptions are parsed and translated into a discrete gate-level netlist [17].

This process starts with elaborating the high-level RTL design into an intermediate more explicit form. In other word, the HDL description is expended to represent all instances into unique objects by reading the standard logics element that composes the RTL modules [19].

Afterward, the design is translated into a gate-level netlist. In other words, converting HDL description into logic-gate and sequential elements to implement the specified ASIC functionalities.

Following this, a technological mapping is undertaken to meet the timing and power requirements of the ASIC. On a side note, delay and power consumption information are contained in the standard cells' libraries [18].

Finally, multiple iterations of verification are performed, including verifying the quality of RTL, STA and the standard cells chosen to maximize PPA metrics.

In the next section will be explained the implementation step of the design, known as place and route. This development phase calls for significant stages to be completed such as floorplanning, clock tree synthesis, routing, and various design verifications.

2.1.3. Physical implementation

The third step of the DoT methodology is the physical implementation in which the physical layout and chip integration are accomplished. As shown in Figure 4, the implementation is divided into five sub-stages : floorplanning, placing, clock tree synthesizing, routing, and physical verification. At each stage, optimization iterations are conducted to maximize PPA and satisfy the timing requirements of the ASIC.

Primarily, the floorplanning is performed. It denotes laying out the physical partitions of the design to determine the size, connectivity, and location of each partition. Once the floorplanning is completed, the placement stage begins, where the optimal positions for all discrete integrated components of the ASIC are determined. Extra attention is required during this process due to the increasing number of logic elements being added to the design that may not have been present in the initial RTL description [17]. Moreover, for CERN or radiation tolerant ASIC, a specific distance defined by analyzing environment may be necessary between the registers and voters of the triplicated path to improve their tolerance to Single Event Effects (**SEE**).

Regarding the further steps, Clock Tree Synthesis (**CTS**) is performed to insert buffers along all clock paths, aiming to reduce clock skew and latency. Subsequently, the design undergoes routing, which manifest connecting the standard cells, macro blocks and Input-Output (**IO**) to the metal layers in the process technology to ensure alignment with the schematic [17].

Lastly, a round of verification is undertaken. Indeed, the parasitic resistance and capacitance of the interconnect of the physical design are extracted and used to calculate the added delay in the design. In addition, signal integrity verifications are conducted since parasitic elements may produce noise, impact the delay on the design or provoke glitches through cross talking or coupling capacitance. Furthermore, Design Rules Checks (**DRC**) and Layout Versus Schematic (**LVS**) verification are completed, ensuring the respect of the specifications and foundry's rules [17].

In the next section, the last step of the DoT approach, which is the sign-offs verification, will be presented. This step betokens accurate timing analysis and power analysis.

2.1.4. Sign-offs verifications.

As stated above, the sign-off verification step is required as the last step in DoT design flow. It encompasses a STA and three types of power analyses : static power analysis, dynamic power analysis, and rail analysis.

Firstly, STA ensures the clock and data paths are optimized, producing no timing violations. As noted in section 2.1.2 various static timing analyses are performed throughout the synthesis and implementation steps. To provide more details, STA are measuring cell's delays and slew to verify if the timing constraints applied on a path are satisfied for setup, hold time and skew. The engine used for early and sign-off STA is the same, however, the timing computation method differs as it can be : Graph-Based Analysis (**GBA**) or Path-Based Analysis (**PBA**) [20].

The GBA computation technique is the default one employed by the Cadence engines. It provides fast and pessimistic analysis by computing only the worst input slew and delay regardless of which path is effectively used by the circuit. Additionally, the output slew is considered to be 25 percent higher than the worst input slew [20].

The PBA computation technique is employed at a late stage of the design process or to precisely verify specific path. Conversely to GBA, it considers the slews and actual timing information based on the path simulated. Therefore, it removes the pessimism caused by GBA to accurately verify if timing constraints are satisfied. However, this comes with a simulation time trade-off as it takes longer to run [20]. In final, this technique is preferred for the sign-off verifications as it accurately reports the required timing information to validate the ASIC.

Subsequently, a power analysis is conducted to precisely determine the power consumption of the chip. The static power analysis permits verification of the average power consumed in a specific corner over a given period. Next, the power analyses are undertaken to assess accurately the static and dynamic power consumption of the design and extract the current and voltage profile of the ASIC for a given corner. Lastly, a rail analysis is executed to verify the quality of the Power Delivery Network (**PDN**) against IR drop and secure adequate power distribution.

The sign-off power analysis being crucial to meticulously simulate jitter, the following section will provide an extensive discussion about the required stages and options to achieve the highest level of precision.

2.2. Power analyses and IR drop impacts on timing.

As discussed previously, the sign-off power analysis verification is a crucial part of the DoT methodology, and it also plays a significant role as the foundation of the later discussed jitter analysis. Typically, the power integrity verification is composed of a static power analysis, a dynamic power analysis, and a rail analysis.

Each analysis must read the extracted parasitic resistance and capacitance from the Standard Parasitic Exchange Format (**.spef**) generated during the previous step. Also, the voltage sources or power domain under studies are defined and optionality, a Value-Change-Dump file (**.vcd**) is used to determine the switching profile of the ASIC.

Elaborating on the .vcd file, it is the output of a gate-level netlist simulation. Lastly, the Liberty files (**.lib**) are defined to represent the timing and power properties of the cells in the design by including cell's delay, internal power and transition time [21].

Next, in the case of rail analysis, the Power Grid View files (**.pgv**) are declared to read the ASIC current taps and coupling capacitance distribution. As well, in the case of macro cell view, for analogue blocks or IP importation, the library embeds the reduced distributed RC network [22].

Prior to detailing any of the power analyses, it must be noted that only one power domain at a time can be simulated. Therefore, all related timing analysis performed on top of any power analysis is valid only in the domain evaluated and does not account for interaction between power domains.

2.2.5. Static power analysis and static IR drop

The first step of the sign-off power analyses is the calculation of the static power consumption. This aims to simulate the various power consumption metrics, and the average voltage drop over a specified time window. The calculations outputs include : the leakage power, which is the power consumed by a cell when it is not switching but has a voltage applied to it [23] ; the average switching power, which is the power consumed by the ASIC when transistors are actively switching states ; and the average internal power consumed when by the transistors when the P and N gate transistors are ON simultaneously [23], [24]. Additionally, regarding static voltage drop, Cadence states :

> "Static IR drop analysis is a first-order approximation that uses the total power dissipation to calculate a constant current draw." [25]

However, the complete Static power analysis embeds all types of power, including internal power which represents around ten percent of the power consumed [23] and the switching power which is one of the major sources of timing uncertainty.

Regarding the static IR drop, it refers to the voltage drop encountered by the power delivery network grid when a constant current is applied on it. This helps identify the section of the ASIC that is affected by significant average voltage drops, which may lead to a static timing degradation.

To shed more light on the calculation of the different types of power, the leakage power is extracted from the state-dependent leakage data from the .lib libraries parsing. Indeed, the .lib libraries embeds the "cell_leakage_power" information which is linked to a set of operating conditions. The computation is done by summing the leakage values based on the instance's input probabilities [25] and the circuit parasitic, which are defined by the voltage sources applied, the duty cycle and the design topology.

Next, the internal power is calculated as the average of the state-dependent arc-based equation. It corresponds to the power consumed by a cell when a short-circuit is happening during a switch. The calculation takes into account the load seen by the output of the cell to look up the associated energy value in the power tables and the slew

[26]. To provide an example, the next formulas consider a buffer with the input "A" and output "Y" [25] :

$$E_{Y,edge}[J] = \frac{D(A) \times \frac{1}{2} \times [\, E(Y_{edge} \times A_{rise}) + E(Y_{edge} \times A_{fall})\,]}{D(A)} \quad (1)$$

$$Internal\ Power[W] = \frac{D(Y)}{2} \times \sum E_{Y,edge} \quad (2)$$

With D(A) and D(Y) the transition density of the input, output, and $E(Y_{edge} \times A_{edge})$ the energy extracted from the lookup tables for a given combination of input and output edges. Furthermore, in the simulated CERN ASICs within this master thesis, internal power accounted for up to ninety percent of the total power consumption.

Lastly, the switching power which is caused by the charging and discharging of load capacitances, the switching activity, and the frequency. In digital design, it represents a consequent challenge to reduce switching power as the embedded logic is increasing with technology scaling. Additionally, the switching power is calculated by the Cadence engine as follows [25] :

$$P = 0.5 \times CL \times V^2 \times F \times A \quad (3)$$

Where CL represents the capacitive load, V is the voltage, F is the frequency and A is the average switching activity. Then, the switching power is averaged over the simulation window in the context of the static power analysis [25]. Furthermore, in the simulated CERN ASICs within this master thesis, switching power accounted for up to thirty percent of the total power consumption.

In the case of .vcd file-based approach, the transition per instance for a given time window is considered to weight the input probabilities.

In conclusion, this analysis is important in ASIC designs as technology nodes continue to shrink and the number of transistors increases. Thus, larger leakage, internal and switching power are expected, potentially leading to diverse issues such as increased cooling requirements or performance degradation. More importantly, as internal power and switching power represents most of the power consumption in ASICs, it is crucial to assess accurately their effects on timing degradation, as it will be discussed later.

2.2.6. Dynamic power and rail analyses for dynamic IR drop extraction

The second step of the sign-off power analyses is the calculation of the dynamic power consumption. This calculation is based on a switching activity profile of the ASIC to determine the switching power consumed. The calculation performed by the Cadence engine follows the formulas defined in (3) to derive accurately the dynamic power consumption.

About the activity profile of the chip under the test, it may be defined at two levels of accuracy : vectorless and vector-based methodologies.

Firstly, the lower accuracy model is the vectorless approach. This technique uses timing windows information to generate dynamic power and current waveform to insert them in the design [25]. In detail, the vectorless methodology may be further categorized into probability-based or state-propagation-based. In the probability approach, switching probabilities are assigned to all the cells in the data path based on user-specified constraints [27]. On the other hand, the state-propagation-based technique implies annotating the rising and falling events on the data path by propagating the events from the output sequential logic elements. This approach prioritizes cells associated with faster clocks, higher fanouts and user constraints [28].

Secondly, the vector-based methodology shows greater accuracy due to the technique being based on a .vcd file. Like for the static power analysis, the .vcd file is parsed to determine which instance is switching and the precise time at which the activity is happening. The difference lies in the fact that in static analysis, the current is averaged, which is not the case for dynamic analyses. This allows for realistic vector generation and accurate dynamic power consumption calculations [29].

Although, it must be noted that each approach enables determining the current and waveform profiles of the dynamic power along with the power metrics.

Once the dynamic currents are computed, a dynamic rail analysis can be conducted to estimate dynamic IR drop. The rail analysis consists of linking the switching activity with the parasitic elements of the PDN by applying the previously generated currents waveform to the current taps of the PDN and observing the voltage drops. Voltage drops are then reported and if specified, the Effective-Instance-Voltages (**EIV**), which are the voltage available considering voltage drops and ground bounces, may be obtained [12]. Importantly, the precision and accuracy of the dynamic rail analysis is crucial as it represents the foundation of the jitter analysis calculations that will be introduced in the following section.

2.2.7. IR drop-induced timing degradation and simulation.

The power analyses presented in the previous section indicate that power consumption and switching activity may significantly impact timing. By all means, multiple studies have demonstrated that IR drop affects delay and slew because the voltage swing range caused by IR drop is smaller than the nominal range [30], [31].

To elaborate further, Figure 5 depicts the impact of IR drop on both setup and hold times. It clearly shows that the delays encountered vary when IR drop is considered. This delay is caused by the reduced peak voltage, which implies that circuit instances have less driving strength compared to the ideal scenario.

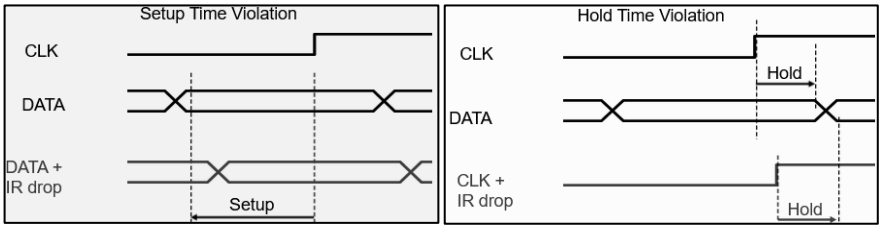

Figure 5 : Cadence figure, Setup and Hold time violations representation [25].

To complete the discussion, Figure 6 below illustrates the PDN grid composed of resistance and capacitance, and the effect of IR drop on the slew rate. IR drop reduces the nominal voltage supply and degrades the slew slope, leading to a longer transition time.

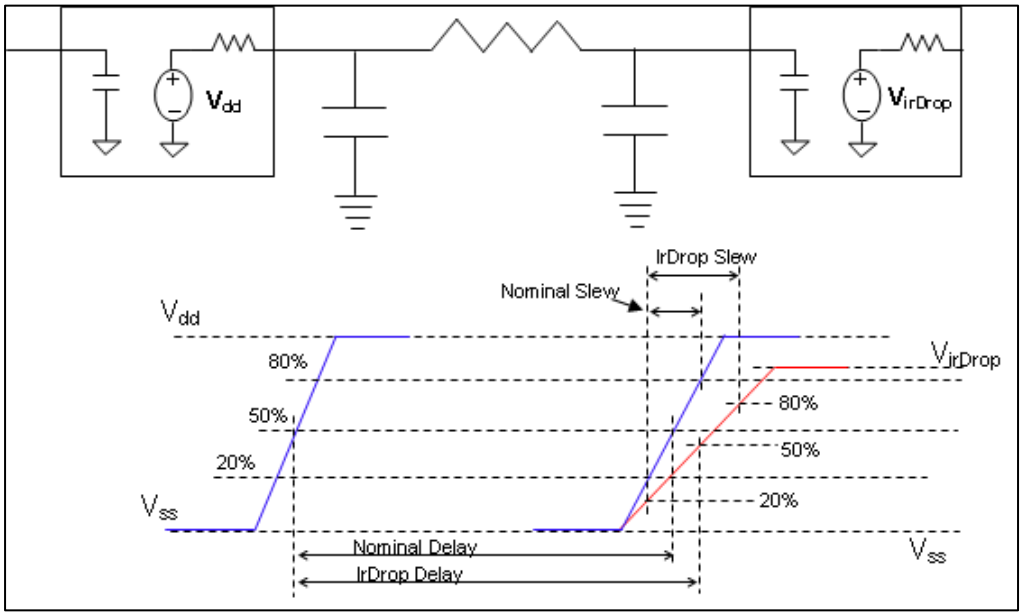

Figure 6 : Cadence figure, dynamic IR drop effects on slew [30].

Overall, timing is degraded. However, since dynamic IR drop is activity-dependent, the delay encountered, and the slew degradation will not be constant from one clock cycle to another. This variability implies an IR drop-induced jitter effect [16].

To shed more light on IR drop-induced jitter, the delay degradation discussed along with Figure 5 in the "DATA + IR drop" waveform is the Time-Interval-Error (**TIE**) jitter. It is the absolute deviation of a clock event from its nominal position [32]. Another visual representation of TIE jitter is depicted in Figure 7.

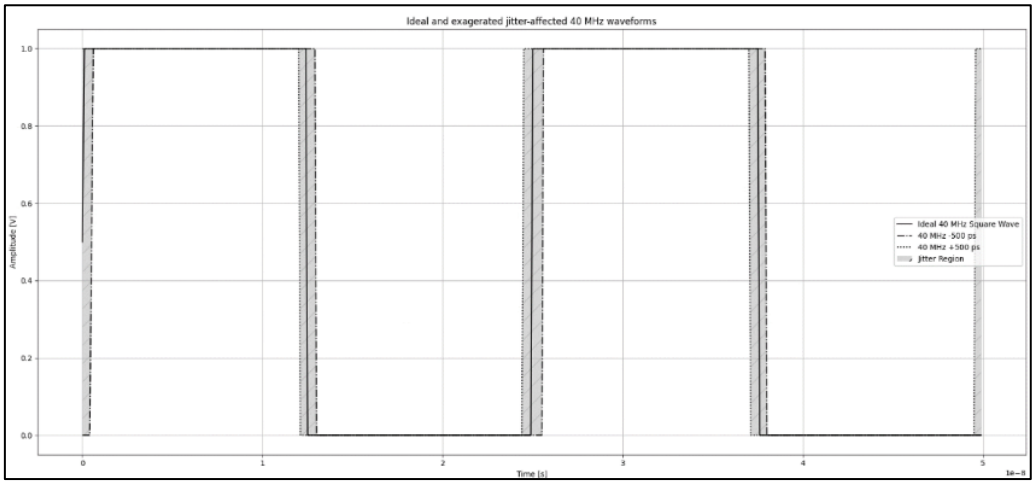

Figure 7 : IR drop-induced jitter representation.

To provide more context, IR drop-induced jitter belongs to the “deterministic” category of jitter because it is caused by predictable and consistent variations in the power supply voltage. This predictability arises from the fact that IR drop is induced by current flowing through the PDN composed of resistive and capacitive elements. Since these resistive elements are intrinsic to the design and the current flow is a regular occurrence regardless of the specific application, the resulting IR drop is consistent. Additionally, IR drop can be related to specific and repeatable causes such as circuit switching or power grid resistance, which leads to bounded and reproducible jitter [31]. However, IR drop-induced jitter is not the only source of deterministic timing uncertainty. Other sources include cross talk or grounding problems [33].

Therefore, as presented in the formula below [34], ASIC timing resolution is influenced by multiple sources :

$$\sigma_{ASIC}^2 = \sigma_{analogFE}^2 + \sigma_{conversion}^2 + \sigma_{clock}^2 \quad (4)$$

Where in this context, σ_{clock}^2 encompasses the effects of clock generation and distribution through the PDN, including the impact of IR drop.

Moving on to simulation, IR drop-induced jitter can be computed using various software, including the Cadence suite. However, as will be discussed in the next chapter, the DoT methodology associated with the Cadence suite lacks critical processes necessary for accurately calculating jitter induced by IR drop. One of the key issues identified is that the EIV file is not correlated to the switching activity.

Specifically, the EIV switching windows are considered regardless of the transition or currently analyzed cycle, leading to incorrect edge annotation and inaccurate jitter results. Furthermore, the current DoT methodology struggles managing clock cross domain and triplicated path, which are crucial for CERN ASICs. Lastly, the jitter resolution provided is in picoseconds preventing accurate analysis for sub picosecond jitter measurements, which are necessary in the case of short path or single cell analysis.

In detail, the simulation starts from the dynamic rail analysis EIV report. From there, the user can prompt to a dedicated command of the Cadence software’s which path should be evaluated, which type of jitter to simulate, what SPICE model files to use, which transition edge to measure and which EIV report should be used as the input voltage for the simulation. Then, SPICE decks are automatically generated by the software and computed, resulting in picosecond resolution jitter reports. The process is transparent from the user point of view, but as discussed above, it lacks critical features which are difficult to verify as the whole process happens without the user intervention.

Additionally, the SPICE decks generated can be observed and it can be extracted their operating principle from them : an “input-even” is defined to represent the voltage waveform input of the path, then in multiple stages are analyzed each cell to extract their parasitic components and verify the receivers and drivers attached. Finally, multiple measurements are explicated with voltage thresholds to measure only the delay of each stage in the path under test.

As a partial conclusion, power analyses including rail analysis are based on .spef file, which provides information on the parasitic resistance and capacitance of the ASIC. Additionally, .lib files are necessary to link timing and power data together. Optionally, a .vcd file can be used to recreate a realistic switching activity. Finally, .pgv libraries are essential to provide information on the current tap, coupling capacitance, circuit geometry and layout of all cells and macro blocks. All these files, combined with the DoT methodology, enable precise assessment of the power integrity in various scenarios. Moreover, power supply variations lead to timing degradation, and consequently, jitter.

Nevertheless, the current DoT methodology lacks accurate techniques to verify IR drop-induced jitter. Therefore, the next chapter describes a new methodology appended to the final stage of the DoT process, addressing the gap between power analysis and precise jitter simulations.

Chapter 3 - Advanced Jitter Analysis Methodology for Digital-On-Top (AJAM-DOT)

In the previous chapter, it was concluded that the current DoT methodology falls short in accurately verifying IR drop-induced jitter. To address this limitation, the Advanced Jitter Analysis Methodology for Digital-On-Top (**AJAM-DOT**) was developed and appended to the final stage of the DoT process. The AJAM-DOT technique is implemented at an advanced stage of ASIC design to ensure an accurate power integrity analysis is available. To provide more context, this new methodology takes inspiration from the work of M. Bergamin and M. Soulier [10], [16].

In this chapter, AJAM-DOT will be introduced and detailed. The integration process with the existing DoT methodology will be explained along with the specific technique used to bridge the gap between the power and jitter simulation. A complete explanation of the current problem with jitter simulation in the DoT process will be described and the solution found will be detailed along with them. In a logical approach, the methodology will start from the switching activity simulation, and the importance of the clock tree synthesis will be discussed. Following this, the power analysis required upgrades will be presented. Finally, the process of accurately simulating timing will be explained.

