UNIVERSIDADE FEDERAL DE MINAS GERAIS
School of Engineering
Graduate Program in Electrical Engineering

Pedro Sartori Locatelli

TIME-DOMAIN MULTIPLY-ACCUMULATE UNIT

Belo Horizonte

2023

Pedro Sartori Locatelli

TIME-DOMAIN MULTIPLY-ACCUMULATE UNIT

Dissertation presented to the Graduate Program in Electrical Engineering of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

Advisor: Prof. Dr. Dalton Martini Colombo

Belo Horizonte
2023

Locatelli, Pedro Sartori.
L811t Time-Domain Multiply-Accumulate Unit [recurso eletrônico] / Pedro Sartori Locatelli. - 2023.
1 recurso online (88 f. : il., color.) : pdf.

Orientador: Dalton Martini Colombo.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Bibliografia: f. 80-88.
Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Aprendizado do computador - Teses. 3. Processamento de sinais - Teses. 4. Multiplicação - Teses. 5. Cálculos numéricos - Teses. 6. Adição - Teses. 7. Energia - Consumo - Teses. 8. Transistores - Teses. 9. Circuitos - Teses. 10. Circuitos elétricos - Teses. 11. Tempo - Medição - Teses. I. Colombo, Dalton Martini. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.

CDU: 621.3(043)

Ficha catalográfica elaborada pela bibliotecária Ângela Cristina Silva CRB/6 2361
Biblioteca Prof. Mário Werneck, Escola de Engenharia da UFMG

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FOLHA DE APROVAÇÃO

"TIME-DOMAIN MULTIPLY-ACCUMULATE UNIT"

PEDRO SARTORI LOCATELLI

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica. Aprovada em 17 de julho de 2023. Por:

Prof. Dr. Dalton Martini Colombo - Orientador
DEE (UFMG)

Prof. Dr. Sergio Bampi
Instituto de Informática (UFRGS)

Prof. Dr. Robson Luiz Moreno
IESTI (UNIFEI)

Prof. Dr. Ricardo Oliveira Duarte
DELT (UFMG)

Documento assinado eletronicamente por **Dalton Martini Colombo**, **Professor do Magistério Superior**, em 13/07/2023, às 15:09, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Ricardo de Oliveira Duarte**, **Professor do Magistério Superior**, em 17/07/2023, às 18:37, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Sergio Bampi**, **Usuário Externo**, em 18/07/2023, às 11:34, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Robson Luiz Moreno**, **Usuário Externo**, em 21/07/2023, às 11:33, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2467518** e o código CRC **A7894F8C**.

Acknowledgements

I would like to express my heartfelt gratitude to my family for his unwavering support throughout my academic journey. Their encouragement, understanding, and love have been invaluable in helping me achieve this milestone.

I extend my sincere appreciation to my esteemed supervisor, Dr. Dalton Colombo, for his guidance, expertise, and unwavering dedication. His invaluable insights, constructive feedback, and constant encouragement have significantly shaped and improved this master's thesis.

I would also like to extend my gratitude to Dr. Kamal El-Sankay for warmly welcoming me as a visiting researcher at Dalhousie University. His support, mentorship, and collaborative environment provided me with a valuable opportunity to expand my knowledge and broaden my research perspectives.

I am also indebted to the faculty members, research colleagues, and friends who have provided assistance, insightful discussions, and valuable feedback throughout the course of my research.

Moreover, I extend my heartfelt appreciation to the organizations and funding agencies whose support played a crucial role in the development of this work. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, in part by the Fundacao de Amparo a Pesquisa do Estado de Minas Gerais (FAPEMIG) under Grant APQ019872 and in part by the University Support Program for Integrated Circuit Design (APCI) from Brazilian Microelectronics Society (SBMICRO). Furthermore, I would like to express my deep gratitude to Global Affairs Canada for awarding me the Emerging Leaders in the Americas Program (ELAP) scholarship. This scholarship not only provided invaluable financial support but also granted me the opportunity to conduct part of my research in Canada.

Finally, I would like to express my deepest appreciation to all those who have directly or indirectly contributed to the completion of this thesis. Your support and encouragement have been instrumental in my academic growth and achievement.

Thank you all for being a part of my journey and for making this accomplishment possible.

"It's all a question of imagination. Our responsibility begins with the power to imagine."

(Kafka on the Shore, Haruki Murakami)

Resumo

O século 21 marca uma revolução nos campos relacionados a dispositivos eletrônicos e engenharia de computação. A ascensão de conceitos como Machine Learning, Internet das Coisas e 5G está moldando a sociedade e a forma como as pessoas vivem. Neste contexto, o processamento digital de sinais (DSP) serve como denominador comum para todos estes conceitos, sendo crucial para que sejam viáveis e eficazes. Um dos componentes críticos necessários para o processamento de sinais e também para diversas outras aplicações, é a unidade multiplicadora-acumuladora (MAC), que é um circuito responsável pela realização das operações de multiplicação, adição e acumulação. Tipicamente, a unidade MAC está inserida em blocos e aplicações que operam com sinais de tensão e corrente; contudo, à medida que a dimensão das tecnologias CMOS emergentes diminui, melhorar o desempenho ao mesmo tempo que reduz-se a área e o consumo de energia torna-se cada vez mais difícil. A fim de evitar problemas causados pela miniaturização dos transistores e de superar algumas limitações da unidade MAC convencional e de circuitos/aplicações que dependem dela, uma solução viável seria realizar operações de multiplicação e acumulação no domínio tempo. No processamento de sinais em modo tempo (TMSP), o tempo é tratado como a variável que transmite a informação, em vez das variáveis analógicas e digitais convencionais. A vantagem é que esse sinal contém características tanto analógicas, o tempo decorrido/largura do pulso, quanto digitais, pois tal sinal pode assumir apenas dois valores distintos (0 e VDD). Dessa forma, é possível unir as vantagens de circuitos analógicos com as de circuitos digitais. Este trabalho propõe um novo conceito de unidade MAC, baseado no processamento de sinais no domínio tempo. O circuito proposto é capaz de multiplicar consecutivamente dois pulsos temporais de entrada e adicioná-los aos sinais previamente armazenados, oriundos de multiplicações anteriores. O projeto da unidade MAC é realizado em tecnologia CMOS comercial de 180-nm, com apenas 193 portas lógicas, e ocupa área estimada de 3167 μm^2 de área de silício. O circuito proposto é capaz de executar operações de multiplicação-acumulação com erro menor que 5%, para 19 ns de alcance dinâmico e apresentando linearidade em R^2 de mais de 0.99. Seu consumo de energia é de 1.72 mW, considerando 1.8 V como fonte de alimentação.

Palavras-chave: unidade MAC; operação de multiplicação-acumulação; multiplicador no domínio tempo; processamento de sinais em modo tempo; registrador de tempo.

Abstract

The twenty-first century marks a revolution in the fields related to electronic devices and computer engineering. The rise of concepts such as Machine Learning, Internet of Things and 5G is shaping the society and the way that people live. In this context, digital signal processing (DSP) serves as a common denominator for all of these concepts, being crucial to their viability and effectiveness. One of the critical components required for signal processing and for many other applications, is the multiply-accumulate (MAC) unit, which is a circuit responsible for performing the operations of multiplication, addition and accumulation. Typically, the MAC unit is incorporated into blocks and applications that operate with voltage and current signals; however, as the feature size of emerging CMOS technologies shrinks, improving performance while reducing area and power consumption becomes increasingly difficult. In order to avoid problems caused by transistor scaling and to overcome some limitations of the conventional MAC unit and applications that rely on this circuit, a viable solution would be to perform multiply-accumulate operations in time-domain. In time mode signal processing (TMSP), time is treated as the variable that transmits the information, instead of the conventional analog and digital variables. The advantage is that this signal contains both analog characteristics, the elapsed time/pulse width, and digital characteristics, since such a signal can take on only two distinct values (0 and VDD). In this way it is possible to combine the advantages of analog circuits with those of digital circuits. This work proposes a new concept of MAC unit, based on time-domain signal processing. The proposed circuit is capable of consecutively multiplying two input time pulses and add them to previously stored signals. The MAC unit design is realized in commercial 180-nm CMOS process, with just 193 logic gates, and occupies an estimated silicon area of about 3167 µm^2, employing . The proposed circuit can perform multiply-accumulate operations with less than 5% error for a dynamic range of 19 ns, presenting an R^2 linearity of over 0.99. Its power consumption is 1.72 mW from a 1.8 V supply.