To provide a clear view of the AJAM-DOT methodology, a visual representation is illustrated in Figure 8. The principle of the methodology entirely relies on correlating the switching activity with the EIV windows generated in the power analysis. Which themselves serves as input for the timing analysis completed by setting up a SPICE deck and simulating it through precise analogue simulations. Finally, a post-data treatment is performed to extract jitter measurements.

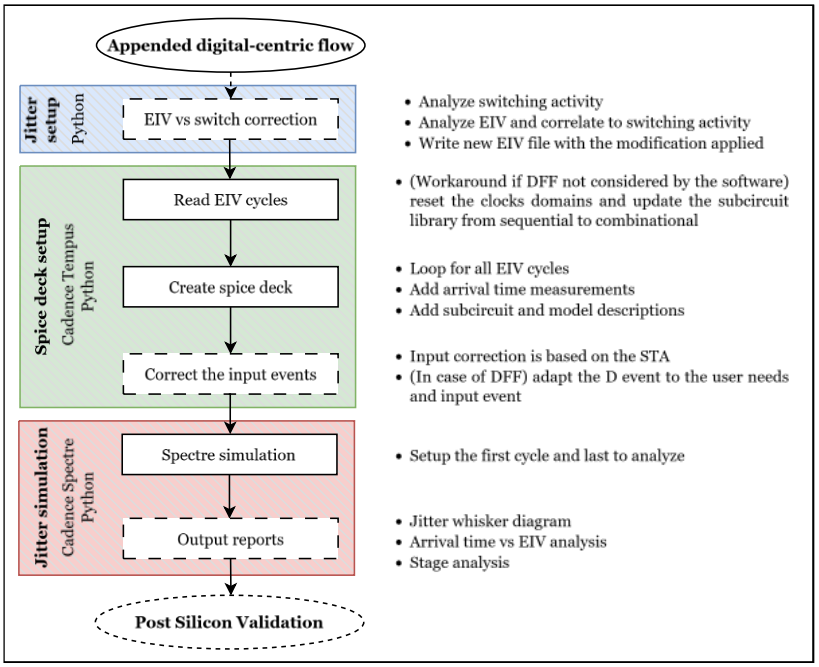

Figure 8 : AJAM-DOT - Appended jitter methodology.

3.1. AJAM-DOT prerequisite

The AJAM-DOT presentation will first begin with a brief introduction of the software used to develop it. Then it will be introduced the requirements an ASIC should fulfill for AJAM-DOT to be applied.

Prior to commencing the analysis, the DoT methodology followed at CERN relies on the Cadence suite, which is presented in Table 1. The appended AJAM-DOT methodology aims to complete the software's limitations in precisely simulating IR drop-induced jitter. Also, Python and Tool Command Language (**TCL**) scripts have been developed to guide the AJAM-DOT methodology and integrate it into the DoT flow. On a side note, AJAM-DOT is supported either on legacy or common-UI, also known as the "stylus" version of the Cadence suite.

Software	Tested version	Purpose of software usage
Cadence Xcelium	20.09.022	Digital logic simulation
Cadence Innovus	22.13.000	Clock tree extraction
Cadence Voltus	22.13.000	IR drop analysis
Cadence Tempus	22.13.000	Timing simulation setup
Cadence Spectre	19.10.541	Analogue simulation

Table 1 : CERN DoT tools.

To provide more details, it was mentioned before that the ASIC should be in an advanced stage, which means that clock trees and switching activity should be available, as well the ASIC must be at the sign-offs stage. It will be deeply explained below the importance of the clock tree analysis. As a general note, clock trees allow for extracting the design topology to ensure a coherent timing analysis which is design aware. Next, the required options to use in the rail analysis will be listed and explained along with a detailed correlation with the switching activity. Finally, the workaround proposed to correct three Tempus flaws will be developed.

3.2. Step 1 : Clock Tree Analysis

In the first place, clock trees analysis is essential for determining the ASIC topology and completeness. Indeed, it provides valuable information about the path under test, such as the presence of inverters or sequential elements. To shed more light on clock trees, Figure 9 illustrates a clock tree composition, which for example shows that multiple clock domains can coexist in an ASIC. Additionally, clock trees may vary in length, as represented by the dotted line on the left side of the figure. Also, clock sinks are symbolized by black boxes. Lastly, the path under analysis for simulating jitter is exemplify in red.

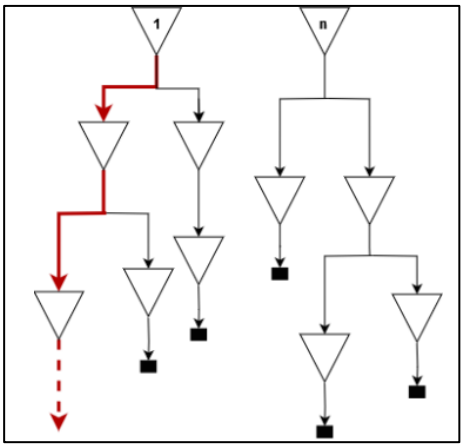

Figure 9 : Simplified clock tree structure.

In the scope of AJAM-DOT, clock trees are employed to optimize reporting by narrowing the focus to the clock path, including all associated fan-ins and fan-outs. This approach effectively reduces the length and parsing time of the reports. However, power analyses are conducted on the entire ASIC. Hence, the use of clock trees does not impact computations.

To perform this step, a clock tree is generated from Cadence Innovus. Next, its structure is extracted utilizing the software's built-in command [35]. Finally, the output is dumped into a file for later parsing and formatting to facilitate use. Once the clock tree file is available, the treatment is performed by a dedicated Python script, which was developed during this internship, to extract the topology and information related to the clock tree. In addition, this step is crucial for the AJAM-DOT methodology to correlate the topology of the circuit to the power reports explained later.

In the next section will be detailed the most important step of AJAM-DOT : how the correlation is done between the logic activity simulated and the power characterization.

3.3. Step 2 : Activity and power analysis correlation

The second step of the AJAM-DOT technique is the correlation between the switching activity and the dynamic power analysis performed in the DoT original flow. This represents the core of the AJAM-DOT as the accuracy of the power results and links to the switching activity are the foundation of the jitter analysis.

To begin, the accurate activity vector is generated using Cadence Xcelium. It must be representative of the ASIC operating conditions as it serves as the foundation for power characterization. Using methodology and languages such as Universal Verification Methodology (**UVM**) and system-Verilog, it is possible to create realistic and, at the same time, randomly constrained working scenarios. Once setup, which in the context of this master thesis implies using already available simulations created by the EP-ESE-ME designer's, these simulations are used to select the desired analysis window over the full simulation time. The activity information is then stored in the .vcd file presented in paragraph 2.2 which records the toggling instances, time by time.

Regarding the number of lines dumped in the .vcd file, it was decided to record two different .vcd files for the same simulation : one containing the full design for power analyses, and another one containing only the instances under analysis which are processed by Python. This approach reduced the required computer resources and time.

Additionally, although the logic activity is automatically parsed and read by Cadence Voltus during the power characterization stage, reports do not account for cycling coherency. Therefore, in a standalone Python process, relevant information is extracted from the .vcd file, such as each instance's transition time. In more detail, the time specified in the power analysis is used as a reference "t=0". However, in the case where no time are specified, the first time encountered in the .vcd file is set as the reference [26].

Furthermore, Figure 10 below shows the standalone Python script flowchart developed to parse and read the switching activity from the .vcd file. The process starts by defining the required variables such as the data path, edges and starting time of the analysis. Then, by parsing the clock tree created in step 1, relevant data can be extracted, including the instance per clock domain, the input signal of each instance as well as the topology of the clock tree. Finally, this clock tree information is applied to the .vcd to extract the frequency, first transition edge and the three first transition time per instance.

The goal of this Python script is to later correlate the precise timing activity with the EIV timing window generated during the previously computed dynamic rail analysis.

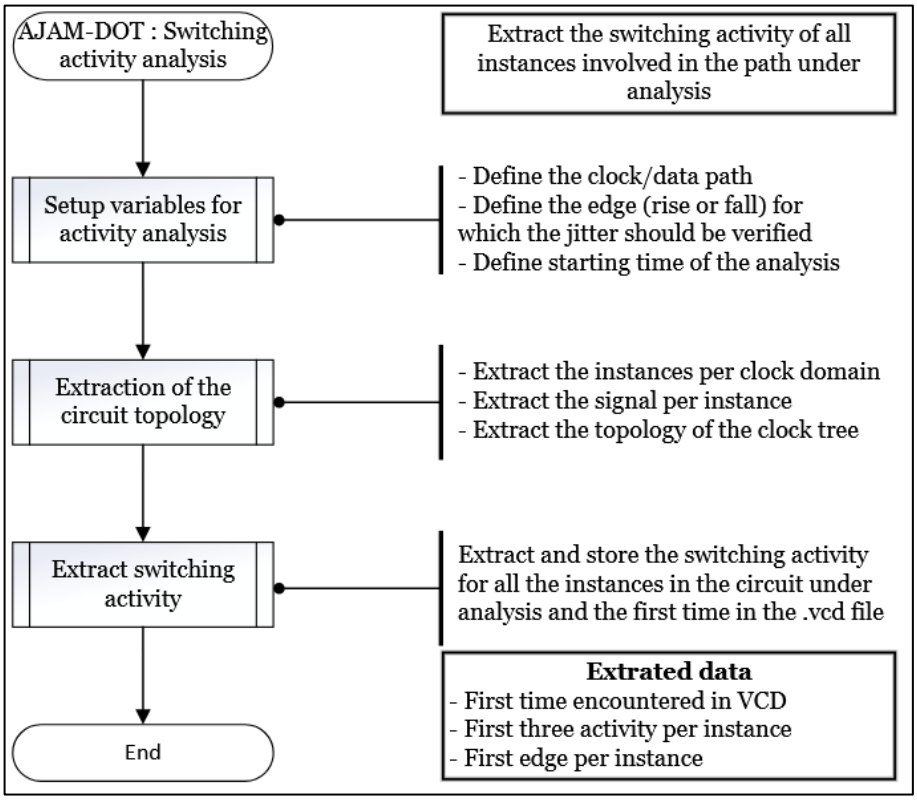

Figure 10 : Python script one, switching activity.vcd analysis.

Once the switching analysis is completed, a Python verification is performed to ensure consistency in between the logic activity and the power reports. A simple flowchart depicted in Figure 11 provides more understanding of the script developed. Fundamentally, instances names are extracted from both the clock tree and the power analysis report to be confronted with each other, pointing out to the potential analysis weaknesses. Those weaknesses are unanalyzed instances : they indicate a consistency problem between the clock tree, switching activity and power analysis. One likely reason is that these instances are not switching during the specified time window.

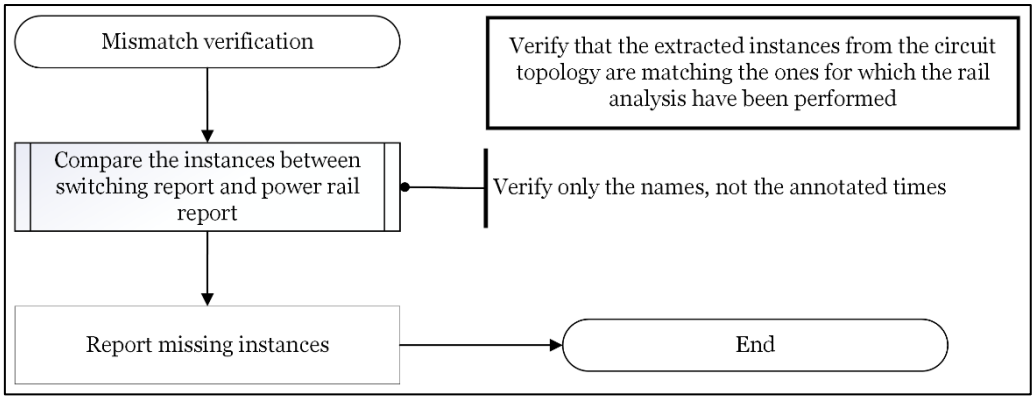

Figure 11 : Python script two, mismatch verification.

Assuming the switching analysis completed without mismatch, the power analysis is then performed. At this stage, it is worth noting that only dynamic power and rail analysis are necessary. Indeed, since the jitter source to be verified is presumed to be induced by the power supply variations, the static analysis may be omitted. Also, a couple of options must be used in the built-in Voltus command to setup the rail analysis reports : set the EIV calculation method to be worst, ensuring a worst-case scenario to the power simulation ; and set the report to be written only for the clock tree instances [36] to limit reports length.

In greater depth, the EIV generation analyzes the switching time of each instance and reports the average voltage seen during the related current spike. Results accuracies are related to the time resolution of the rail analysis which leads to larger or smaller time windows. EIV and logic transition are illustrated in Figure 12.

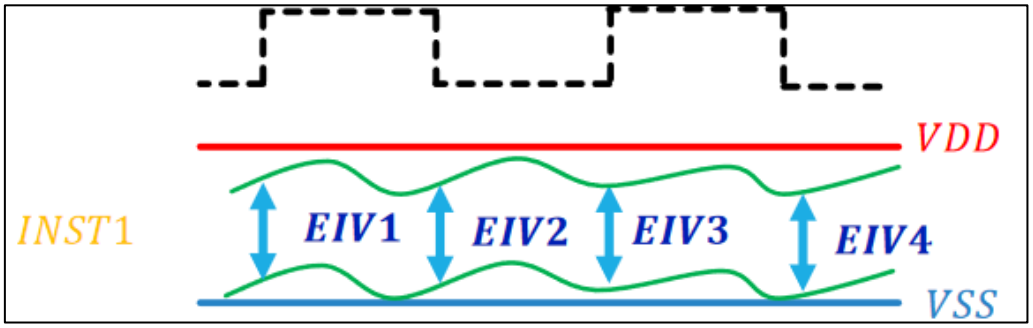

Figure 12 : EIV representation [10].

However, as observed in the figure above, the EIV windows are generated for all transitions, regardless of the rising or falling edge. At this stage, it does not yet represent a problem. Nevertheless, the next step of AJAM-DOT is common with the state-of-the-art flow, which implies creating SPICE decks using Tempus built-in commands [37]. During the SPICE deck generation, EIVs are annotated on the circuit as they appear in the report file, without considering if they belong to the correct cycle.

In other words, as illustrated in the following Figure 13, EIVs are annotated against the switching edges for AJAM-DOT and Cadence, in a clock cross domain example. AJAM-DOT specifically targets edges that are rising or falling, if specified by the user, and correlates them to the generation of the edges in the second clock domain. Conversely, Cadence annotation considers all the edges without correlating them, leading to incorrect SPICE deck voltages annotation, since the temporality of the clock domains and edges coherency are not respected.

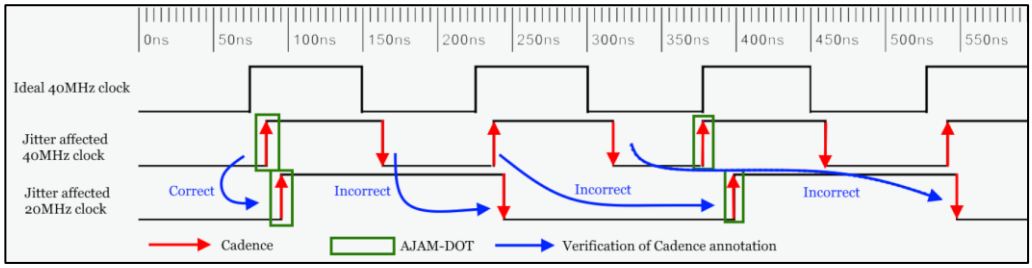

Figure 13 : Edges annotation for clock cross domain, Cadence versus AJAM-DOT.

To provide more insight into AJAM-DOT, a detailed flowchart depicting how the correlation between the switching analysis and EIV reporting is portrayed in Figure 14. The information conveyed by the diagram clearly defines three stages : setup, correlation, and output. Also, the entire process is performed in a standalone Python script developed during the master thesis.

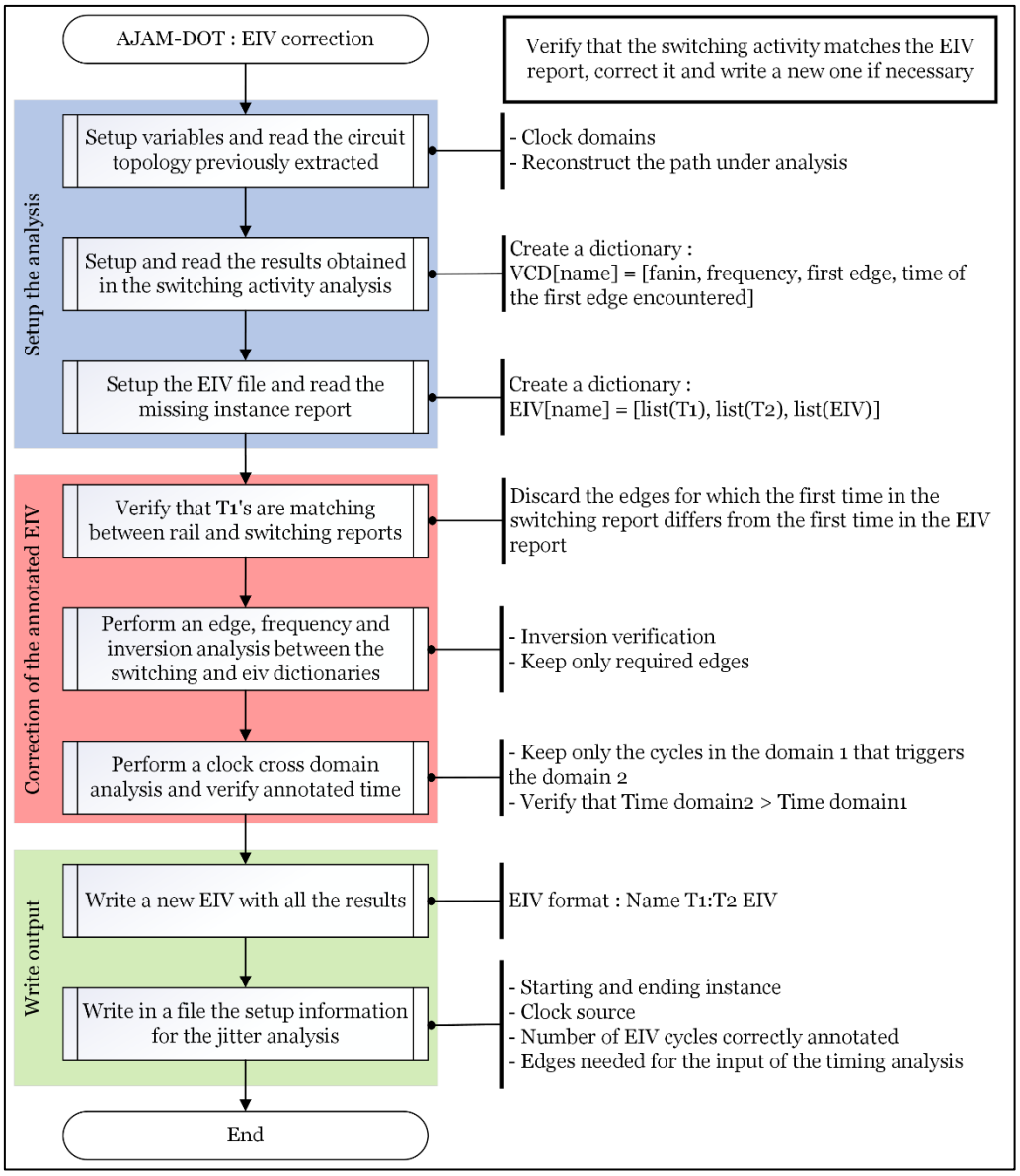

Figure 14 : Python script three, EIV analysis and correction.

Primarily, the analysis environment needs to be set up, which involves reading the results from the .vcd analysis and EIV reports. The analysis path is then reconstructed, ensuring the edges correlate with the circuit topology in cases implying inverters or clock cross domains. Moreover, the frequency and first edge of each instance are stored for later verification.

Secondly, the correlation process begins by verifying that the first switching time of each instance matches the first timing window recorded in the rail report for the same element. This guarantees that the timing resolution of the EIV can be compared with the timing resolution of the .vcd file, confirming that both analyses start from the same time point.

Next, in sequential processes, edges are selected by verifying if an inverter preceded the instance under analysis in the circuit topology. If a wrong edge is analyzed regarding the circuit topology, the current analyzed edge is discarded, and the following edge, spaced by half the frequency, is saved instead. Consequently, the edges being evaluated are always correct regarding the chip circuitry. Additionally, the frequency is used to establish that the spacing between each time window in the EIV report is accurate. Indeed, it has occurred that two EIV windows were so close to each other that it was necessary to implement a time spacing verification to determine which edge is correct and which is a glitch.