Keywords: MAC unit; multiply-accumulate operation; time-domain multiplier; time-mode signal processing; time-register.

List of Figures

List of Tables

List of abbreviations and acronyms

ASIC	Application Specific Integrated Circuits
CLA	Carry Look-ahead Adder
CLK	Clock
CMOS	Complementary metal-oxide-semiconductor
CSKA	Carry-Skip Adder
CSLA	Carry Select Adder
DR	Dynamic Range
DSP	Digital Signal Processing
EN	Enable
FA	Full-Adder
FPGA	Field Programmable Gate Array
GRO	Gated Ring Oscillator
HA	Half-Adder
IC	Integrated Circuit
LSB	Least significant bit
MAC	Multiply–accumulate
MSB	Most significant bit
NMOS	N-channel metal-oxide semiconductor
PMOS	P-channel metal-oxide semiconductor
RCA	Ripple Carry Adder
SNR	Signal-to-Noise Ratio

TD	Time-Domain
TMSP	Time Mode Signal Processing
TREG	Time-Register

Contents

Chapter 1

Introduction

For a long time, one of the focal points for advancements in the CMOS technology was related to the reduction of transistor dimensions. Mainly due to transistor miniaturization, that is, the manufacture of these components in ever-smaller dimensions, designing circuits with higher transistor density that took up less space on the chip was not a very complex task. Not only that, but even with the physical dimensions of the devices being reduced, it was possible to manufacture circuits with greater performance and better efficiency. All of these characteristics were very well predicted and mapped by the Moore and Dennard scaling laws [1], with which the industry has been able to keep pace for decades. However, recent technologies have taken longer to develop as the node shrink, due to the increased complexity of the process and the higher number of masks required during microfabrication [2, 3].

This trend towards miniaturization, while beneficial in many ways, is a challenge especially when it comes to integrate analog and mixed-signal circuits with digital ones, since CMOS processes are typically optimized for the needs of digital circuitry. As the feature size of emerging CMOS technologies shrinks, the thickness of the transistor gate oxide reduces, forcing the system voltage to decrease. This negatively affects analog and mixed-signal circuit performance by reducing the input and output voltage swings and increasing the sensitivity to noise. As a result, a decrease in dynamic range (DR) and signal-to-noise ratio (SNR) is to be expected. Another issue that has arisen as a result of scaling is the degradation of device matching characteristics due to limitations in lithography and resolution enhancement techniques used during fabrication, which can compromise the speed and resolution of the circuits [4, 5].

In order to avoid the problem of reduced supply voltage, an alternative is to design circuits with current signals rather than voltage ones. In this approach, referred to as current mode, the advantage is that there is no predetermined limit on the magnitude of current signals, even in miniaturized devices [6]. However, circuits in the current domain also do not scale well with technology and typically consume more power than their voltage mode counterparts [7]. Thereby, while current mode can be advantageous in some ways, the transmitted information is still represented by the magnitude of the signal, just as it is in voltage mode. In an effort to address the design challenges posed by deep submicron CMOS processes and mitigate the performance degradation caused by scaling, this work will explore the utilization of a promising alternative known as time-domain operation [8–11]. By employing time-domain techniques, it is anticipated that the aforementioned challenges can be overcome, providing a viable alternative to conventional voltage/current operation.

The concept behind Time-Mode Signal Processing (TMSP) is to treat time as the variable under processing rather than conventional analog or digital variables. Basically, in TMSP the amplitude of analog signal is represented in proportion to pulse width of time signal or as the time (phase) difference between the occurrences of two digital events. Not only that, but since the signal has only two largely distinct values it can also be interpreted as a digital one. Therefore, because of the duality of the time variable used, it is possible to perform analog signal processing in a digital environment, which is very beneficial since analog accuracy and digital advantage can be combined together [11]. It is worth noting that digital signal processing can also be done in this environment. To better illustrate the possibilities of TMSP, a block diagram containing the range of its applications is shown in Figure 1.

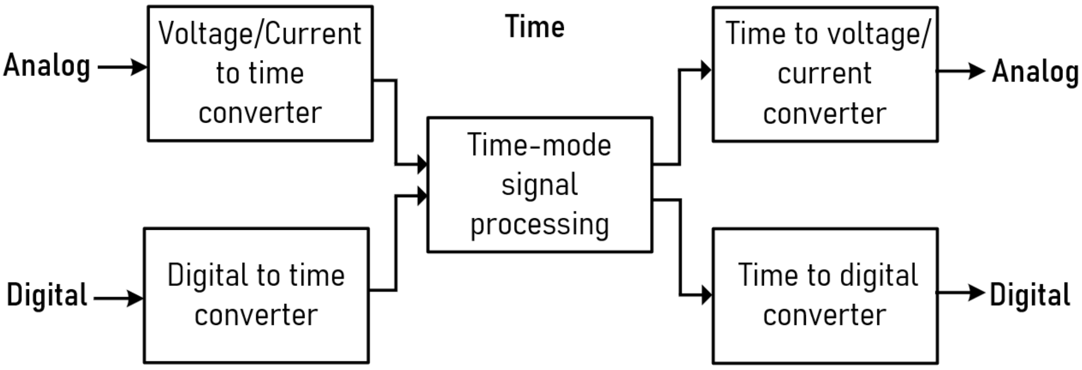

Analog
Voltage/Current to time converter
Time
Time to voltage/ current converter
Analog
Time-mode signal processing
Digital
Digital to time converter
Time to digital converter
Digital

Figure 1 – Time-Mode Signal Processing concept for Analog and Digital processing (based on [11] but modified)

In view of the fact that time-mode circuits are essentially digital circuits, the detrimental effects of technology scaling on analog signal processing are less significant when adopting TMSP. By using the time-domain, analog and mixed-signal circuits tend to be more scalable in terms of performance and power consumption as the gate delay of digital circuits reduces with technology. Also, due to high switching speed of the latest MOS transistors, time resolution associated with digital circuits has already surpassed the voltage resolution of analog ones and the tendency is for this disparity to grow even more with scaling [12]. In addition to that, the digital nature of time-mode circuits allows them to be migrated from one generation of technology to another with minimum design time, increasing the portability and, subsequently, lowering the cost [7]. The time-domain not only benefits analog circuits, but it also brings new perspectives for digital ones too. One of them is the reduction of buses and circuits in arithmetic operations, because the time variable can transmit a number using only one track, whereas the digital variable requires one track for each bit [13]. Furthermore, in digital circuits some metrics, such as area and power consumption, tend to increase exponentially with the number of bits, which is not desirable. These parameters are critical and can be limiting, because, in addition to being directly related to increased costs, the chip area is limited.

Another recent challenge is the integration of analog and digital circuits on the same integrated circuit due to performance disparities in newer CMOS technologies and the way they handle ever lower voltages. Many times, either analog and digital chips are separated, or higher power supplies for the analog portion of an integrated system-on-chip are used. However, both solutions would consume more power and occupy larger silicon area than an integrated solution that only use the core power supply. Thus, by employing the time-domain this issue could be alleviated because it allows analog and digital circuits to be designed on the same chip and with the same power supply without compromising their performance.