Optionally, if two clock domains are engaged in the path under test, a frequency factor verification must be performed. During this stage, the clock trees of both clock domains are correlated to find the frequency factor F between them :

$$F = \frac{f_{generated\ domain}}{f_{generator\ domain}} = \frac{f_{domain2}}{f_{domain1}}\ [Hz] \qquad (5)$$

Where F, defined in Hertz, permits to save only the clock cycles in the first domain that generate the second one. To provide a visual representation of this technique, Figure 15 below illustrates the simulation output of a buffer followed by a D flipflop. It highlights that only one edge must be annotated, as it respects both the presumed rising edge requirement set by the user and the circuit topology, including the clock cross domain.

Moreover, Figure 15 shows that current peaks are happening on the rising and falling edges of the VCD waveforms. Also, EIV shows a tendency to drop when an important current peak, which is logic since the resistance of the grid is constant. Therefore, to satisfy the Ohm's law, during a current spike, there is a voltage drop which depends mostly on the current strength, the surrounding switching activity and associated parasitic components.

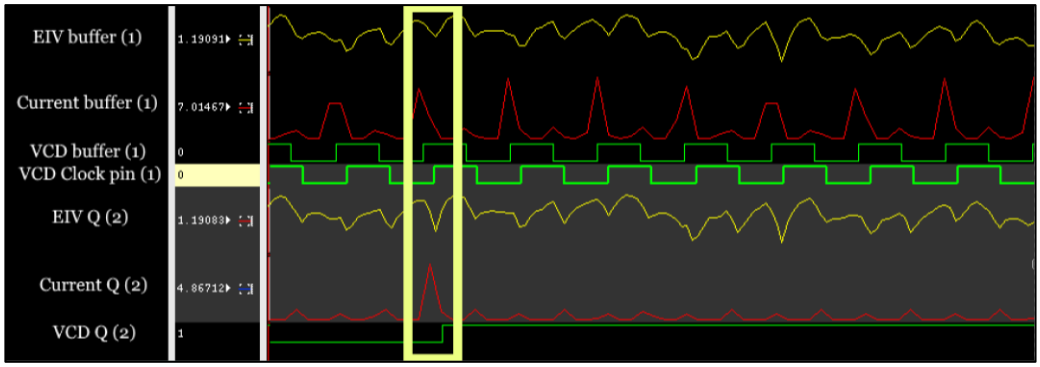

Figure 15 : Clock cross domain EIV annotation, example derived from AltirocA.

Finally, once the correlation between the switching activity and the rail report is completed, a new EIV file considering all modification is written for accurate and circuit coherent timing simulations. Along with it are saved the setup data for the later timing analysis, including the necessary options to indicate to Cadence Tempus for simulating the correct behavior.

In the next section of the AJAM-DOT, the automatic SPICE deck generation will be presented. Additionally, the flaws of the current tools used to create the SPICE decks will be discussed, along with the proposed corrections.

3.4. Step 3 : SPICE deck and corrections

As previously discussed, timing simulations are contingent with the accuracy of the power analysis results. Specifically, while the initial power rail results may be inaccurate regarding the circuit topology and cycles coherency, the AJAM-DOT methodology successfully recreates a reasonable and realistic cycling annotation. Therefore, the flow illustrated in before in Figure 8 moves on with the creation of multiple precise SPICE decks for analogue simulations.

As an aside, any tools that incorporate the key features presented in Table 2 can be used for generating SPICE deck. For example, Cadence Innovus, Voltus, and Tempus are tools that possess the necessary options to create accurate SPICE decks [37].

Software	Can generate SPICE deck ?	Key features
Innovus	Yes	Merge SPICE deck. Report_timing-based path. Supply voltage files read. Path and depths limitations.
Voltus		
Tempus		

Table 2 : SPICE deck crucial features.

In greater depth, it is essential for the tool to be capable of merging SPICE decks, as this will simplifies the analysis of triplicated paths. Additionally, the tool must be able to read the EIV files to incorporate IR drop effects into the SPICE decks. Lastly, it should provide options to precisely control or limit the path under test, ensuring the correctness of the path analyzed.

In the next segment of the document will be covered the SPICE decks working principle. Following this will be discussed the flaws discovered during the development of AJAM-DOT and the solutions found.

3.4.1. SPICE deck working principle.

In the initial part of the discussion regarding SPICE decks, it is essential to provide a brief explanation of their operating principle once again. According to the Tempus Stylus Text Reference Manual 23.10 :

> "While generating a SPICE deck, the [...] command looks for library cells required to run SPICE and adds all cells as ".include" in the generated SPICE netlist. The output SPICE deck includes: Active and passive devices such as Field-Effect Transistors, capacitors, and resistors. Initial conditions and voltages sources. Actual names of the gates, nets, and cell instances, which helps the simulator interpret this information effectively." [37]

In other words, the SPICE deck reads the SPICE model, netlist, subcircuits description, and voltage file provided by the user. Subsequently, an input voltage waveform is derived from the STA and used as the input event, which is the waveform voltage sent on the first cell of the path. Next, the SPICE decks process the circuit topology,

annotating the parasitic elements seen at each stage of the circuit. On top of that, the SPICE decks define measurement statements to monitor delays and arrival times at each stage of the simulation. The difference between writing SPICE deck from there or from the jitter command of Cadence Tempus is the fact that more options are available to modify the circuit's behavior as it should be, rather than having a black box which is difficult to control. Among the added options there are : choosing the measure points, merging SPICE decks, and using the report timing command as the path reference [37]. Although, writing SPICE decks does not allows for easy EIV control as for the analyze jitter command. For example, choosing the starting and ending EIV cycles to annotate [13] is not possible with the "write_spice_deck" command. Therefore, these options have been manually implemented in the TCL script that launches the SPICE decks generation.

In detail, the delays are measured at fifty percent of the voltage supply applied to the cell concerned, this method is called the "mid-point measurement". Taking the measurement at fifty percent of the voltage supply emphasizes again the fact that accurately calculating IR drop effects on voltage is crucial as it will directly modify the voltage supply annotated on each cell. Regarding arrival time, the measurements are also taken at fifty percent of the voltage applied on a cell but with a time constant reference, which is the time at which the first cell switches.

In the subsequent section will be introduced the problems associated with SPICE decks.

3.4.2. SPICE decks issues.

As previously mentioned, SPICE decks are not exempt from complication. However, the issues are now related to clock domain crossing, triplication and input conditions.

Firstly, in the case of clock cross domain, a significant lock has been identified by M. Bergamin : D flipflops may obstruct the creation of SPICE decks following the Q output of the flipflop. The underlying reason is that Cadence suite is not always able to find a path connecting the Q output to the clock source. However, this problem occurs where the first part of the path under test is analogue, preventing the clock pin to be tracked down to the digital flipflop, which was the case for the ASIC's path analyzed during this master thesis.

To present the problem mentioned above, the following Figure 16 portrays an example in which the clock source pin is placed at the output of an analogue phase-shifter followed by a digital multiplexor, buffer, D flipflop, and inverter. In this scenario, the flipflop is effectively "cut" into two segments, resulting in a disconnection between the clock pin and the Q output, as no hardwired path does exist between them.

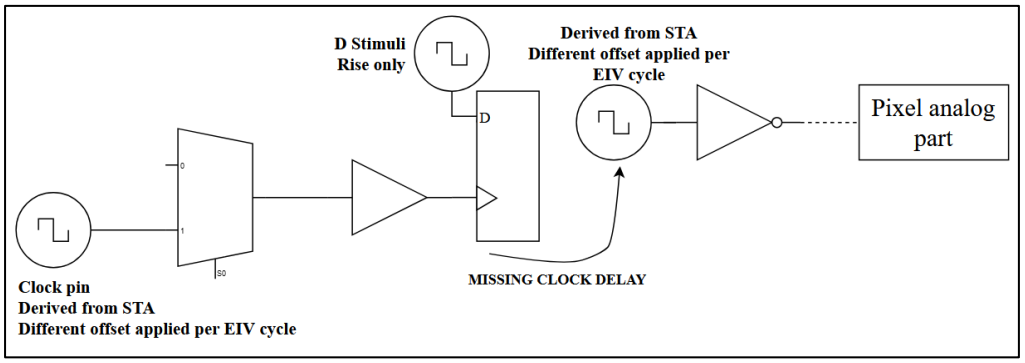

Figure 16 : Cadence D flipflop problem.

Consequently, two SPICE decks must be created : one from the clock source pin to the clock pin of the flipflop, and another one from the Q output to the endpoint of the analyzed path. This approach implies that the delay of the flipflop is not considered. Additionally, generating two SPICE decks introduces inconsistencies in the analysis, as the input event applied on the Q output may not correspond to the event induced by the D signal of the flipflop.

The solution applied to reconnect the two segments of the flipflop involve manually modifying the .lib description of the Q signal of the flipflop, changing it from sequential to combinational. Although this alters the circuit behavior, SPICE simulations are conducted on a single cycle, mitigating potential problems induced by the modification. Additionally, timing and power are not dependent on the sequential nature of the flipflop, as this information is extracted from the .lib table. Therefore, the delay read rest unchanged.

Subsequently to the .lib modification, the clock of the second domain is reset, prompting the software to recompute both domains within a single clock tree. Lastly, another issue is that the D event of the flipflop is challenging to setup properly, indeed, the edge transition always happens at the very beginning of the simulation, while measurements are performed on the last transition of each instance, resulting in a negative simulated delay. Moreover, the D signal is constrained to either a rising transition or no transition at all. However, there is one command option to send a vector that the D and Q signal should follow, but this option has never been successfully utilized. The reason behind that fail is mostly due to a lack of explicit documentation or to software version compatibility. Thus, the solution proposed to correct the D event will be discussed in the next section.

Prior to describing the D event, a related potential bias in analysis setup will be presented : input events differ from each simulated EIV cycle, suggesting different initial conditions per cycle.

To prevent input bias and control the D input of D flip flop, a standalone Python script has been developed to recalculate the SPICE input event and the correct D flipflop input. To shed more light on it, the example illustrated in Figure 17 depicts the input event observed by the first instance in the circuit when 1.2V is applied at 640MHz. It is evident

that the voltage input experiences a voltage jump between all edges. Additionally, the frequency of the input event does not align with the circuit behavior.

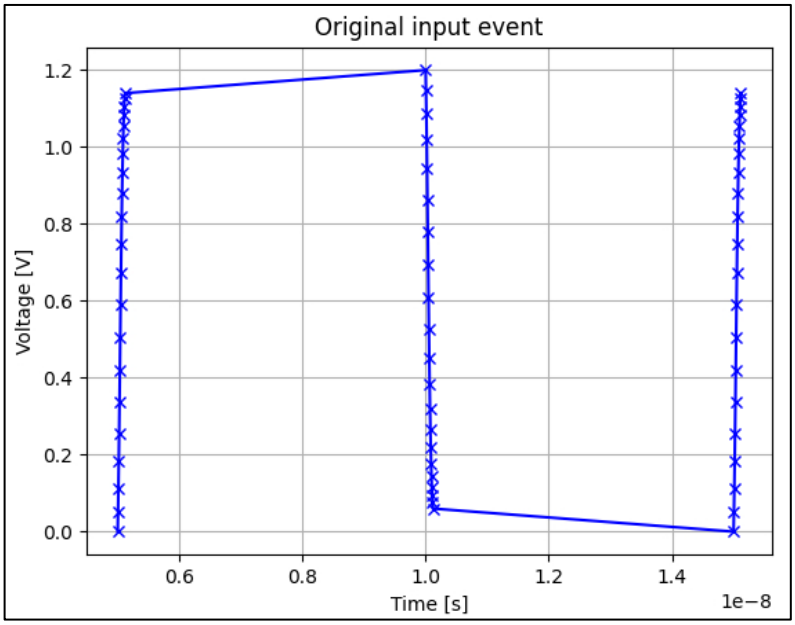

Figure 17 : Example input event.

In greater depth, the issue is related to the fact that the IR drop seen by the first instance influenced the input event. However, the input event should be assumed ideal to monitor only IR drop effects on the path under test, not on the input waveform already.

A proposed solution implies extracting the transition of both rising and falling edge from either the report delay command [38] or from the SPICE decks input event by isolating the rising and falling edges. Then, the transitions are extended using a spline interpolation and used to recreate the input event from them. The second solution was preferred as it did not involve using another command to lead to the same result and it allows for controlling the D stimuli on the same Python standalone script.

To provide more context, spline interpolation is a technique used to create a curve that passes through a given set of points. In this context, the spline interpolation takes the incomplete transition data from the input event and generates a continuous curve that extends up to the required voltage levels [39] as it portrayed in Figure 18.

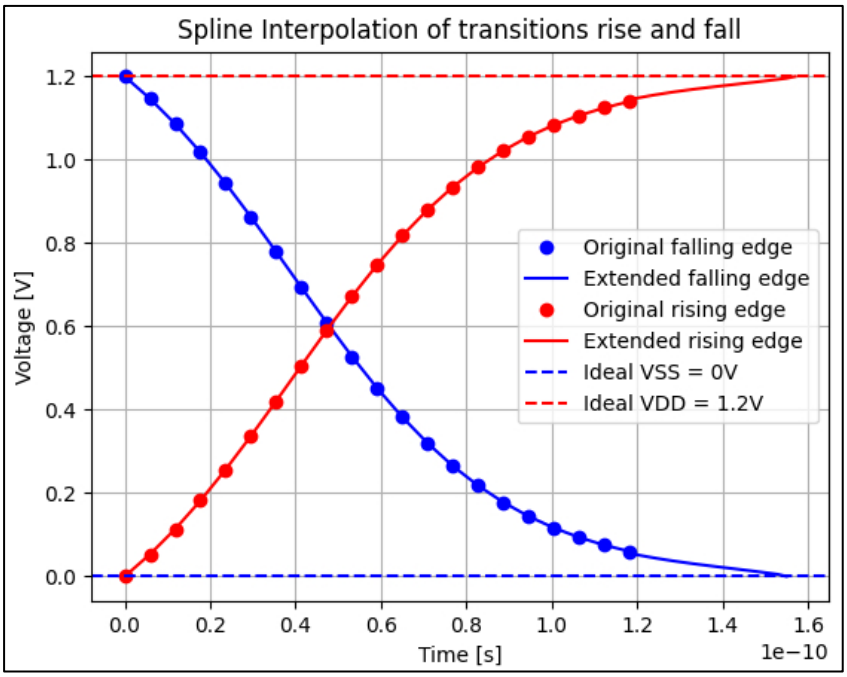

Figure 18 : Spline interpolation to complete the transition time.

Once the transitions are realistically extended, the input event is recreated, considering this time the frequency, as it can be seen on Figure 19.

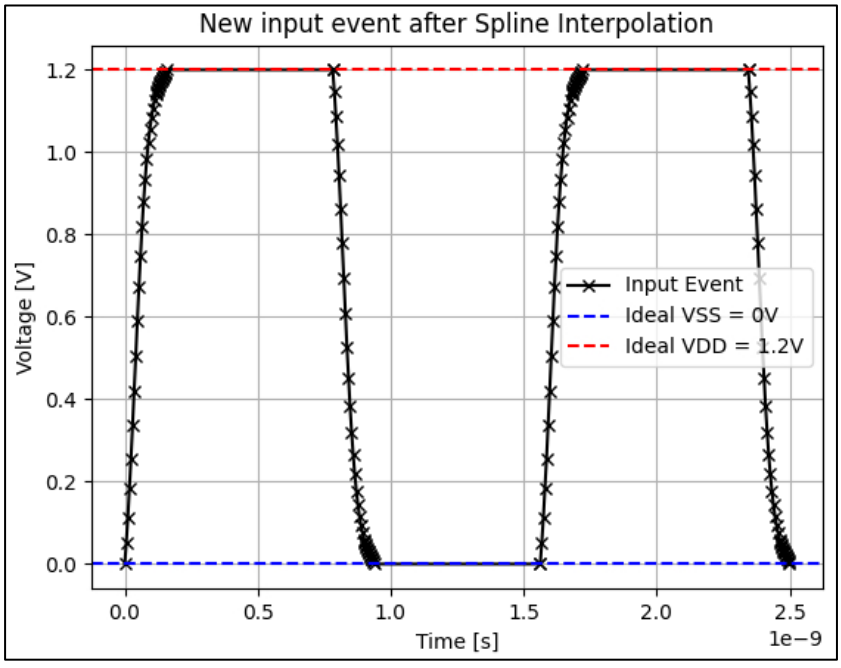

Figure 19 : Example input corrected in frequency and value.

Afterward, the D stimuli which was previously set to a rising edge by default has been modified to fit the newly created input event. Also, to prevent negative timing measurements induced by a wrong edge being tested, the D stimuli is constrained to

occur in the middle of the input event. Therefore, this ensures that the D propagation aligns with either the next rising or falling edge as shown in Figure 20, which correspond to where the measurements are taken to respect setup and hold.

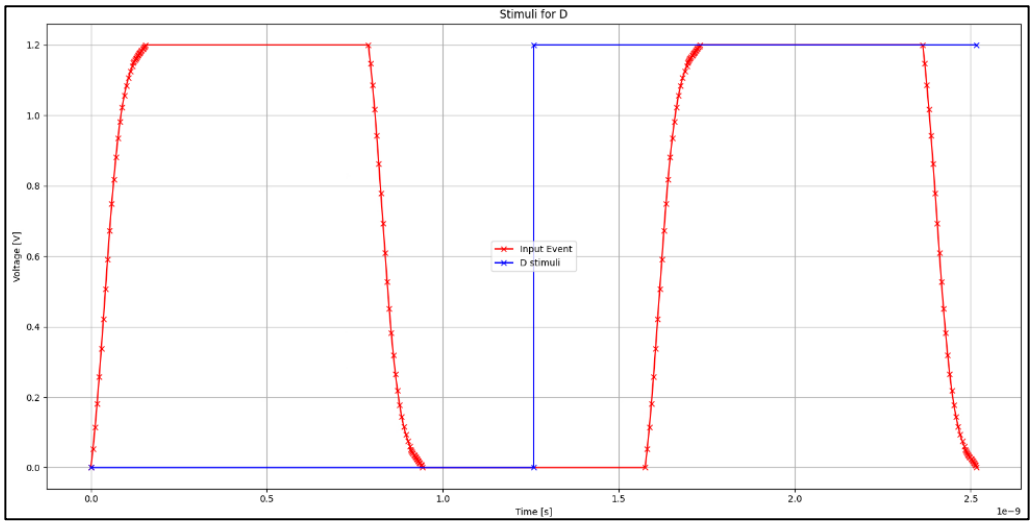

Figure 20 : D stimuli created from the new input event.

Finally, a new schematic illustrated in Figure 21 depicts the correction applied on the circuit example presented in Figure 16. It can clearly be observed that the D flipflop is now combinational, as well the input events are now the same for all instances, and the D stimuli can be controlled. Although, it must be noted once again that the D flipflop behavior modification to combinational has been used as the simplest solution found to force the path generation from the clock source pin to the user specified ending point. Therefore, it may be other options or workaround that were not discovered or applicable in this context.

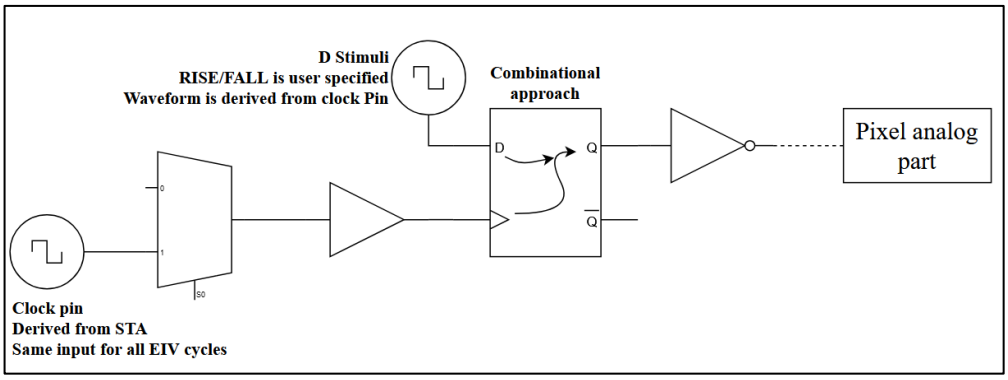

Figure 21 : AJAM-DOT D flipflop workaround.

Regarding the triplicated paths presented in Figure 22, the issue was to merge the deck together. Unexpectedly, by combining the report timing options with the merge SPICE deck feature, the SPICE decks were generated entirely, including all paths. It is important to consider that at least the six first earliest timing report must be taken as

reference in the report timing option [40] to isolate which combination of path two by two are necessary for the voter to propagate the signal.

To offer further context, when two or more registers transmit identical digital values to a voter, the voter will propagate this value to the subsequent part of the path. In detail, the voters consist of three AND-gate and one OR-gate arranged as illustrated on the right side of Figure 22. It can be seen that one voter has six inputs that are arranged in three combinations of two register outputs. In other words, to cover all inputs, at least six simulations are required, one per path. Nevertheless, input events must be corrected for the three registers to prevent bias, and in the case of DFF, they must all be adapted to combinational as well.