The application for TMSP is not that recent since time-mode signals have been used in a variety of scenarios for decades, including time of flight measurement, digital storage oscilloscopes, and medical imaging instruments, to name a few [14]. Nowadays the use of this approach is being extended to a wider range of applications and circuits that fully operate in the time-domain are becoming increasingly common. As an example one can mention phase locked loops (PLLs), delay locked loops (DLLs), temperature sensors [15], clock data recovery circuits [16], frequency synthesizers [17] and finite-impulse-response (FIR) filters [5].

1.1 Motivation and Objective

The twenty-first century marks a revolution in the fields related to electronic devices and computer engineering. The rise of concepts such as Machine Learning, Internet of Things and 5G is shaping the society and the way that people live. Signal processing, which is the process of analyzing and modifying a signal to optimize its performance, is of utmost importance in all of the aforementioned applications and, for that reason, is currently being a topic of high interest among researchers. One of the critical components required for signal processing is the Multiply-accumulate (MAC) unit that has gained attention in academic and corporate fields due to the wide range of applications in which it can be used.

Typically, the MAC unit is incorporated into blocks and applications that operate with voltage and current signals; however, as the feature size of emerging CMOS technologies shrinks, improving performance while reducing area and power consumption becomes increasingly difficult [18]. In order to avoid problems caused by transistor scaling and to overcome some limitations of the conventional MAC unit and applications that rely on this circuit, a viable solution would be to perform multiply-accumulate operations in a time-domain environment.

Because the MAC unit is a fundamental component in signal processing, designing this block in the time-domain may be interesting, not only for the benefits that the circuit itself may provide, but also for allowing better integration with other time mode circuits. Despite the fact that the time-domain is already being employed in many applications, arithmetic operations are still often performed analogically or digitally, which require the need for the use of converters, increasing energy consumption and silicon area.

The goal of this master thesis is to explore the time mode operation by developing a complete MAC unit capable of operating with time signals. In order to do so, first a fully time-domain multiplier, which is the most complex block of that MAC unit, is proposed and designed. Overall, this thesis discusses the architecture of the proposed time-domain MAC unit, as well as all of the subcircuits that are used to implement it. All circuits are designed in 180 nm process technology.

This master's thesis serves as an initial endeavor in disseminating highly significant time-domain circuits, such as a fully time-domain multiplier and MAC unit. As far as the author is aware, these circuits have not been documented in the existing literature. The development of time-domain circuits is an emerging research field and several subsequent projects are enabled by this master thesis, since the circuits developed here can be used for a wide range of applications.

The main results achieved in this work are also available in [19].

1.2 Thesis Overview

This master thesis is organized into five chapters as follows:

Chapter 2 elucidates key aspects of the conventional MAC unit that are required to comprehend the development of this work. Traditional architectures of adders and multipliers, which are the most fundamental circuits needed to implement the MAC unit, are also addressed. Chapter 3 focuses on the presentation of the developed circuits, from the most basic ones to the entire MAC unit, emphasizing their principles of operation and all of their nuances. Furthermore, basic concepts of the time-domain and how signals are manipulated in this domain are addressed, as they are critical for understanding the time-domain MAC unit operation. Chapter 4 analyzes the results obtained after simulating the proposed time-domain MAC unit and its sub-blocks. Also, conventional architectures of digital MAC units were implemented to serve as a basis of comparison for the proposed time-domain one. Chapter 5 presents the conclusions of this master thesis and delineates the future works.

Chapter 2

Conventional Multiply-accumulate unit: Fundamental Blocks and operation

This chapter provides an overview of conventional MAC units, as well as its most basic blocks, the adders and multipliers, that are relevant for understanding the time-domain solution. Several classic architectures of these circuits, suitable for VLSI implementation, will be covered along this chapter. Because the time-domain inherits characteristics from both analog and digital circuits they will both be addressed. However, the emphasis will be on the digital ones, since they share more similarities with the circuits in the time-domain. Furthermore, the circuits discussed here will play a vital role in the design of digital MAC units, allowing for further performance comparison against the time-domain solution.