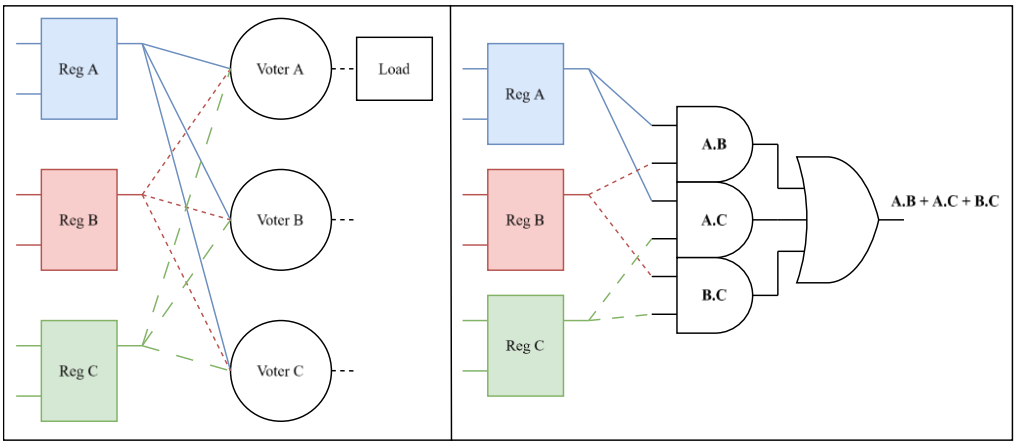

Figure 22 : (left) triplication example, (right) single voter implementation example.

3.5. Python jitter calculations

The last step of the AJAM-DOT methodology is to gather all SPICE decks output and compute jitter from them. Thoroughly, SPICE decks output indicates the arrival time of the signal seen at each circuit stage, plus the delay and slew seen by each cell. The timing information is read by a standalone Python script, which extracts arrival time and delay data for all circuit stages. Hence, it can be verified each cell delay view along with the accumulated delay.

Regarding jitter computation, it has been found that the formulas to calculate jitter are written on the output report of Cadence built-in command to calculate jitter [13]. Additionally, the jitter's definition and formulas defined by NXP Semiconductor [32] are coherent with the Cadence ones. Therefore, it has been decided to keep the Cadence formulas for the TIE jitter, Period (indicated as **PER** in graph) jitter and Cycle-to-Cycle (indicated as **C2C** in graphs) jitter :

$$TIE[i] = Delay[i]_{non-ideal} - Delay[i]_{ideal} \quad (6)$$

$$PER[i] = Delay[i]_{non-ideal} - Delay[i-1]_{non-ideal} \quad (7)$$

$$C2C[i] = Delay[i]_{non-ideal} - 2 \times Delay[i-1]_{non-ideal} + Delay[i-2]_{non-ideal} \quad (8)$$

Where "Delay" is being computed using Cadence Voltus or Tempus, and [i] the clock cycle under analysis. Additional information can be found in appendix D.

As a warning, it must be noted and emphasized that the AJAM-DOT simulations are addressing IR drop-induced jitter only. Therefore, the results obtained must not be considered as the total jitter that the ASIC will experience.

This concludes the AJAM-DOT methodology, which correlates the switching activity with the power reports, sets up the analogue SPICE decks for precise simulations, and computes various jitter calculations. It has been discussed that the input event, D stimuli of D flipflop, and triplicated path represents challenges in generating SPICE decks. However, the solution proposed to modify the .lib D flipflop description, recreate the input event and D stimuli and using six paths as reference for triplicated paths are effective to correct the mentioned problems.

In the next chapter, two ASICs developed at CERN and used to develop AJAM-DOT will be presented. Following this, the application of AJAM-DOT to accurately simulate jitter will be discussed, along with the results collected. Additionally, the reports and the information that can be extracted from them will be explained.

Chapter 4 - Application and results of AJAM-DOT on CERN ASICs

To validate the AJAM-DOT technique, it has been decided to apply it to two ASICs developed at CERN. The first one is the Low Power GigaBit Transceiver (**lpGBT**), which is a clock generator, and the second is AltirocA, which is a pixel-grid detector.

4.1. Application 1 : clock generator - lpGBT

The first ASIC analyzed is lpGBT, which is a 65nm node-based high-performance chip designed for precise data transmission in radiation environment. It can be decomposed into a transmission and receiver side. In the first place, the receiver section of the ASIC will be discussed, then the transmission will be presented.

Inside lpGBT architecture lies a Clock and Data Recovery (**CDR**) and Phase-Locked Loop (**PLL**) circuit. These circuits aim at recovering and generating a high-speed clock from serial data sent in input. The recovering clock ensures that the data is sampled properly to maintain data integrity. Upon receiving input, data are processed in a series of transformation : firstly, it is de-serialized for the data to be processed efficiently, then it is decoded to correct any errors, and finally data are de-scrambled to remove any scrambling applied during transmission [41].

On the transmitter side, lpGBT prepares data for transmission by scrambling it to ensure a DC-balanced signal. Next, data are encoded to prevent transmission errors, and finally, data are serialized for high-speed communication link [41].

A crucial component of lpGBT is the clock management system, which controls and generates programmable user clocks. These clocks can range in frequency from 40MHz to 1.28GHz, with phase adjustments available to ensure frequencies coherency with LHC bunch crossing frequency [41].

Finally, an overview of lpGBT architecture is illustrated below in Figure 23.

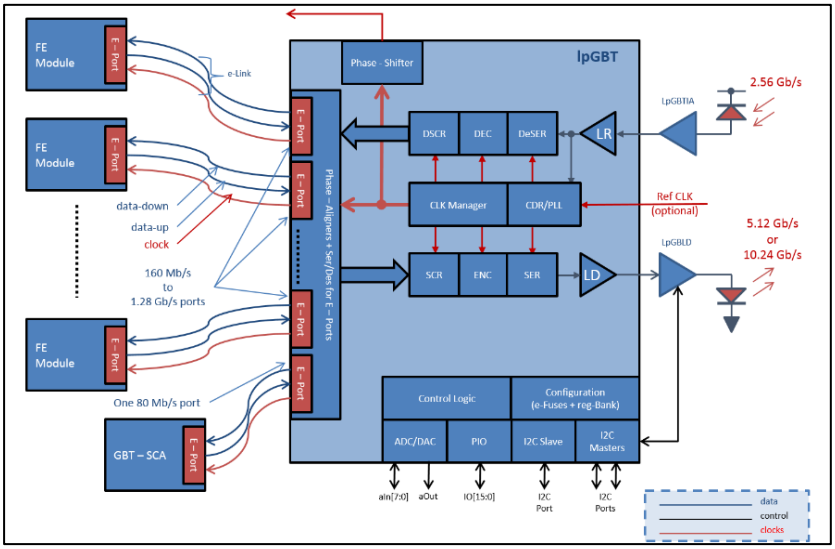

Figure 23 : lpGBT architecture [41].

Prior to digging in simulations, the path analyzed to validate AJAM-DOT has been chosen to be the 1.28GHz clock generation section of the ASIC, as the design team recommended it, and additionally, it implies the most switching activity.

Importantly, real tests results are available for lpGBT, which will be confronted with the simulation results obtained with AJAM-DOT. In greater depth, below are illustrated the TIE jitter results in Table 3 and Figure 24 drawn by the test team on the first lpGBT released version.

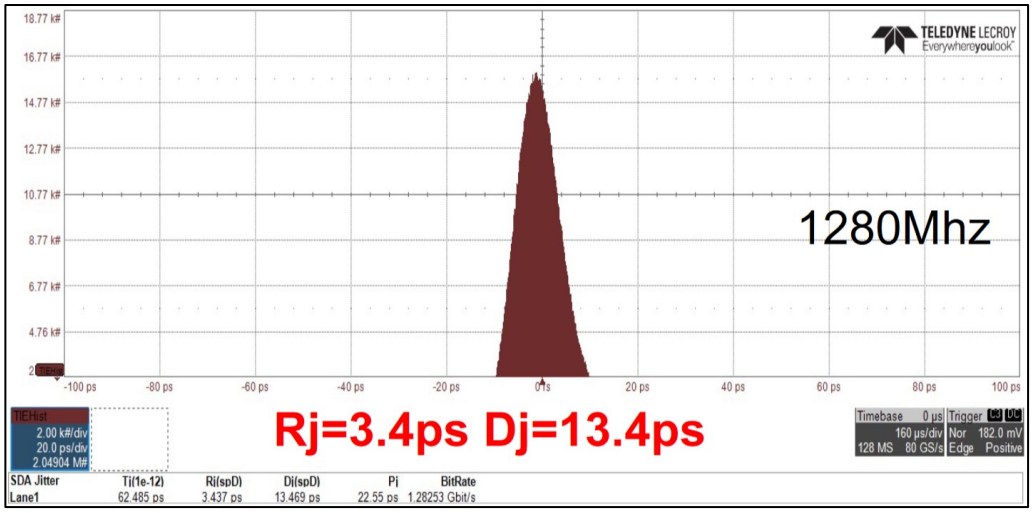

Figure 24 : 1.28 GHz TIE diagram [42].

Jitter nature	TIE jitter [ps]
Worst deterministic jitter (peak-to-peak)	13.4
Worst random jitter (RMS)	3.4

Table 3 : lpGBT TIE results [42].

Both figures and the table above present a 3.4ps RMS random jitter and 13.4ps peak-to-peak deterministic jitter. Regarding IR drop-induced jitter, it must be confronted the simulation results with the worst deterministic jitter as the power supply variations contribute to the deterministic jitter.

As a reminder, deterministic jitter measured includes all deterministic jitter sources : cross talk, electromagnetic interference, device function dependency and simultaneous switching outputs [31]. Again, AJAM-DOT aims at assessing only IR drop-induced jitter, therefore, it will not be possible to simulate the 13.4ps deterministic peak-to-peak jitter exactly. However, the TIE jitter simulated will helps to determine how much IR drop is contributing to the total jitter.

Therefore, in the next section will be presented the simulations performed on lpGBT along with the jitter extracted from them. Additionally, the jitter will be confronted with the mentioned jitter above.

4.2. lpGBT simulation results

Prior to explaining the results, the AJAM-DOT analysis aims at assessing IR drop-induced jitter, hence, it is logical to present the rail analysis results obtained during the original DoT flow before discussing jitter. Additionally, two corners have been analyzed for lpGBT but only the worst corner which is the RC low temperature at -40°C will be presented as it produces the worst simulated jitter. However, results for the typical corner at 25°C can be found in appendix A.

Firstly, Figure 25 presents the "static representation of the worst dynamic IR drop" simulated for lpGBT, on which can be seen that the worst dynamic IR drop represents only two percent of the ideal VDD. Additionally, most of the grid experiences a worst IR drop of less than 1mV voltage drop across the grid. Hence, this suggests that the jitter simulated should be low over all the simulation cycles. Although the result on the scale indicates a 28.42mV "static representation of the worst dynamic IR drop", the worst EIV cycle simulated for lpGBT evaluated is 1.3157V for the worst corner, which represents 0.02412V difference between the IR drop that can be seen on the figure below and the voltage drop simulated. This difference can be explained by the fact that the visual IR drop considers the worst IR drop simulated for both edges as well as the whole ASIC, meanwhile the EIV analysis is focused on rising edges for a specific path and varies over time.

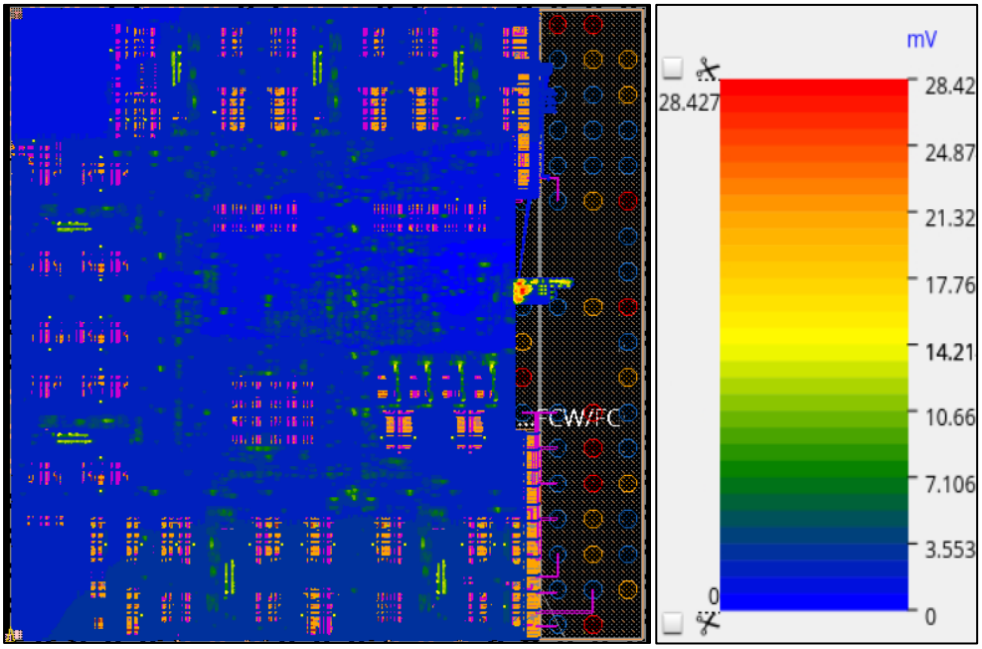

Figure 25 : lpGBT IR drop RC_ltworst corner.

The results above serve as an indication of which regions of the ASIC might justify closer attention. However, IR drop does not necessarily imply the presence of jitter, as a stable IR drop over time would not cause any IR drop-induced jitter. Another way of assessing visually where IR drop-induced jitter would be to create a movie to visualize the voltage drop in real time, which is possible using Cadence built-in commands [43].

Moving on to the jitter analysis and starting from the highest view, the worst arrival time of lpGBT is illustrated in Figure 26. In detail, the y-axis represents the arrival time at the endpoint of the path under analysis, which is a buffer, and the x-axis symbolizes the clock cycles simulated affected by IR drop that are already correlated to the switching activity.

It can be observed that the arrival time difference is closely bounded, with 0.55ns difference between the ideal and worst arrival time, which represents a time increase of 0.551%. Additionally, the maximum EIV difference is 0.0016V, emphasizing on a low variation and evenly distributed voltage across the path. Moreover, by considering the ideal voltage, the maximum drop observed is 0.326%, which is equal to 4.3mV.

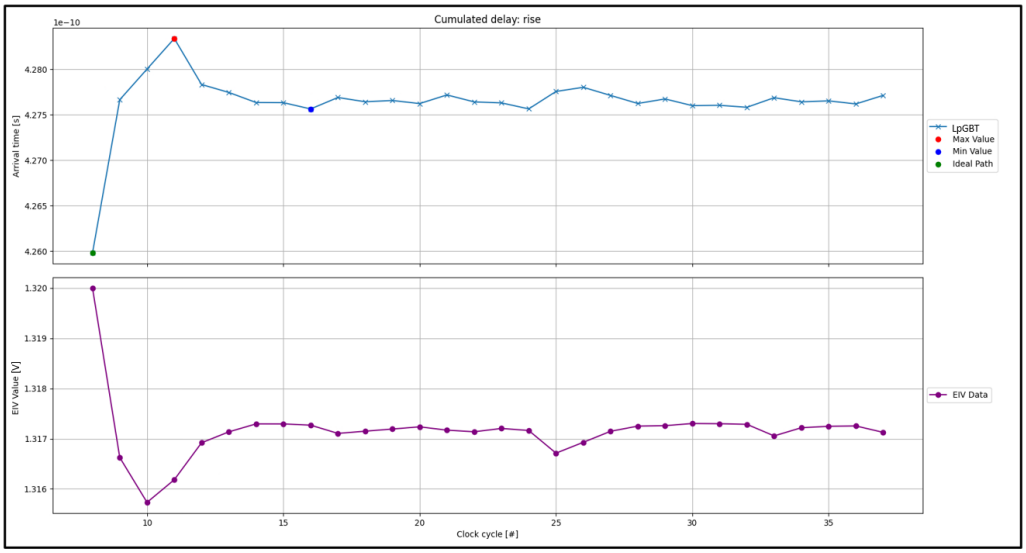

Figure 26 : Arrival time and EIV simulation | lpGBT, lt_rcworst corner.

In greater detail, the delay difference seen by each tested stage between the worst and best arrival time are illustrated in Figure 27. It can be observed a triangle pattern, each positive peaks corresponding to a cell's delay and in between are the signal wire's delay.

From the graph can be determined the worst delay difference at stage 13, which is equal to 0.00932ps and the lowest delay difference seen at stage 14, which is -0.00155ps. Summing all the stages delays, the difference in delays between the longest and shortest arrival time returns a 0.77ps TIE jitter, which confirms a professionally designed PDN, hence, a low jitter.

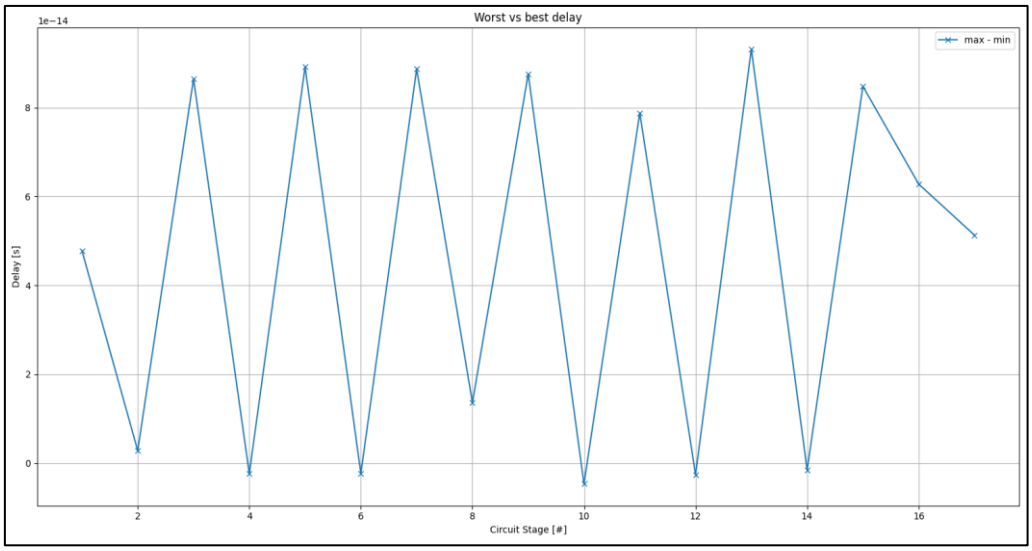

Figure 27 : Delay stage analysis | lpGBT, lt_rcworst corner.

Additionally, a statistical view for jitter is provided in Figure 28. The view is obtained by applying formulas (6), (7), and (8) on the delay measured for the full path. Three types of jitters are included : cycle-to-cycle jitter, TIE jitter, and period jitter. The TIE jitter illustrated includes a normalization to the ideal arrival time reference, which allows for observing the arrival time difference between the ideal and simulated cases along with jitter. The jitter values are extracted by subtracting both whiskers, which for TIE leads to 0.77ps, which is the same as the TIE jitter calculated by summing the stages delays difference together.

Moreover, the cycle-to-cycle and period jitter indicates a low timing variation over the simulation with a maximum period deviation of 0.161% between two consecutive cycles and 0.107% period deviation over the full simulation range. Below is the formula used to estimate the variation percentages :

$$Variation\ percentage = 100 - \frac{jitter(TIE, PER, C2C) + clock\ period}{clock\ period} \times 100 \qquad (9)$$

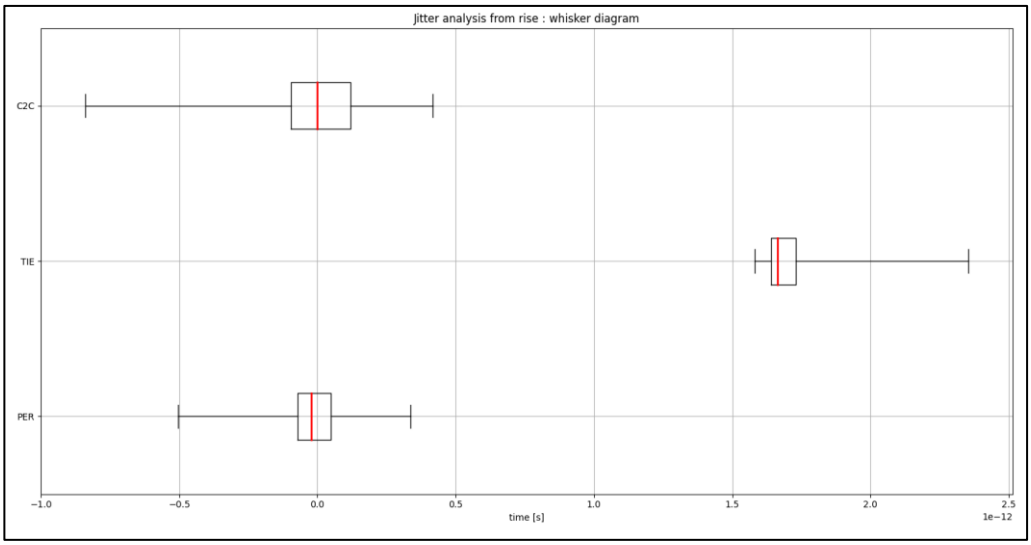

Figure 28 : Jitter whisker diagram | lpGBT, lt_rcworst corner.