2.1 Multiply-accumulate unit

In the vast majority of real-time Digital Signal Processing (DSP) applications the Multiply–accumulate (MAC) unit is an indispensable component when it involves multiplication and/or accumulation. The DSP applications include filtering, image processing, video coding, convolution operation and speech processing [20]. However, the MAC unit usage is not limited to DSP since this block is also used extensively in many other applications such as microprocessors, micro-controller and other data processing units [21, 22].

The conventional architecture of the MAC unit consists of three main circuits, namely the multiplier, the adder and the accumulator. Fundamentally, the multiplier produces partial products whose results are sent to the adder. It will then add the results of the multiplier to the previously accumulated results. Multipliers are typically a combination of a partial-product generation unit and a carry propagate adder to add the partial products to find the final sum. The simplified block diagram of this circuit is shown in Figure 2.

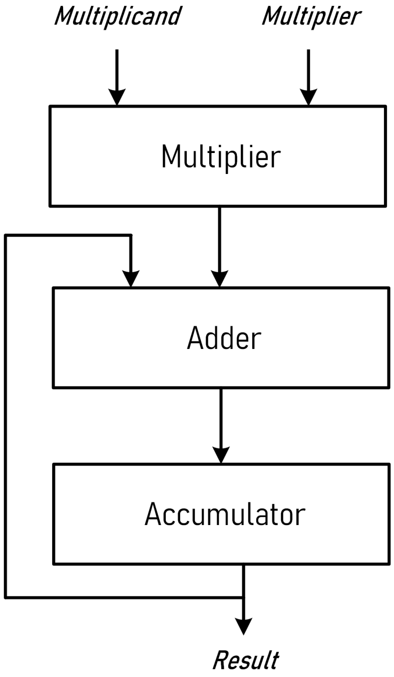


Figure 2 – MAC unit block diagram

One of the current challenges concerning the MAC unit is regarding the design of this circuit as it lies in the critical path of the system which determines the overall operational speed and power of the hardware. In order to fulfill the requirements of a given application the MAC unit must be designed taking into account some aspects which are directly associated to its operation and performance. These aspects can be divided into two major groups: design methodology and performance metrics, which are strongly correlated.

Firstly, in terms of design methodology some more general features must be heeded when conceiving the MAC unit. To name a few:

- Architecture Schemes: The architecture selection typically involves the organization of functional blocks in MAC and the number of pipes involved in the computation stage. The choice of the MAC unit architecture generally depends on the type of application since it is directly associated with power, speed and area. Typically, three MAC unit architectures schemes are most widely used these being recursive, parallel and shared segmented MAC [23]. The recursive architecture scheme employs the "divide-and-conquer" strategy to obtain large size data elements by deploying smaller modules that use iterative data calculation through several clock cycles. Due to its innate serial nature, the recursive architecture tends to be slower than the others and utilizes more number of computation stages and clock cycles with minimum resource utilization. The parallel architecture, on the other hand, is implemented with parallel and pipelined structure with larger elements to achieve high speed at the expense of area and power consumption. In addition to the incorporation of pipelines, several intermediate registers are also implemented which improves the latency and throughput of the system. The last type MAC architecture is the shared segmented vector method which lies between the recursive and parallel schemes. In the shared segmented solution there are fewer recursive components when compared to the parallel approach in order to reduce the power and area constraint but the latency and throughput is high when compared to recursive MAC [23]. The MAC architecture can be implemented in Field Programmable Gate Array (FPGA) and Application Specific Integration Circuit (ASIC) [24]. FPGAs have limited resources and fixed logic technology for MAC structure implementation, whereas ASICs are semi-custom or fully custom, allowing for optimization from the architectural level to the transistor level.

- Multipliers: As stated before, the multiplier is a fundamental component required to implement the MAC unit. Different schemes for designing efficient digital multiplier structures for MAC units have been proposed from digital to analog ones [25]. More details will be given further in Sections 2.3 and 2.4 which are dedicated to this block.