To summarize, the worst TIE jitter simulated for the worst-case scenario is 0.77ps. Additionally, the jitter simulated for the typical scenario is 0.75ps as indicated in Appendix A, confirming a stable power delivery network grid regardless of the scenario or operating conditions. On top of that, real tests measurements show a 13.4ps deterministic peak-to-peak jitter [42], which is above the simulations results. However, it must be noted that real tests measurements consider the packaging, all power domains coupling effects and all parasitic elements, which is not the case yet for the simulations. Additionally, the scenario in which the simulation has been evaluated may differs from the real test measurements.

On a side note, the same simulation performed with “analyze_jitter” returned 1ps TIE jitter, which is the lowest resolution available, confirming that AJAM-DOT has a better sub picoseconds timing resolution.

To conclude, the IR drop-induced jitter for the ASIC lpGBT is stable and is not the major source of jitter. However, the AJAM-DOT methodology is not validated completely as the results obtained does not correlate with test results but only to the IR drop, although it confirms that lpGBT PDN is professionally designed by showing that IR drop has only a 0.77ps impact on jitter, which corresponds to 5.7% of the peak-to-peak deterministic jitter measured.

In the next section, it will be introduced AltirocA, which is a pixel-based detector to track particles for the ATLAS experiment.

4.3. Application 2 : pixel-grid based detector - AltirocA

AltirocA is a 130nm node-based ASIC designed for the ATLAS experiment discussed in the introduction. Its primary goal is to precisely measure the time of particle tracks with a 25ps maximum precision including all sources [44] to mitigate pile-up effects. In particle physics, pile-up refers to multiple particles’ collisions happening simultaneously, which complicates the identification and reconstruction of individual

collision events [45]. Hence, preventing pile-up phenomenon is crucial for identifying particles produced in collisions.

Regarding the physical layout of AltirocA, an simplified view in Figure 29 portrays the main fifteen by fifteen-pixel grid and the peripheral that composed AltirocA.

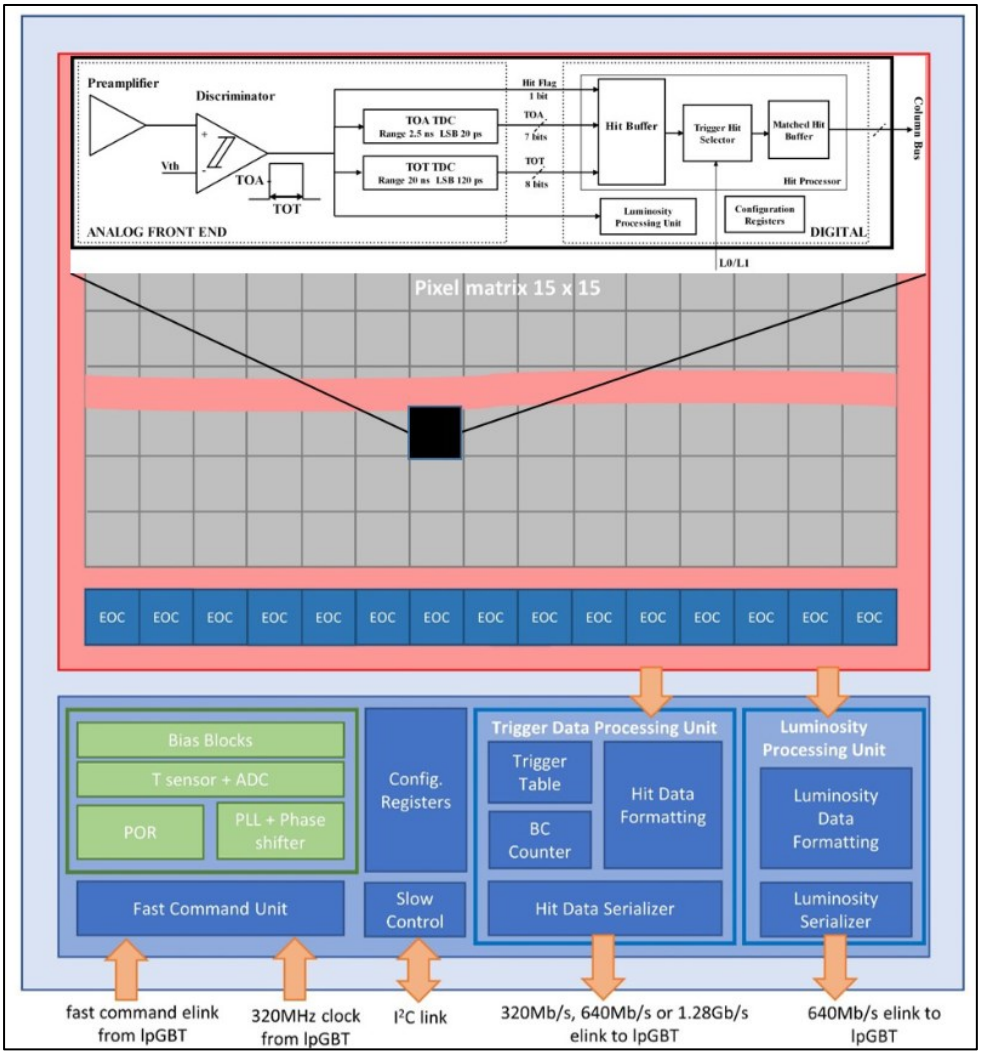

Figure 29 : AltirocA ASIC.

In detail, each single pixel is composed of an analogue front-end which employs Low Gain Avalanche Detector (**LGAD**) to detect a particle strike. This generates for the Minimum Ionizing Particles (**MIP**) a charge of at least four femtocoulombs [6]. Additionally, each LGAD detector is connected to a preamplifier to enhance the signal collected and forward it to a discriminator, which discriminates hits from the background noise. Next, hits are processed to measure the TOA and TOT, which respectively are the time difference between a hit and a reference clock, and the time at which the discriminator sends a zero again [10]. Due to TOA and TOT being measured, AltirocA is considered as a 5D detector, which implies a measurement in time, three

dimensions and energy. The energy is used to correct the time walk effect seen on the TOA measurement.

Once done, both time measurement are sent to two Time-to-Digital Converters (**TDC**) operating on Vernier's delay lines, which compares a starting slow clock signal and fast data signal and return the number of registers required for the fast data to catch up with the clock [46]. Following this, the Back-end section of each pixel processes the data and temporarily stores them in a hit buffer and luminosity buffer. Then, upon receiving a trigger signal, hits are verified to ensure that the read entry matches the hits analyzed, and in the positive case, the hits information are moved to the matched hit buffer waiting to be read [6].

Moving on to the digital peripherals, they are composed of : a trigger data processing unit to format and serialize time for high-speeds transmission ; a luminosity processing unit to handle luminosity data proportional to the instantaneous luminosity of the collisions ; and multiple clock-generator blocks to control the ASIC and ensure clean clock distribution across the chip. Additionally, multiple paths participate in multiple clocks domains. For example, a 640MHz clock is used at the output of a phase shifter right after a PLL to isolate the required phase, then the 640MHz clock is divided into a 40 MHz clock for digital processing and measurements.

Overall, the AltirocA system aims at providing accurate timing and luminosity measurements to upgrade the performance of the ATLAS detector.

Additionally, a realistic 1.8us simulated scenario is illustrated in Figure 30. The top figure presents the waveform generated from the .vcd file, showing particle detections by the pixels. This waveform can be divided into two distinct regions : one without any hits and another with a random hit distribution across the entire ASIC. Correspondingly, the bottom figure depicts the EIV waveform of a component in the path under test. This waveform highlights three regions of interest : a stabilization region, which will be discarded manually as it is a period necessary for calculation to converge ; a "no hit" region, which can be analyzed to assess the ASIC grid's behavior without any events ; and lastly, a region with a high hit activity, used to simulate realistic events and verify the transition between the no-hit state and high hit activity.

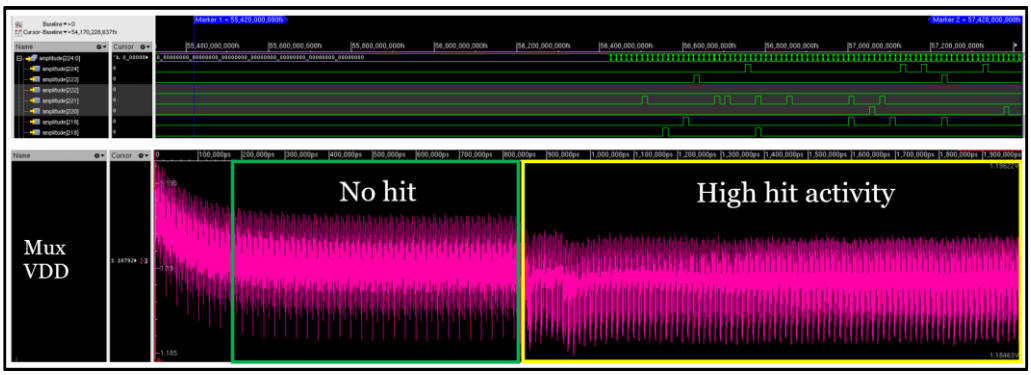

Figure 30 : (top) AltirocA simulation scenario, (bottom) AltirocA EIV example.

From the previous figure, it can be seen that during the hit period, the EIV waveform drops from 1.195V to 1.18463V, demonstrating the importance of an accurate scenario to prevent simulating unrealistic IR drop, hence jitter.

Finally, Figure 31 symbolizes the hit count per pixel over a simulation period of 1.8us, which indicates an even hit distribution scenario over the ASIC.

14	4	2	3	5	8	3	3	7	2	7	2	7	0	2	4
13	4	8	3	1	4	7	2	5	6	1	6	7	3	2	5
12	8	3	3	7	7	2	6	2	3	6	6	6	2	3	9
11	4	6	4	3	3	7	6	1	6	3	3	7	1	4	1
10	7	8	5	6	4	8	4	2	3	7	5	3	6	4	4
9	1	6	4	8	3	5	3	2	1	4	5	4	5	6	4
8	3	6	6	5	7	4	5	5	7	3	2	7	**2**	**15**	5
7	4	6	7	4	7	2	8	9	4	2	5	1	2	9	2
6	4	6	7	3	7	3	7	5	6	2	4	6	5	8	3
5	8	8	7	4	4	5	8	5	7	9	6	8	5	4	7
4	7	6	8	5	2	6	8	7	6	7	6	12	6	7	7
3	9	4	5	10	9	4	7	7	3	6	5	6	5	7	11
2	5	8	5	9	5	6	7	5	3	7	5	10	5	6	4
1	5	5	4	4	3	6	5	5	9	5	2	6	5	6	4
0	5	4	10	5	6	8	9	4	6	6	4	7	5	3	0
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Figure 31 : Simulation scenario, pixel hit per pixel over 2us.

In the next section will be covered the results of AJAM-DOT analysis performed on AltirocA. The path to be analyzed is a chain of buffers and inverters that distributes the clocks from the output of the phase shifter to each pixel front-end. Therefore, two clock domains are crossed : a 640MHz clock from the phase shifter to a clock divider register, and a 40MHz clock from the output of the clock divider register until the analogue front-end of each pixel, which together corresponds to sixty-two stages to analyze.

4.4. AltirocA simulations results

Primarily, the rail analysis results will be discussed, then the jitter results will be presented.

Firstly, Figure 32 portrays the worst IR drop simulated for the full pixel grid on which can be seen that IR drop degradation increases with the pixel distance to the peripheral. This suggests that the jitter simulated will be worse for the top of each column. Although, like lpGBT, the representation is a static view of a dynamic phenomenon. Therefore, results must be considered as a trend only. The result on the scale indicates a 57mV worst static representation of the dynamic IR drop, the worst EIV cycle simulated on each pixel path is 1.1574mv, which is 15mV smaller than the value visually indicated. This difference can be explained by the fact that the IR drop that can be seen below considers rising and falling edges on the full ASIC, meanwhile the EIV analysis is focused on rising edges for the given paths.

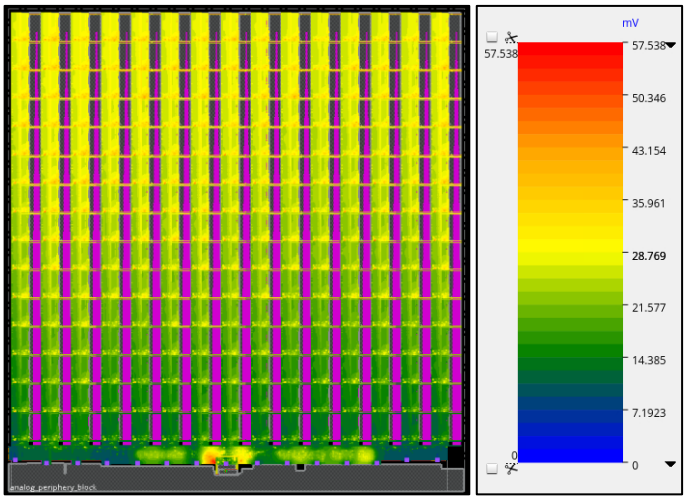

Figure 32 : AltirocA, typical corner IR drop.

Prior to digging into the analysis, AltirocA is a large 1.3 per 1.3cm ASIC, and each pixel must be verified, resulting in at least 225 simulations to perform. Therefore, it has been decided to narrow the simulations results discussion to column zero and detailing only pixel [0/0]. However, 53% of the ASIC has been assessed as it will be presented later in the document.

Starting from the highest view, the arrival time of column zero is illustrated in Figure 33. As for Figure 26 presented for lpGBT, the y-axis represents the arrival time at each column zero pixel's analogue front-end, and the x-axis symbolize the clock cycles simulated affected by IR drop, already correlated to the switching activity. The column's skew is equal to 70ps, which implies a well distributed clock across the column. It can be identified as well that all pixels are following the exact same tendency for all cycles, resulting in a constant skew throughout the simulation.

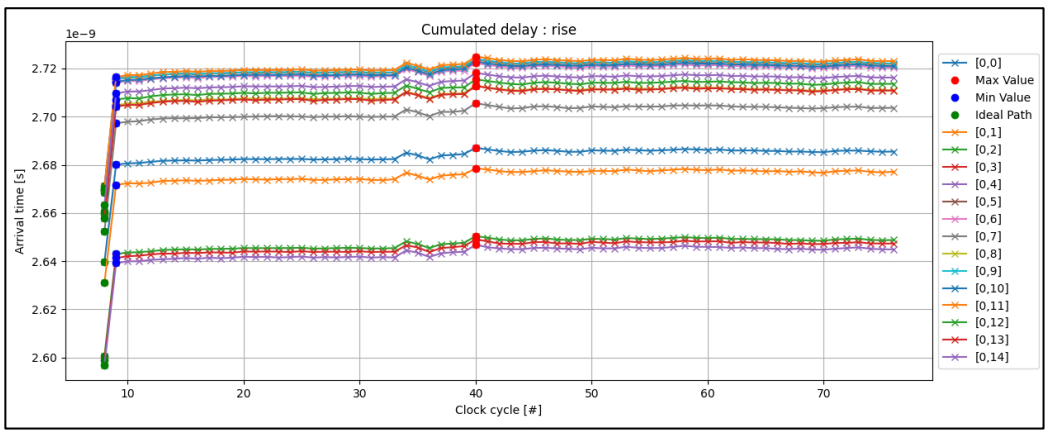

Figure 33 : Arrival delay | AltirocA, typ corner, column zero.

Narrowing the simulation results to pixel [0/0], it can be seen on Figure 34 that the EIV and arrival time curves portray a mirrored tendency, suggesting that a larger IR drop implies a longer arrival time. However, it emphasizes as well that IR drop is cycle dependent, leading to uncertainty in arrival time, therefore TIE jitter.

From the graph, by subtracting the maximum and minimum arrival time, the TIE jitter is approximated to 7ps.

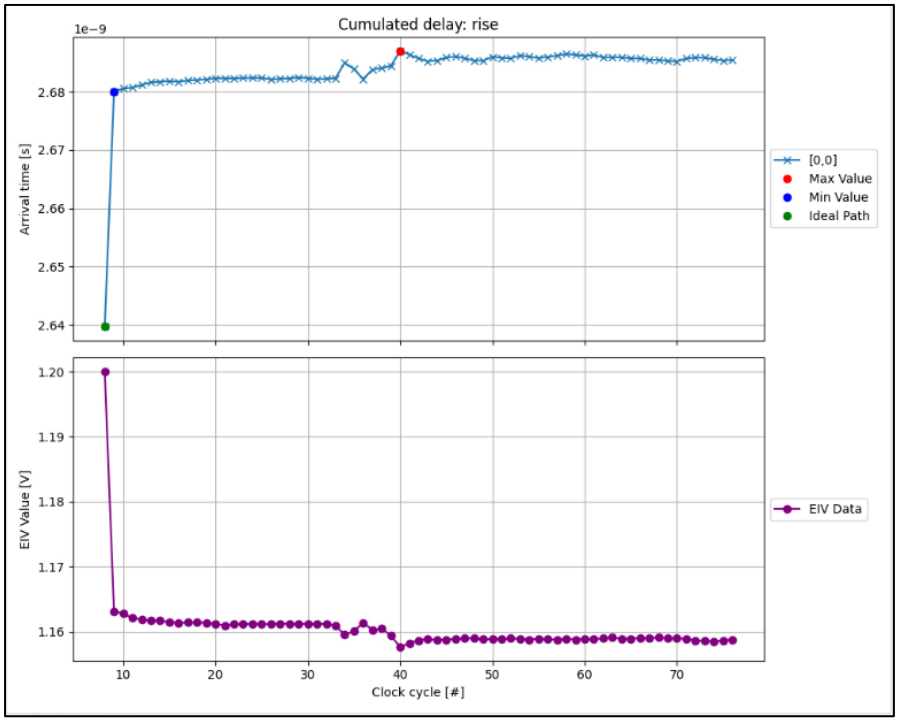

Figure 34 : Arrival time and EIV simulation | AltirocA, typ corner, pixel [0/0].

In greater detail, the delay difference seen by each cell between the worst and best arrival time are illustrated in Figure 35. To be precise, two areas can be distinguished from the graph : from 1 to 5 is the first clock domain with a frequency of 640MHz and a maximum jitter of 0.272ps, and from 6 to 62 is the second clock domain with a frequency of 40MHz and a maximum jitter of 0.584ps.

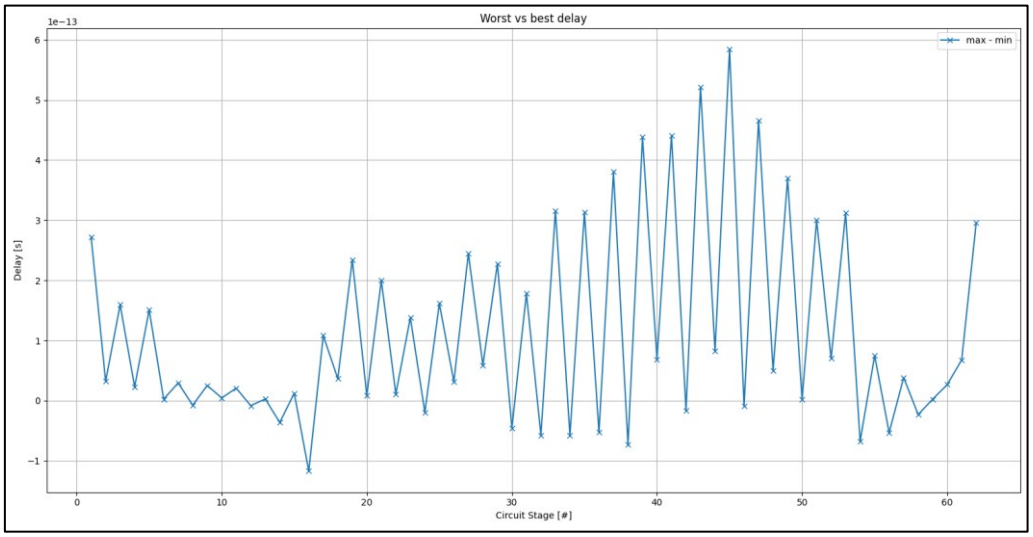

Figure 35 : Delay stage analysis | AltirocA, typ corner, pixel [0/0].

Additionally, a statistical view for jitter is provided in Figure 36 by applying formulas (6), (7), and (8) on the accumulated delay on the path for each cycles. The TIE jitter extracted by subtracting both whiskers is equal to 6.95ps, which is the same as the TIE jitter calculated by summing the stages delays difference together.

Differently from lpGBT simulations results, it cannot be determined any straightforward information from the cycle-to-cycle and period jitter as these results encompass both clock domains. Therefore, it cannot be calculated each clock period and cycle-to-cycle variations independently from a global result. However, it still suggests that the path does not experience large clock variations.

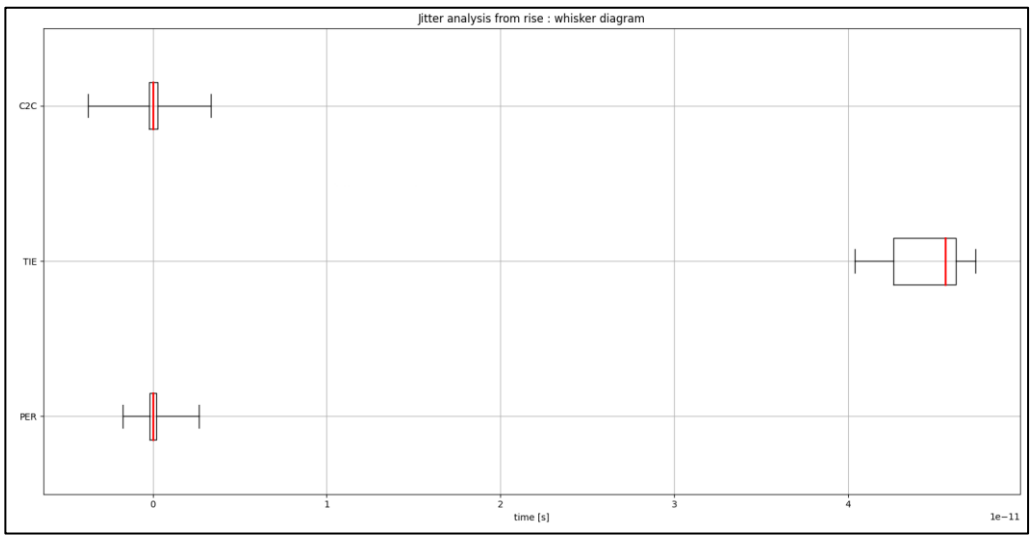

Figure 36 : Jitter whisker diagram | AltirocA, typ corner, pixel [0/0].

To conclude about AltirocA pixel [0/0], the TIE jitter simulated considering IR drop effect is 6.95ps. On top of that, cycle-to-cycle and period jitter advocate a stable clock in time and frequency simulated over seventy-six clock cycles.

However, it must be noted that the jitter analysis lacks real tests measurements to be confronted with the simulations. Therefore, the results must be criticized by stating that only one power domain has been analyzed for AltirocA, in a single typical corner, without taking the packaging parasitic elements into consideration. The reason behind that is that AltirocA is a large ASIC that requires a full day to perform a single rail analysis and requires more than a week for simulating all the pixels. Also, AJAM-DOT is still not matured enough to embed packaging and power domains coupling effects. Additionally, the simulation has not been performed on the whole of ASIC due to the forementioned lack of time.

Nevertheless, Figure 37 portrays the pixel hit cumulated count on each pixel per column. The interesting side of this view is to symbolize and foresee which pixel will present the most TIE jitter. Hence, it can be observed that the further the pixel is from the peripheral, the larger the jitter is to be expected.

14	78	86	81	79	79	76	88	71	72	75	66	97	57	86	70
13	74	84	78	74	71	73	85	64	70	68	64	90	57	84	66
12	70	76	75	73	67	66	83	59	64	67	58	83	54	82	61
11	62	73	72	66	60	64	77	57	61	61	52	77	52	79	52
10	58	67	68	63	57	57	71	56	55	58	49	70	51	75	51
9	51	59	63	57	53	49	67	54	52	51	44	67	45	71	47
8	50	53	59	49	50	44	64	52	51	47	39	63	40	65	43
7	47	47	53	44	43	40	59	47	44	44	37	56	38	50	38
6	43	41	46	40	36	38	51	38	40	42	32	55	36	41	36
5	39	35	39	37	29	35	44	33	34	40	28	49	31	33	33
4	31	27	32	33	25	30	36	28	27	31	22	41	26	29	26
3	24	21	24	28	23	24	28	21	21	24	16	29	20	22	19
2	15	17	19	18	14	20	21	14	18	18	11	23	15	15	8
1	10	9	14	9	9	14	14	9	15	11	6	13	10	9	4
0	5	4	10	5	6	8	9	4	6	6	4	7	5	3	0
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Figure 37 : AltirocA cumulated pixel hits over each column.

Even if the full ASIC has not been simulated, Figure 38 below presents the results obtained for the tested columns, indicating a worst-case jitter of 8.97ps for pixel [4/13]. On top of that, it can be observed that the jitter heatmap correlates with the tendency seen in Figure 32 and Figure 37, implying that the distance to the clock source plays a significant role in the total TIE jitter, with a maximum difference of 1.45ps on column one from the first pixel to the last one.

14	8.40E-12		8.70E-12	8.62E-12	8.63E-12		8.30E-12	8.14E-12	8.07E-12	8.02E-12
13	8.38E-12		8.71E-12	8.97E-12	8.62E-12		8.29E-12	8.09E-12	8.16E-12	7.95E-12
12	8.32E-12		8.67E-12	8.91E-12	8.58E-12		8.32E-12	8.17E-12	8.17E-12	7.96E-12
11	8.46E-12		8.70E-12	8.50E-12	8.57E-12		8.26E-12	8.20E-12	8.11E-12	8.04E-12
10	8.39E-12		8.60E-12	8.51E-12	8.48E-12		8.26E-12	8.13E-12	8.03E-12	7.90E-12
9	8.38E-12		8.59E-12	8.57E-12	8.51E-12		8.22E-12	8.16E-12	8.07E-12	8.00E-12
8	8.16E-12		8.31E-12	8.46E-12	8.24E-12		7.99E-12	7.95E-12	7.81E-12	7.78E-12
7	8.22E-12		8.57E-12	8.31E-12	8.52E-12		8.18E-12	7.98E-12	8.05E-12	7.83E-12
6	8.11E-12		8.18E-12	8.40E-12	8.16E-12		7.89E-12	7.90E-12	7.76E-12	7.68E-12
5	8.07E-12		8.27E-12	8.38E-12	8.12E-12		7.90E-12	7.83E-12	7.68E-12	7.58E-12
4	7.58E-12		7.80E-12	7.64E-12	7.75E-12		7.52E-12	7.30E-12	7.21E-12	6.97E-12
3	7.76E-12		7.62E-12	7.61E-12	7.63E-12		7.43E-12	7.49E-12	7.07E-12	7.33E-12
2	7.41E-12		7.80E-12	7.64E-12	7.77E-12		7.55E-12	7.53E-12	7.22E-12	7.17E-12
1	6.78E-12		7.43E-12	7.46E-12	7.40E-12		7.17E-12	7.07E-12	6.91E-12	6.75E-12
0	6.95E-12		7.33E-12	7.21E-12	7.26E-12		7.13E-12	7.10E-12	6.86E-12	6.76E-12
	0	1 and 2	3	4	5	6 to 10	11	12	13	14

Figure 38 : AltirocA jitter heatmap.

Finally, IR drop-induced jitter simulated for AltirocA's pixels contributes to a maximum of 8.97ps jitter. However, it requires to be evaluated once the ASIC is manufactured. Additionally, the simulation must be performed again when the method includes features to assess packaging effect, power domain coupling effects and on different corners.

Chapter 5 - Conclusion and Future Work

In the HEP community, ASIC designs have become harder than ever due to the accuracy requirements being elevated as future detectors and electronics systems are developed. Indeed, ECFA roadmap for future accelerators defined that jitter and power consumption are becoming critical limitations for further LHC upgrades. Additionally, the EP-ESE-ME section has discovered that the current framework to assess IR drop-induced jitter lacks accuracy and important analysis features such as clock cross domains handling and triplicated path management. On top of that, it has been determined that simulations using the framework were missing an important correlation between the switching activity derived from the DoT methodology and the power analyses performed on it, leading the jitter analyses using the power results to be incorrect. In that context the advanced jitter analysis methodology for digital-on-top, AJAM-DOT, has been developed.

AJAM-DOT has shown potential in addressing the current challenges and limitations of simulating and assessing IR drop-induced jitter within the digital-on-top methodology. By utilizing Cadence tools such as Innovus, Voltus and Tempus, guided by seven Python and three TCL scripts, AJAM-DOT effectively manages complex simulations like clock cross domains and triplicated paths, which are difficult to oversee with the current jitter analysis available in the DoT approach. This is achieved through correcting D flip-flop handling, adapting liberty files to change flip-flop behavior from sequential to combinational, normalizing SPICE simulation input events to a clock reference to avoid initial condition bias, and most importantly, correlating dynamic IR drop reports with realistic switching activity to ensure accurately that simulations are transition edge coherent for timing analysis, which is a significant advancement. Additionally, AJAM-DOT offers a sub picosecond timing resolution which is not the case for the current state-of-the-art approach.

Its application to the lpGBT clock generator ASIC revealed a contribution from IR drop-induced jitter of 0.77ps out of the 13.4ps measured peak-to-peak deterministic jitter. Although there are inconsistencies between simulation and reality, the application of AJAM-DOT on lpGBT confirmed the well-designed PDN, which contributed to only 5.4% to the total measured deterministic jitter, which was expected by the design team.

Comparably, the application of AJAM-DOT on the AltirocA particle detector ASIC demonstrated a maximum IR drop-induced TIE jitter of 8.97ps. This result suggests that the clock distribution and PDN in AltirocA may requires modifications if tests results present a critic deterministic jitter. Additionally, the absence of real test data and partial simulation coverage limits the overall validation of AJAM-DOT.

To conclude, AJAM-DOT provides detailed and reliable jitter analysis within simulations, however, it misses replicating real-world conditions due to the lack of external factors inclusion such as packaging interactions and power domain coupling. This highlights the need for further refinement of the methodology. In the long-term development, this excluded information may lower the precision results, therefore, they must be implemented.

To make AJAM-DOT a complete jitter analysis tool, future work should focus on upgrading simulation accuracy, applying it to more tested ASICs, and mature the Python scripts to make them practicable for the user to manage. Regardless of the current limitations, AJAM-DOT will provide great insight into ASIC design for the high-energy physics community and specifically for CERN future accelerators. However, AJAM-DOT is on the promise of becoming a significant tool to master for future ASIC development, even outside the HEP community.

References

[1] CERN, "Where did it all begin?," 2024. [Online]. Available: https://home.cern/about/who-we-are/our-history. [Accessed 9 August 2024].

[2] E. Lopienska, "The CERN accelerator complex," 3 February 2022. [Online]. Available: https://cds.cern.ch/images/CERN-GRAPHICS-2022-001-1. [Accessed 9 August 2024].

[3] CERN, "ATLAS," 2024. [Online]. Available: https://home.cern/science/experiments/atlas. [Accessed 9 August 2024].

[4] ATLAS EXPERIMENT, "Detector & Technology," [Online]. Available: https://atlas.cern/Discover/Detector. [Accessed 12 August 2024].

[5] ATLAS EXPERIMENT, "Trigger and Data Acquisition," CERN, [Online]. Available: https://atlas.cern/Discover/Detector/Trigger-DAQ. [Accessed 12 August 2024].

[6] N. M. L. R. L. s. N. S.-M. A. S. R. Casanova, "HGTD ALTIROC ASIC : Specifications for HL-LHC," 2021.

[7] CERN, "ESE OVERVIEW," 2024. [Online]. Available: https://ep-ese.web.cern.ch/eseoverview. [Accessed 9 August 2024].

[8] the Detector R&D Roadmap Process Group of the European Committe for Future Accelerators, The 2021 ECFA Detector Research and Development Roadmap, CERN, 2021.

[9] S. Baron, "4D Techniques DRDT7.3," 2023.

[10] G. Bergamin and A. Soulier, "A simulation methodology for establishing IR-drop-induced clock jitter for high precision timing ASICs.," TWEPP2023, 2023.

[11] J. Christiansen, "TDC ARCHITECTURES IN ASIC'S," 2011.

[12] Cadence Design Systems, "Dynamic Rail Analysis," April 2024. [Online]. Available: https://support.cadence.com/apex/techpubDocViewerPage?xmlName=voltusugcom.xml&title=Voltus%20Stylus%20Common%20UI%20User%20Guide%20--%20Dynamic%20Power%20and%20IR%20Drop%20Analysis%20-%20Dynamic%20Power%20and%20IR%20Drop%20Analysis&hash=pgfId-1014799&c_ve. [Accessed 2 August 2024].

[13] Cadence Design Systems, "analyze_jitter," April 2024. [Online]. Available: https://support.cadence.com/apex/techpubDocViewerPage?xmlName=tempustxtcmdref.xml&title=Tempus%20Text%20Command%20Reference%20--%20analyze_jitter%20-%20analyze_jitter&hash=&c_version=23.11&path=tempustxtcmdref/tempustxtcmdref23.11/analyze_jitter.html. [Accessed 9 August 2024].

[14] M. Aparicio, M. Comte, F. Azaïs, Y. Bertrand, M. Renovell, J. Jiang, I. Polian and B. Becker, "An IR-Drop Simulation Principle Oriented to Delay Testing," *DCIS'12: 27th Conference on Design of Circuits and Integrated Systems,* 2012.

[15] A. Vakil, H. Homayoun and A. Sasan, "IR-ATA: IR Annotated Timing Analysis, A Flow for Closing the Loop Between PDN design, IR Analysis & Timing Closure," 2019.

[16] G. Bergamin and A. Pierre Soulier, "A simulation methodology for establishing IR-drop-induced clock jitter for high precision timing ASICs," *Journal of Instrumentation,* vol. 19, 16 February 2024.

[17] Cadence Design Systems, "Digital IC Design Fundamentals v2.0 (Online)," Cadence Design Systems, [Online]. Available: https://cadence.docebosaas.com/learn/course/1703/digital-ic-design-fundamentals-v20-online?hash=532d22b4b567b3c47257e3ea84c1e5c9747a675e&generated_by=309814.

[18] G. Sassatelli, *Architectures & Hardware modeling languages WS,* Montpellier, 2019 - 2020.

[19] J. Tech, "Terminology," 2024. [Online]. Available: https://vhdlwhiz.com/terminology/elaboration/. [Accessed 21 July 2024].

[20] Cadence Design Systems, "Tempus Signoff Timing Analysis and Closure with Styus Common UI v23.1," 2024. [Online]. Available: https://support.cadence.com/apex/CosLms_DoceboPage?deeplink=/learn/global-search/STA%20signoff%20or%20early#q=STA%20signoff%20or%20early&t=All&searchboxDropdown=m. [Accessed 14 August 2024].

[21] Team VLSI, "LIB File in Physical Design | Liberty File | .lib file," 8 Mai 2020. [Online]. Available: https://teamvlsi.com/2020/05/lib-and-lef-file-in-asic-design.html. [Accessed 02 August 2024].

[22] Cadence Design Systems, "What are the contents of Cell Power Grid Views (PGV) and how they are used in rail analysis," 10 May 2024. [Online]. Available:

https://support.cadence.com/apex/ArticleAttachmentPortal?id=a1Od000000onXAOEA2&pageName=ArticleContent. [Accessed 2 August 2024].

[23] J. Prinzie, "Lecture 4 : Power," in *Digital Chip Design*, Leuven, 2023.

[24] Cadence Design Systems, "Types of Static Average Power," April 2024. [Online]. Available: https://support.cadence.com/apex/techpubDocViewerPage?xmlName=voltusugcom.xml&title=Voltus%20Stylus%20Common%20UI%20User%20Guide%20--%20Vectorless%20Methodology%20-%20Vectorless%20Methodology&hash=pgfId-999387&c_version=23.11&path=voltusUGcom/voltusUGcom2. [Accessed 2 August 2024].

[25] Cadence Design Systems, "Voltus™ Power-Grid Analysis and Signoff Engineer Explorer Series v23.1," Cadence Design Systems, 2024. [Online]. Available: Restricted URL. [Accessed 30 July 2024].

[26] Cadence Design Systems, "Power and Rail Analysis Using Voltus 20.13 (Signoff Power Analysis)," March 2024. [Online]. Available: https://support.cadence.com/apex/ArticleAttachmentPortal?id=a1O0V00000912FMUAY&pageName=ArticleContent. [Accessed 4 August 2024].

[27] Cadence Design Systems, "Probability-Based Vectorless Methodology," April 2024. [Online]. Available: https://support.cadence.com/apex/techpubDocViewerPage?xmlName=voltusugcom.xml&title=Voltus%20Stylus%20Common%20UI%20User%20Guide%20--%20Vectorless%20Methodology%20-%20Vectorless%20Methodology&hash=pgfId-999387&c_version=23.11&path=voltusUGcom/voltusUGcom2. [Accessed 2 August 2024].

[28] Cadence Design Systems, " State-Propagation-Based Vectorless Methodology," April 2024. [Online]. Available: https://support.cadence.com/apex/techpubDocViewerPage?xmlName=voltusugcom.xml&title=Voltus%20Stylus%20Common%20UI%20User%20Guide%20--%20Vectorless%20Methodology%20-%20Vectorless%20Methodology&hash=pgfId-999387&c_version=23.11&path=voltusUGcom/voltusUGcom2. [Accessed 2 August 2024].

[29] Cadence Design Systems, " Vector-Driven Methodology," April 2024. [Online]. Available: https://support.cadence.com/apex/techpubDocViewerPage?xmlName=voltusug.xml&title=Voltus%20IC%20Power%20Integrity%20Solution%20User%20Guide%20--%20Vector-Driven%20Methodology%20-%20Vector-

Driven%20Methodology&hash=pgfId-999300&c_version=23.11&path=voltusUG. [Accessed 2 August 2024].

[30] Cadence Design Systems, " IR Drop Effect on Delays," April 2024. [Online]. Available: https://support.cadence.com/apex/techpubDocViewerPage?xmlName=voltusugcom.xml&title=Voltus%20Stylus%20Common%20UI%20User%20Guide%20--%20IR%20Drop%20Effect%20on%20Delays%20-%20IR%20Drop%20Effect%20on%20Delays&hash=pgfId-998794&c_version=23.11&path=voltusUG. [Accessed 2 August 2024].

[31] J. N. Tripathi, V. K. Sharma and H. Shrimali, "A Review on Power Supply Induced Jitter," *IEEE TRANSACTIONS ON COMPONENTS, PACKAGING AND MANUFACTURING TECHNOLOGY,* vol. 9, March 2019.

[32] NXP Semiconductor, "Understanding SYSCLK Jitter," February 2010. [Online]. Available: https://www.nxp.com/docs/en/application-note/AN4056.pdf. [Accessed 27 June 2024].

[33] ANALOG DEVICES, "An Introduction to Jitter in Communications Systems," 6 March 2003. [Online]. Available: https://www.analog.com/en/resources/technical-articles/an-introduction-to-jitter-in-communications-systems.html. [Accessed 20 August 2024].

[34] M. Bartock, J. Brule, Y.-S. Li-Badoub, S. Lightman, J. McCarthy, K. Meldrof, K. Reczek, D. Northrip, A. Scholz and T. Suloway, "Foundational PNT Profile:Applying the Cybersecurity Framework for the Responsible Use of Positioning, Navigation, and Timing (PTN) Services," U.S. Department of Commerce, 2023.

[35] Cadence Design Systems, "Clock Tree Network Structure Reporting," March 2024. [Online]. Available: https://support.cadence.com/apex/techpubDocViewerPage?xmlName=innovusug.xml&title=Innovus%20User%20Guide%20--%20Clock%20Tree%20Network%20Structure%20Reporting%20-%20Clock%20Tree%20Network%20Structure%20Reporting&hash=&c_version=23.11&path=innovusUG/innovu. [Accessed 15 June 2024].

[36] Cadence Design Systems, "set_rail_analysis_mode," April 2024. [Online]. Available: https://support.cadence.com/apex/techpubDocViewerPage?xmlName=voltustxtcmdref.xml&title=Voltus%20IC%20Power%20Integrity%20Solution%20Text%20Command%20Reference%20--%20set_rail_analysis_mode%20-%20set_rail_analysis_mode&hash=&c_version=23.11&path=voltustxt. [Accessed 5 August 2024].

[37] Cadence Design Systems, "write_spice_deck," November 2023. [Online]. Available: https://support.cadence.com/apex/techpubDocViewerPage?xmlName=tempuscui.xml&title=Tempus%20Stylus%20Text%20Reference%20Manual%20--%20write_spice_deck%20-%20write_spice_deck&hash=&c_version=23.10&path=tempusCUI/tempusCUI23.10/write_spice_deck.html. [Accessed 5 August 2024].

[38] Cadence Design Systems, "report_delay_calculation," April 2024. [Online]. Available: https://support.cadence.com/apex/techpubDocViewerPage?xmlName=tempustxtcmdref.xml&title=Tempus%20Text%20Command%20Reference%20--%20report_delay_calculation%20-%20report_delay_calculation&hash=&c_version=23.11&path=tempustxtcmdref/tempustxtcmdref23.11/repo. [Accessed 19 August 2024].

[39] DataTechNotes, "Spline Interpolation Example in Python," 11 December 2021. [Online]. Available: https://www.datatechnotes.com/2021/11/spline-interpolation-example-in-python.html. [Accessed 20 06 2024].