- Adders: Just like the multiplier, the adder will be detailed further, in Sections 2.2 and 2.4. As expected, this block can also be implemented by employing distinct schemes depending on the application, with the possibility of being designed either analogically or digitally [26].

- Design style: Nowadays the vast majority of MAC units are implemented using the conventional static CMOS structure, mostly due to this technology wide spread, low power consumption, low cost and high noise immunity . However, other logic styles, static or dynamic, can also be used when designing the MAC unit, such as complementary Pass Transistor logic (CPL) [27], Swing restored pass-transistor logic (SRPL) [28], and Domino logic [29], for instance. It is also possible to hybrid design the MAC unit by employing different styles in any of the circuit blocks [30].

- Technology node: The term "technology node" refers to a specific semiconductor manufacturing process and the design rules that govern it. Different nodes frequently imply various circuit generations and architectures. In general, the smaller the technology node, the smaller the feature size, resulting in smaller transistors that are both faster and more power-efficient. However, the fabrication prices exponentially increases as the process node decreases [31]. Therefore, the node choice when design the MAC unit is very crucial since each node has its own features and specificities.

As indicated previously the MAC unit operation is directly affected by the design choices. For that reason it is crucial that these choices are made in order to allow the desired performance metrics to be achieved. A few relevant and most common performance metrics will be addressed with regard to the MAC unit:

- Power: Power consumption is one of the most important parameters regarding VLSI design since by minimizing this parameter one can make electronics more efficient, increase battery life and standby time on portable devices and lower the chance of failure from electrical or thermal instability, to name a few. In the MAC unit the major components of power are: static , short circuit and dynamic power. The static power, or leakage, is due to the sub-threshold currents and by reverse biased diodes in a CMOS transistor. The short circuit power is due to the direct current path between VDD and ground which occurs during the switching of both NMOS and PMOS transistors in the circuit. These two powers are due to the technology and logic style through which the MAC is designed. Finally, dynamic power consumption occurs when signals which go through the CMOS circuits change their logic state charging and discharging of output node capacitor which is directly associated to the node activity factor of the MAC unit [32].

- Clock Frequency: This parameter is directly proportional to the power factor of the MAC unit and to the speed and throughput of the circuit. This means that a MAC unit with a high clock frequency will consume more power but will be faster than one with low clock frequency. Often the choice of this parameter is due to the speed and power requirements of the application in which the MAC unit will be implemented.

- Throughput: The number of MAC operations at a given time interval is undoubtedly one of the most important parameters when it comes to the performance of this circuit. As mentioned, the throughput of the MAC unit is directly associated with its clock frequency and can be numerically expressed in Eq. (2.1):

$$MAC\ operations/sec = (\frac{Clock\ Frequency}{Cycle\ of\ each\ MAC\ operation}) \times MAC\ unit\ quantity \tag{2.1}$$

- Figure of Merit: There are many ways to summarize the performance of a given circuit and figures of merit can be used in order to quickly and objectively determine their fitness for a particular application. Usually, the performance of a MAC unit, can be expressed in Eq. (2.2):

$$FOM = \frac{f}{PV} \times 100 \tag{2.2}$$

where, P is the total power of the MAC unit for a given voltage V operating at a given frequency f [33].

However, many other figures of merit can be used in order to characterize the performance of the MAC unit, such as the power-delay product or energy-delay-area product for example [34,35].

- Area: In terms of area occupation a few trade-offs must be taken into account when designing the MAC unit. By using a larger area one can increase the speed and throughput of the circuit, for instance. However, larger circuits are more expensive to fabricate and tend to be less power efficient.

2.2 Digital Adders

Along with multiplication, the addition is the most often used arithmetic operation in microprocessors, digital signal processors and application specific integrated circuits. This operation is required to run specific algorithms such as convolution, correlation, and digital filtering and, often, dictates the speed of the circuit [36,37].