[40] Cadence Design Systems, "report_timing," April 2024. [Online]. Available: https://support.cadence.com/apex/techpubDocViewerPage?xmlName=tempustxtcmdref.xml&title=Tempus%20Text%20Command%20Reference%20--%20report_timing_format%20-%20report_timing_format&hash=&c_version=23.11&path=tempustxtcmdref/tempustxtcmdref23.11/report_timin. [Accessed 7 August 2024].

[41] lpGBT Design Team, "lpGBT Documentation release," 2020.

[42] J. Troska, "lpGBT Deterministic Jitter v0 vs v1," 2021.

[43] Cadence Design Systems, "Dynamic Results," November 2023. [Online]. Available: https://support.cadence.com/apex/techpubDocViewerPage?xmlName=voltusmr.xml&title=Voltus%20IC%20Power%20Integrity%20Solution%20Menu%20Reference%20--%20Dynamic%20Results%20-%20Movies&hash=DynamicResults-Movies&c_version=23.10&path=voltusMR/voltusMR23.10/Dyn. [Accessed 19 August 2024].

[44] S. Seguin-Moreau, S. Conforti, C. de La Taille, P. Dinaucourt, G. Martin-Chassard, A. Dragone, B. Markovic, L. Ruckman, A. G. Schwartzman, D. Su, C. Agapopoulou, N. Makovec, S. Sacerdoty, L. Serin, D. Gong, J. Ye and W. Zhou, "ALTIROC, a 25 ps time resolution ASIC for ATLAS HGTD," TWEPP, 2019.

[45] K. Stenson and A. Rao, "Reconstructing a multitude of particle tracks within CMS," [Online]. Available: https://cms.cern/news/reconstructing-multitude-particle-tracks-within-cms. [Accessed 6 August 2024].

[46] N. Seguin-Moreau, "HGTD-ALTRIOC-A FDR ASIC general architecture/overview," 2024.

[47] Vectron, "Clock Jitter Basics," August 2016. [Online]. Available: https://ww1.microchip.com/downloads/aemDocuments/documents/VOP/ApplicationNotes/ApplicationNotes/Clock_Jitter_Basics.pdf. [Accessed 27 June 2024].

[48] M. Moorthly and M. Jayasimha, "Power Delivery Network Design Strategy for Next Generation High Speed IO," in *2018 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS)*, 2018, pp. 1-3.

[49] T. Kugathasan, "Update on Jitter optimization on the lpGBT V1," 2020.

[50] E. Mendes, J. Troska, S. Biereigel, S. Kulis and S. Baron, "Technical evaluation, Preliminary lpGBT clock characterization," 2019.

Appendix A : lpGBT typical

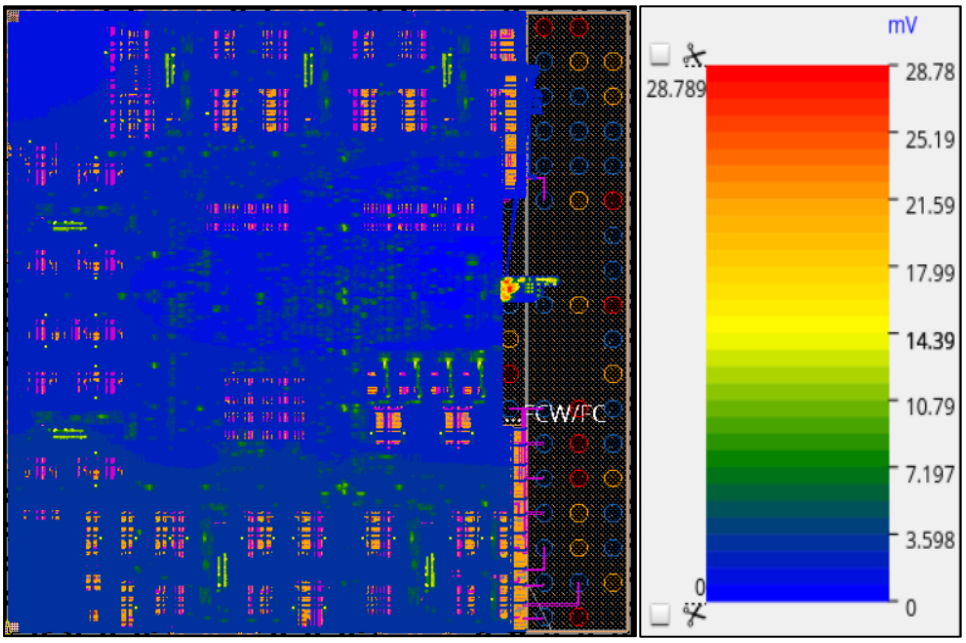

Figure 39 : lpGBT IR drop typical corner.

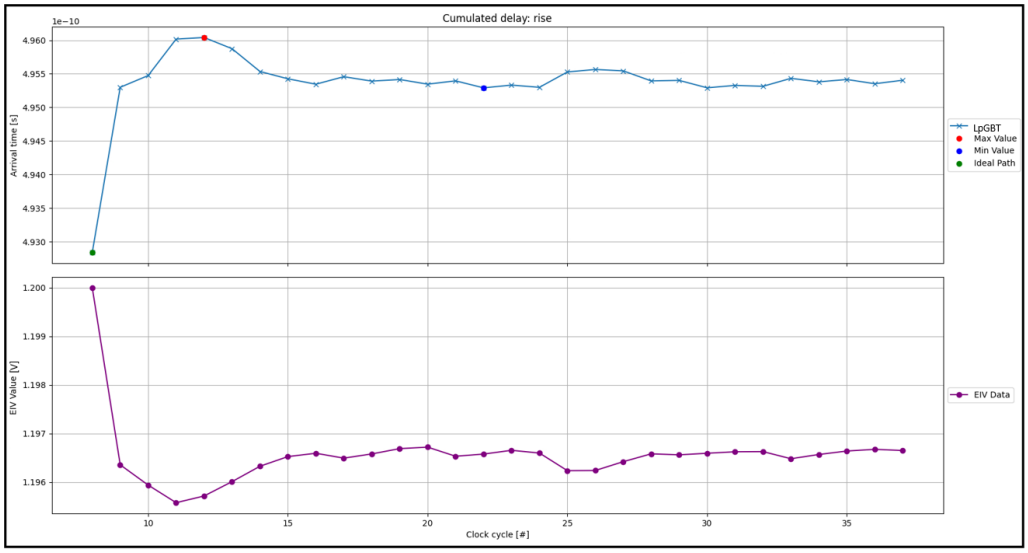

Figure 40 : Arrival time and EIV simulation for lpGBT, typ corner.

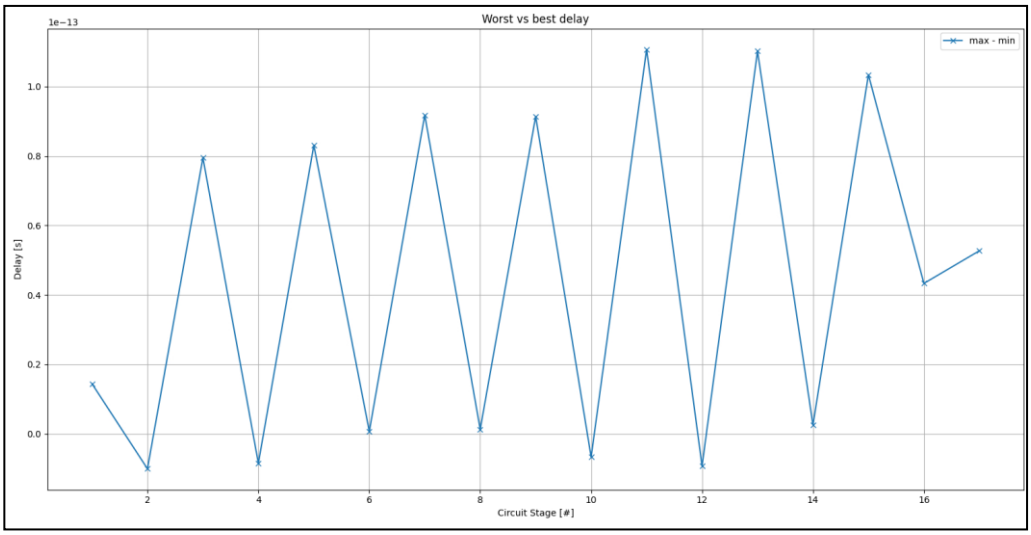

Figure 41 : Delay stage analysis | lpGBT, typ corner.

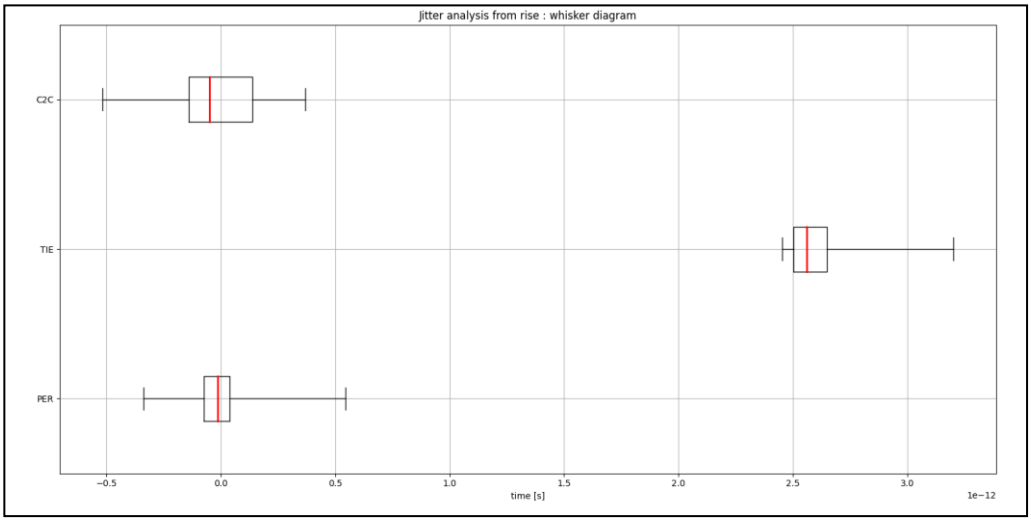

Figure 42 : Jitter whisker diagram | lpGBT, typ corner.

Cycle	**Ideal**	**1**	**2**	**3**
Arrival time [s]	4.928E-10	4.953E-10	4.955E-10	4.960E-10
Average EIV [V]	1.2000	1.1964	1.1959	1.1956
Cycle	**4**	**5**	**6**	**7**
Arrival time [s]	4.960E-10	4.959E-10	4.955E-10	4.954E-10
Average EIV [V]	1.1957	1.1960	1.1963	1.1965
Cycle	**8**	**9**	**10**	**11**
Arrival time [s]	4.953E-10	4.955E-10	4.954E-10	4.954E-10
Average EIV [V]	1.1966	1.1965	1.1966	1.1967
Cycle	**12**	**13**	**14**	**15**
Arrival time [s]	4.953E-10	4.954E-10	4.953E-10	4.953E-10
Average EIV [V]	1.1967	1.1965	1.1966	1.1967
Cycle	**16**	**17**	**18**	**19**
Arrival time [s]	4.953E-10	4.955E-10	4.956E-10	4.955E-10
Average EIV [V]	1.1966	1.1962	1.1962	1.1964
Cycle	**20**	**21**	**22**	**23**
Arrival time [s]	4.954E-10	4.954E-10	4.953E-10	4.953E-10
Average EIV [V]	1.1966	1.1966	1.1966	1.1966
Cycle	**24**	**25**	**26**	**27**
Arrival time [s]	4.953E-10	4.954E-10	4.954E-10	4.954E-10
Average EIV [V]	1.1966	1.1965	1.1966	1.1966
Cycle	**28**	**29**		
Arrival time [s]	4.954E-10	4.954E-10		
Average EIV [V]	1.1967	1.1967		

Table 4 : Detailed data of Figure 40.

Stage	Description		Jitter [s]
11	From	core/ct/clksouth[2].buf/bo_preserve/i	1.11E-13
	To	core/ct/clksouth[2].buf/bo_preserve/z	
13	From	core/ct/clksouth[3].buf/bo_preserve/i	1.10E-13
	To	core/ct/clksouth[3].buf/bo_preserve/z	
15	From	core/ct/clkeclk2eclk3[0].buf/bo_preserve/i	1.03E-13
	To	core/ct/clkeclk2eclk3[0].buf/bo_preserve/z	
7	From	core/ct/clksouth[0].buf/bo_preserve/i	9.18E-14
	To	core/ct/clksouth[0].buf/bo_preserve/z	
9	From	core/ct/clksouth[1].buf/bo_preserve/i	9.14E-14
	To	core/ct/clksouth[1].buf/bo_preserve/z	
5	From	core/ct/clkeast[2].buf/bo_preserve/i	8.31E-14
	To	core/ct/clkeast[2].buf/bo_preserve/z	
3	From	core/ct/clkeast[1].buf/bo_preserve/i	7.95E-14
	To	core/ct/clkeast[1].buf/bo_preserve/z	
17	From	core/ct/clkeclk2eclk3[1].buf/bo_preserve/i	5.27E-14
	To	core/ct/clkeclk2eclk3[1].buf/bo_preserve/z	
16	From	core/ct/clkeclk2eclk3[0].buf/bo_preserve/z	4.33E-14
	To	core/ct/clkeclk2eclk3[1].buf/bo_preserve/i	
1	From	core/ct/clkeast[0].buf/bo_preserve/i	1.44E-14
	To	core/ct/clkeast[0].buf/bo_preserve/z	
14	From	core/ct/clksouth[3].buf/bo_preserve/z	2.50E-15
	To	core/ct/clkeclk2eclk3[0].buf/bo_preserve/i	
8	From	core/ct/clksouth[0].buf/bo_preserve/z	1.27E-15
	To	core/ct/clksouth[1].buf/bo_preserve/i	
6	From	core/ct/clkeast[2].buf/bo_preserve/z	6.70E-16
	To	core/ct/clksouth[0].buf/bo_preserve/i	
10	From	core/ct/clksouth[1].buf/bo_preserve/z	-6.69E-15
	To	core/ct/clksouth[2].buf/bo_preserve/i	
4	From	core/ct/clkeast[1].buf/bo_preserve/z	-8.50E-15
	To	core/ct/clkeast[2].buf/bo_preserve/i	
12	From	core/ct/clksouth[2].buf/bo_preserve/z	-9.30E-15
	To	core/ct/clksouth[3].buf/bo_preserve/i	
2	From	core/ct/clkeast[0].buf/bo_preserve/z	-1.01E-14
	To	core/ct/clkeast[1].buf/bo_preserve/i	

Table 5 : Detailed data of Figure 41 from largest delay to lowest.

Type	Edge	Q1 [s]	Q3 [s]	Min [s]	Max [s]
PER	Rise	-7.47E-14	3.90E-14	-3.40E-13	5.43E-13
TIE	Rise	2.50E-12	2.65E-12	2.45E-12	3.20E-12
C2C	Rise	-1.41E-13	1.38E-13	-5.19E-13	3.68E-13

Range = jitter [s]	Median [s]	StandDev [s]	VarCoeff [%]	Mean [s]
8.83E-13	-1.32E-14	1.54E-11	78.0338	1.97E-15
7.50E-13	2.56E-12	1.96E-11	0.0748763	2.62E-12
8.87E-13	-4.90E-14	1.86E-11	-40.8546	-4.56E-15

Table 6 : Detailed data for Figure 42.

Appendix B : lpGBT low temperature, RC worst

Cycle	**Ideal**	**1**	**2**	**3**
Arrival time [s]	4.2598E-10	4.2766E-10	4.2800E-10	4.2834E-10
Average EIV [V]	1.3200	1.3166	1.3157	1.3162
Cycle	**4**	**5**	**6**	**7**
Arrival time [s]	4.2783E-10	4.2774E-10	4.2763E-10	4.2763E-10
Average EIV [V]	1.3169	1.3171	1.3173	1.3173
Cycle	**8**	**9**	**10**	**11**
Arrival time [s]	4.2756E-10	4.2769E-10	4.2764E-10	4.2766E-10
Average EIV [V]	1.3173	1.3171	1.3171	1.3172
Cycle	**12**	**13**	**14**	**15**
Arrival time [s]	4.2762E-10	4.2772E-10	4.2764E-10	4.2763E-10
Average EIV [V]	1.3172	1.3172	1.3171	1.3172
Cycle	**16**	**17**	**18**	**19**
Arrival time [s]	4.2756E-10	4.2776E-10	4.2780E-10	4.2771E-10
Average EIV [V]	1.3172	1.3167	1.3169	1.3171
Cycle	**20**	**21**	**22**	**23**
Arrival time [s]	4.2762E-10	4.2767E-10	4.2760E-10	4.2760E-10
Average EIV [V]	1.3173	1.3173	1.3173	1.3173
Cycle	**24**	**25**	**26**	**27**
Arrival time [s]	4.2758E-10	4.2769E-10	4.2764E-10	4.2765E-10
Average EIV [V]	1.3173	1.3171	1.3172	1.3172
Cycle	**28**	**29**		
Arrival time [s]	4.2762E-10	4.2771E-10		
Average EIV [V]	1.3173	1.3171		

Table 7 : Detailed data of Figure 26.

Stage	Description		Jitter [s]
13	From	core/ct/clksouth[3].buf/bo_preserve/i	9.32E-14
	To	core/ct/clksouth[3].buf/bo_preserve/z	
5	From	core/ct/clkeast[2].buf/bo_preserve/i	8.92E-14
	To	core/ct/clkeast[2].buf/bo_preserve/z	
7	From	core/ct/clksouth[0].buf/bo_preserve/i	8.87E-14
	To	core/ct/clksouth[0].buf/bo_preserve/z	
9	From	core/ct/clksouth[1].buf/bo_preserve/i	8.76E-14
	To	core/ct/clksouth[1].buf/bo_preserve/z	
3	From	core/ct/clkeast[1].buf/bo_preserve/i	8.65E-14
	To	core/ct/clkeast[1].buf/bo_preserve/z	
15	From	core/ct/clkeclk2eclk3[0].buf/bo_preserve/i	8.48E-14
	To	core/ct/clkeclk2eclk3[0].buf/bo_preserve/z	
11	From	core/ct/clksouth[2].buf/bo_preserve/i	7.88E-14
	To	core/ct/clksouth[2].buf/bo_preserve/z	
16	From	core/ct/clkeclk2eclk3[0].buf/bo_preserve/z	6.28E-14
	To	core/ct/clkeclk2eclk3[1].buf/bo_preserve/i	
17	From	core/ct/clkeclk2eclk3[1].buf/bo_preserve/i	5.13E-14
	To	core/ct/clkeclk2eclk3[1].buf/bo_preserve/z	
1	From	core/ct/clkeast[0].buf/bo_preserve/i	4.78E-14
	To	core/ct/clkeast[0].buf/bo_preserve/z	
8	From	core/ct/clksouth[0].buf/bo_preserve/z	1.36E-14
	To	core/ct/clksouth[1].buf/bo_preserve/i	
2	From	core/ct/clkeast[0].buf/bo_preserve/z	2.80E-15
	To	core/ct/clkeast[1].buf/bo_preserve/i	
14	From	core/ct/clksouth[3].buf/bo_preserve/z	-1.55E-15
	To	core/ct/clkeclk2eclk3[0].buf/bo_preserve/i	
6	From	core/ct/clkeast[2].buf/bo_preserve/z	-2.36E-15
	To	core/ct/clksouth[0].buf/bo_preserve/i	
4	From	core/ct/clkeast[1].buf/bo_preserve/z	-2.40E-15
	To	core/ct/clkeast[2].buf/bo_preserve/i	
12	From	core/ct/clksouth[2].buf/bo_preserve/z	-2.60E-15
	To	core/ct/clksouth[3].buf/bo_preserve/i	
10	From	core/ct/clksouth[1].buf/bo_preserve/z	-4.70E-15
	To	core/ct/clksouth[2].buf/b0_preserve/i	

Table 8 : Detailed data of Figure 27 from highest to lowest delay per stage.

Type	Edge	Q1 [s]	Q3 [s]	Min [s]	Max [s]
PER	Rise	-7.134E-14	4.773E-14	-5.051E-13	3.366E-13
TIE	Rise	1.641E-12	1.731E-12	1.581E-12	2.355E-12
C2C	Rise	-9.559E-14	1.183E-13	-8.410E-13	4.181E-13

Range = jitter [s]	Median [s]	StandDev [s]	VarCoeff [%]	Mean [s]
8.4172E-13	-2.191E-14	1.517E-11	-88.620	-1.712E-15
7.735E-13	1.665E-12	1.522E-11	0.089	1.715E-12
1.259E-12	-1.170E-15	2.134E-11	-23.731	-8.993E-15

Table 9 : Detailed data for Figure 28.

Appendix C : AltirocA typical

Cycle	Ideal	9	10	11
Arrival time [s]	2.6396E-09	2.6801E-09	2.6806E-09	2.6808E-09
Average EIV [V]	1.2000	1.1632	1.1628	1.1621
Cycle	**12**	**13**	**14**	**15**
Arrival time [s]	2.6812E-09	2.6817E-09	2.6818E-09	2.6819E-09
Average EIV [V]	1.1619	1.1617	1.1617	1.1615
Cycle	**16**	**17**	**18**	**19**
Arrival time [s]	2.6818E-09	2.6820E-09	2.6821E-09	2.6822E-09
Average EIV [V]	1.1614	1.1614	1.1614	1.1614
Cycle	**20**	**21**	**22**	**23**
Arrival time [s]	2.6824E-09	2.6823E-09	2.6823E-09	2.6824E-09
Average EIV [V]	1.1612	1.1610	1.1611	1.1611
Cycle	**24**	**25**	**26**	**27**
Arrival time [s]	2.6824E-09	2.6824E-09	2.6822E-09	2.6823E-09
Average EIV [V]	1.1612	1.1612	1.1611	1.1612
Cycle	**28**	**29**	**30**	**31**
Arrival time [s]	2.6823E-09	2.6825E-09	2.6824E-09	2.6822E-09
Average EIV [V]	1.1612	1.1612	1.1612	1.1611
Cycle	**32**	**33**	**34**	**35**
Arrival time [s]	2.6823E-09	2.6824E-09	2.6850E-09	2.6839E-09
Average EIV [V]	1.1612	1.1609	1.1596	1.1601
Cycle	**36**	**37**	**38**	**39**
Arrival time [s]	2.6822E-09	2.6839E-09	2.6841E-09	2.6845E-09
Average EIV [V]	1.1614	1.1603	1.1604	1.1594
Cycle	**40**	**41**	**42**	**43**
Arrival time [s]	2.6871E-09	2.6864E-09	2.6858E-09	2.6853E-09
Average EIV [V]	1.1576	1.1582	1.1587	1.1589
Cycle	**44**	**45**	**46**	**47**
Arrival time [s]	2.6854E-09	2.6859E-09	2.6861E-09	2.6858E-09
Average EIV [V]	1.1587	1.1588	1.1589	1.1590
Cycle	**48**	**49**	**50**	**51**
Arrival time [s]	2.6854E-09	2.6854E-09	2.6860E-09	2.6859E-09
Average EIV [V]	1.1590	1.1588	1.1589	1.1589
Cycle	**52**	**53**	**54**	**55**
Arrival time [s]	2.6858E-09	2.6863E-09	2.6861E-09	2.6858E-09
Average EIV [V]	1.1590	1.1589	1.1588	1.1589
Cycle	**56**	**57**	**58**	**59**
Arrival time [s]	2.6860E-09	2.6863E-09	2.6865E-09	2.6864E-09
Average EIV [V]	1.1589	1.1587	1.1589	1.1588
Cycle	**60**	**61**	**62**	**63**
Arrival time [s]	2.6862E-09	2.6864E-09	2.6859E-09	2.6860E-09
Average EIV [V]	1.1588	1.1589	1.1590	1.1591

Cycle	**64**	**65**	**66**	**67**
Arrival time [s]	2.6860E-09	2.6858E-09	2.6858E-09	2.6855E-09
Average EIV [V]	1.1589	1.1589	1.1590	1.1590
Cycle	**68**	**69**	**70**	**71**
Arrival time [s]	2.6855E-09	2.6853E-09	2.6853E-09	2.6858E-09
Average EIV [V]	1.1591	1.1590	1.1590	1.1589
Cycle	**72**	**73**	**74**	**75**
Arrival time [s]	2.6859E-09	2.6859E-09	2.6856E-09	2.6854E-09
Average EIV [V]	1.1586	1.1586	1.1585	1.1586
Cycle	**76**			
Arrival time [s]	2.6855E-09			
Average EIV [V]	1.1587			

Table 10 : Detailed data of Figure 34.

Stage	Description		Jitter [s]
45	From	cts_ccl_buf_00766/i	5.84E-13
	To	cts_ccl_buf_00766/z	
43	From	cts_ccl_buf_00774/clk	5.21E-13
	To	cts_ccl_buf_00774/c	
47	From	cts_ccl_buf_00758/i	4.66E-13
	To	cts_ccl_buf_00758/z	
41	From	cts_ccl_buf_00784/clk	4.40E-13
	To	cts_ccl_buf_00784/c	
39	From	cts_ccl_buf_00793/i	4.38E-13
	To	cts_ccl_buf_00793/z	
37	From	cts_ccl_buf_00795/clk	3.80E-13
	To	cts_ccl_buf_00795/c	
49	From	cts_ccl_buf_01226/i	3.69E-13
	To	cts_ccl_buf_01226/z	
33	From	cts_ccl_buf_00801/i	3.16E-13
	To	cts_ccl_buf_00801/z	
35	From	cts_ccl_buf_00799/i	3.12E-13
	To	cts_ccl_buf_00799/z	
53	From	cts_ccl_a_buf_00666/i	3.11E-13
	To	cts_ccl_a_buf_00666/z	
51	From	cts_ccl_buf_01221/i	3.00E-13
	To	cts_ccl_buf_01221/z	
62	From	cts_ccl_a_inv_00032/cn	2.95E-13
	To	pixel_analog_part_block/ck40	
1	From	clk640mhz_tdc_int_o_mux_preserve/io	2.72E-13
	To	clk640mhz_tdc_int_o_mux_preserve/z	
27	From	cts_ccl_buf_00702/clk	2.44E-13
	To	cts_ccl_buf_00702/c	
19	From	cts_ccl_buf_00765/clk	2.34E-13
	To	cts_ccl_buf_00765/c	
29	From	cts_ccl_a_buf_00654/clk	2.27E-13
	To	cts_ccl_a_buf_00654/c	
21	From	cts_ccl_buf_00760/clk	2.00E-13
	To	cts_ccl_buf_00760/c	
31	From	cts_ccl_buf_00803/clk	1.78E-13
	To	cts_ccl_buf_00803/c	
25	From	cts_ccl_buf_00749/clk	1.61E-13
	To	cts_ccl_buf_00749/c	
3	From	my_buff_clk_640_to_gen_preserve/clk	1.59E-13
	To	my_buff_clk_640_to_gen_preserve/c	
5	From	clk40tdc_reg_reg/cp	1.51E-13
	To	clk40tdc_reg_reg/q	
23	From	cts_ccl_buf_00755/clk	1.38E-13
	To	cts_ccl_buf_00755/c	
17	From	cts_ccl_buf_00770/clk	1.08E-13
	To	cts_ccl_buf_00770/c	

44	From	cts_ccl_buf_00774/c	8.25E-14
	To	cts_ccl_buf_00766/i	
55	From	cts_ccl_a_inv_00041/clk	7.48E-14
	To	cts_ccl_a_inv_00041/cn	
52	From	cts_ccl_buf_01221/z	7.01E-14
	To	cts_ccl_a_buf_00666/i	
40	From	cts_ccl_buf_00793/z	6.88E-14
	To	cts_ccl_buf_00784/clk	
61	From	cts_ccl_a_inv_00032/clk	6.78E-14
	To	cts_ccl_a_inv_00032/cn	
28	From	cts_ccl_buf_00702/c	5.91E-14
	To	cts_ccl_a_buf_00654/clk	
48	From	cts_ccl_buf_00758/z	5.04E-14
	To	cts_ccl_buf_01226/i	
57	From	cts_ccl_inv_00038/clk	3.84E-14
	To	cts_ccl_inv_00038/cn	
18	From	cts_ccl_buf_00770/c	3.66E-14
	To	cts_ccl_buf_00765/clk	
2	From	clk640mhz_tdc_int_o_mux_preserve/z	3.22E-14
	To	my_buff_clk_640_to_gen_preserve/clk	
26	From	cts_ccl_buf_00749/c	3.15E-14
	To	cts_ccl_buf_00702/clk	
7	From	cts_ccl_inv_00221/clk	2.95E-14
	To	cts_ccl_inv_00221/cn	
60	From	cts_cci_inv_00036/cn	2.65E-14
	To	cts_ccl_a_inv_00032/clk	
9	From	cts_cdb_inv_02621/clk	2.52E-14
	To	cts_cdb_inv_02621/cn	
4	From	my_buff_clk_640_to_gen_preserve/c	2.29E-14
	To	clk40tdc_reg_reg/cp	
11	From	cts_cdb_inv_02622/clk	2.07E-14
	To	cts_cdb_inv_02622/cn	
15	From	cts_ccl_buf_00775/clk	1.23E-14
	To	cts_ccl_buf_00775/c	
22	From	cts_ccl_buf_00760/c	1.12E-14
	To	cts_ccl_buf_00755/clk	
20	From	cts_ccl_buf_00765/c	8.30E-15
	To	cts_ccl_buf_00760/clk	
21	From	cts_cdb_inv_02621/cn	4.54E-15
	To	cts_cdb_inv_02622/clk	
13	From	cts_ccl_a_inv_00219/clk	2.80E-15
	To	cts_ccl_a_inv_00219/cn	
6	From	clk40tdc_reg_reg/q	2.54E-15
	To	cts_ccl_inv_00221/clk	
59	From	cts_cci_inv_00036/clk	2.50E-15
	To	cts_cci_inv_00036/cn	
50	From	cts_ccl_buf_01226/z	1.70E-15
	To	cts_ccl_buf_01221/i	

8	From	cts_ccl_inv_00221/cn	-7.65E-15
	To	cts_cdb_inv_02621/clk	
12	From	cts_cdb_inv_02622/cn	-8.33E-15
	To	cts_ccl_a_inv_00219/clk	
46	From	cts_ccl_buf_00766/z	-8.90E-15
	To	cts_ccl_buf_00758/i	
42	From	cts_ccl_buf_00784/c	-1.59E-14
	To	cts_ccl_buf_00774/clk	
24	From	cts_ccl_buf_00755/c	-1.92E-14
	To	cts_ccl_buf_00749/clk	
58	From	cts_ccl_inv_00038/cn	-2.31E-14
	To	cts_cci_inv_00036/clk	
14	From	cts_ccl_a_inv_00219/cn	-3.62E-14
	To	cts_ccl_buf_00775/clk	
30	From	cts_ccl_a_buf_00654/c	-4.57E-14
	To	cts_ccl_buf_00803/clk	
36	From	cts_ccl_buf_00799/z	-5.26E-14
	To	cts_ccl_buf_00795/clk	
56	From	cts_ccl_a_inv_00041/cn	-5.30E-14
	To	cts_ccl_inv_00038/clk	
32	From	cts_ccl_buf_00803/c	-5.75E-14
	To	cts_ccl_buf_00801/i	
34	From	cts_ccl_buf_00801/z	-5.76E-14
	To	cts_ccl_buf_00799/i	
54	From	cts_ccl_a_buf_00666/z	-6.75E-14
	To	cts_ccl_a_inv_00041/clk	
38	From	cts_ccl_buf_00795/c	-7.27E-14
	To	cts_ccl_buf_00793/i	
16	From	cts_ccl_buf_00775/c	-1.17E-13
	To	cts_ccl_buf_00770/clk	

Table 11 : Detailed data of Figure 35 from highest to lowest delay per stage.

Type	Edge	Q1 [s]	Q3 [s]	Min [s]	Max [s]
PER	Rise	-1.78E-13	2.01E-13	-1.71E-13	2.65E-12
TIE	Rise	4.26E-11	4.62E-11	4.04E-11	4.73E-11
C2C	Rise	-2.28E-13	2.64E-13	-3.37E-12	3.32E-12

Range = jitter [s]	Median [s]	StandDev [s]	VarCoeff [%]	Mean [s]
4.36E-12	2.93E-14	6.10E-11	7.57	8.06E-14
6.95E-12	4.56E-12	1.92E-10	0.04	4.45E-11
7.06E-12	4.58E-15	9.14E-11	-143	-6.37E-15

Table 12 : Detailed data for Figure 36 pixel [0/0].

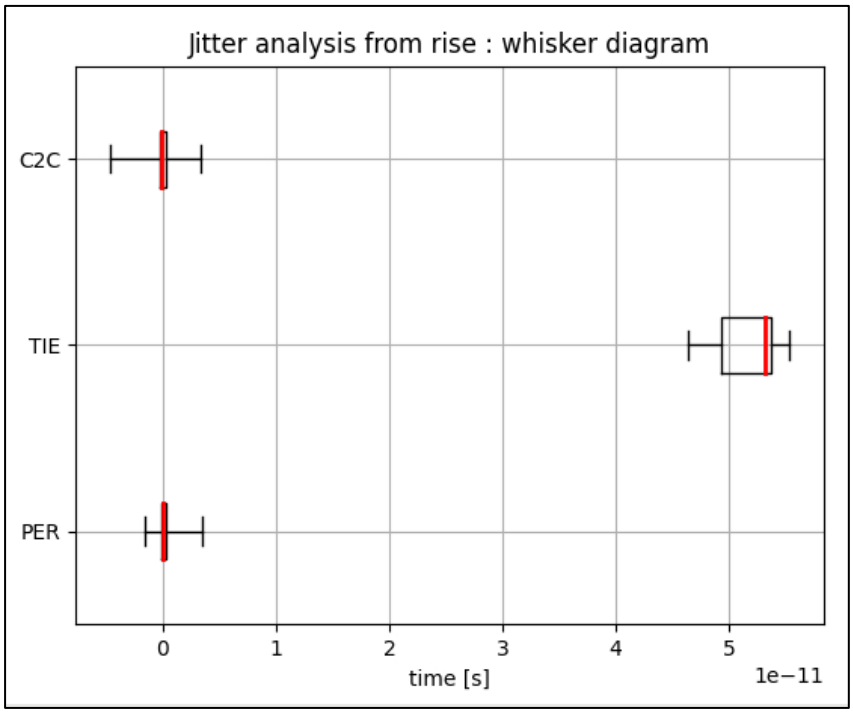

Figure 43 : Whisker diagram for pixel [4/13], worst jitter.

Type	Edge	Q1 [s]	Q3 [s]	Min [s]	Max [s]
PER	Rise	-1.51E-13	3.04E-13	-1.60E-12	3.40E-12
TIE	Rise	4.92E-11	5.36E-11	4.63E-11	5.53E-11
C2C	Rise	-2.81E-13	3.33E-13	-4.68E-12	3.38E-12

Range = jitter [s]	Median [s]	StandDev [s]	VarCoeff [%]	Mean [s]
5.06E-12	7.50E-14	7.27E-11	5.96	1.21E-13
8.97E-12	5.32E-12	2.38E-10	0.04	5.16E-11
8.07E-12	-4.81E-14	1.10E-10	-64.1	-1.72E-14

Table 13 : Detailed data for Figure 36 pixel 4/13, worst jitter.

Appendix D : What is the "Jitter"

Jitter, the timing indeterminism.

The jitter defined by the National Institute of Standards and Technology (**NIST**) is :

> "The short-term variations of the significant instants of a timing signal from their ideal positions in time [...]." [34]

Two companies, NXP Semiconductor and Vectron International, provide a methodical explanation of the terminologies and physical phenomenon associated with jitter, applied to oscillators or a specific board, but general enough to be applied here. A summary of their findings is available below.

- **Random jitter** : Every electronic component, regardless of its scale, is affected by multiple factors such as thermal noise [32], trace width variations in the case of printed circuit board (**PCB**), shot noise and flicker noise [47]. It is exceedingly difficult to deal with random jitters as it often implies changing the frequency of the ASIC or controlling the temperature with high precision.
- **Deterministic jitter** : As the name suggests, deterministic jitter means that it can be calculated and accurately predicted. It arises from all sources that are not stochastic, such as impedance mismatch, crosstalk, IR drop, and ground bounce [32]. In other words, every manufacturing imperfection, design flaw, or uncertainty in the design leads to deterministic jitter. It can be replicated and simulated with ease in comparison to random jitter. The focus on this type of jitter comes from the fact that its effects can be controlled and attenuated in a design process. By controlling the path placement and routing on a design or the threshold voltages of the Complementary Metal-Oxide-Semiconductor (**CMOS**) transistor used for example.
- **Total jitter** : Random jitter and deterministic jitter are concurrent phenomenon that are happening independently but have a cumulative effect on electronic devices.

Assessing jitter is crucial as it symbolizes an uncertainty applied to a clock path in an electronic device. Every designer must know that the clock path of a chip may be responsible for critical failures if affected by significant variability. Therefore, it can lead to functionality loss in the worst-case scenario.

The jitter being defined, and its importance declared, it must be detailed the different jitter classification that exists in ASIC design.

Jitter classification.

There are three ways of calculating total jitter known as period jitter (**PER** in graph), cycle-to-cycle jitter (**C2C** in graph), and time-interval-error jitter (**TIE**). Period jitter

and cycle-to-cycle jitters are commonly used as they are simpler to measure than TIE jitter [32] but may be less relevant from a designer point-of-view.

Time-Interval-Error Jitter

TIE jitter is the difference observed between an ideal reference clock and a simulated or observed clock for each transitional edge.

Moreover, TIE jitter is the main interest of this study since it recreates at best the reality of clock propagation. NXP Semiconductor emphasize on it stating that :

> "TIE is important because it shows the cumulative effect that even a small amount of period jitter can have over time" [32].

Informatively, the TIE jitter will be calculated following Cadence Design Systems formula :

$$TIE[i] = Delay[i]_{non-ideal} - Delay[i]_{ideal} \quad (10)$$

With the delay being computed using Cadence Voltus or Tempus, and [i] the clock cycle under analysis.

It will be seen in the next two sections that period and cycle-to-cycle jitter are used due to the difficulty of measuring TIE jitter, but they also provide diverse types of information that can be complementary.

Period Jitter

The period jitter describes the difference between a real measured period against the mean period of a theoretical or ideal clock [32]. It can be used to verify that the period is constant over time. It appears that even if TIE jitter is the focus of this document, simulating the period jitter is useful as it becomes practical in the case of multiple clock domain on a single chip.

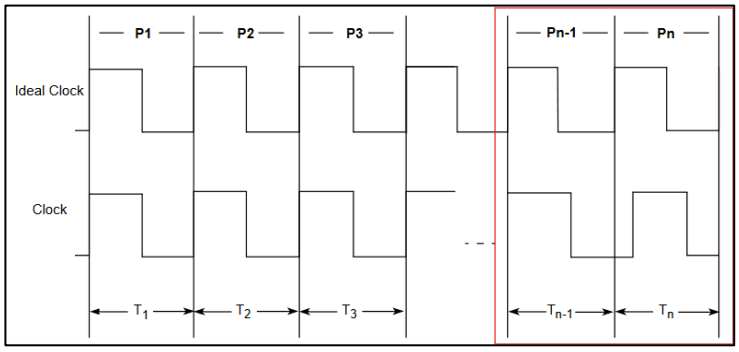

Figure 44 : Period jitter visual representation by NXP [32].

Therefore, Figure 44 above portray a period drift that can happen over time. This implies that thousands of clock cycles are needed to accurately simulate this effect, indeed, clock frequency tends to happen over thousands of clock cycles. In fact, simulation time is a critical aspect of any jitter analysis, hence, generating thousands of cycles may lead to thousands of files and days of simulations.

The formula (11) is used by Cadence in their function “Analyze_jitter” available on Cadence Tempus. It has been decided to keep this formula in the standalone Python verification scripts. The maximum jitter will be extracted from all the period jitter cycle simulated.

$$PER[i] = Delay[i]_{non-ideal} - Delay[i-1]_{non-ideal} \quad (11)$$

Cycle-to-Cycle Jitter

The cycle-to-cycle jitter is the difference between two adjacent clock period [32]. It is often used to determine the quality and consistency of a clock period between two cycles. Although, as it is a punctual approach, it does not witness any large-scale drift or impact on a clock.

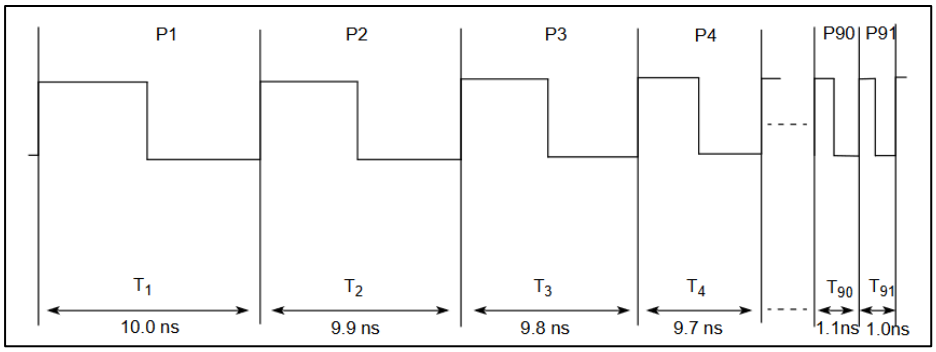

Figure 45 : Cycle-to-cycle jitter, visual representation by NXP [32].

Again, NXP on Figure 45 depict a good representation of C2C jitter. It emphasizes and exaggerates a bit the frequency of drift of a clock, here passing from 100MHz to 1GHz.

$$C2C[i] = Delay[i]_{non-ideal} - 2 \times Delay[i-1]_{non-ideal} + Delay[i-2]_{non-ideal} \quad (12)$$

The formula that is used in this analysis follows the Cadence one presented in (12).