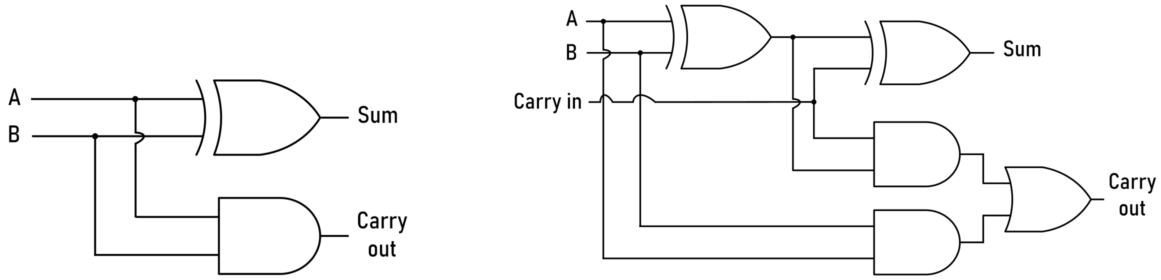
The core element of the adders architectures are full and half adders, that will be covered in Section 2.2.1. The role of those circuits in computer arithmetic can be divided into two main categories. One category involves the chain structured applications such as ripple carry adders (RCA) and array multipliers (Section 2.3.2). In these applications, where is demanded that the generation of the carry-out signal is fast, the critical path frequently runs from the carry-in to the carry-out of the full adders. Otherwise, the slower carry-out generation will not only prolong the worst-case delay, but will also introduce more glitches in later stages, causing more power to be dissipated. The other category involves the tree structured applications, which is frequently used in Wallace tree multipliers (Section 2.3.3) and multiplierless digital filters. Full adders and half adders in these applications form a tree of several layers to compress the partial products to a carry-saved number before a final carry propagation adder converts it to a normal binary number [30].

In the matter of the MAC unit, as discussed in Section 2.1, adders are essential for the proper operation of the circuit, primarily, because this block is responsible for adding up the results of the multiplier to the previously accumulated results. They are also of the utmost importance for the multipliers since the vast majority of those circuits operates by means of successive additions, which can be done in a variety of ways depending on the architecture used. Having said that, the purpose of this section is to cover some common adder architectures, encompassing different techniques to do the addition, that are suitable for implementation at transistor level.

2.2.1 Half Adder and Full Adder

Half adders and full adders are the most basic circuits for adding two numbers, serving as the foundation for various multipliers and more complex adder architectures. Overall, these circuits can perform all of the mathematical operations required for digital computing.

In the half adder, as shown in Figure 3a, only two logic gates, XOR and AND, are needed to do the sum. The output of this circuit is the result of the addition and the carry out, which is generated when both inputs logic levels are high. The full adder, on the other hand, is a bit more complex and it can be implemented by using three types of logic gates, two XOR, two AND and one OR, as shown in Figure 3b. This circuit is able to perform the addition of three inputs, that are usually the two numbers that will be added and the carry in, which is usually the carry out of another full adder or half adder. The outputs of the full adder are the sum and the carry out, just like in the half adder.



(a) Half Adder

(b) Full Adder

Figure 3 – Half and Full Adders

2.2.2 Ripple Carry Adder

The Ripple Carry Adder (RCA) is one of the simplest adder designs, consisting of full adders cascaded together. Essentially, this circuit takes in two N-bit inputs and produces an N-bit sum and a 1-bit carry-out [38]. To do so, in this architecture the carry-out of each full adder is the carry in of the succeeding next most significant full adder, as shown in Figure 4.

From a VLSI design perspective, the RCA is easy to implement, since the full adder is the only circuit that needs to be designed. To create an N-bit RCA one just needs to arrange N full adders, connecting the carry-out of each circuit into the carry-in of the next. Despite the fact that it is a simple circuit with little design time, the RCA performs poorly in terms of speed, which is its most significant disadvantage. In general, this architecture tends to be slow, since it suffers from higher delays. This is because to obtain the partial sum in each stage, the carry-in of the previous stage is needed. As a

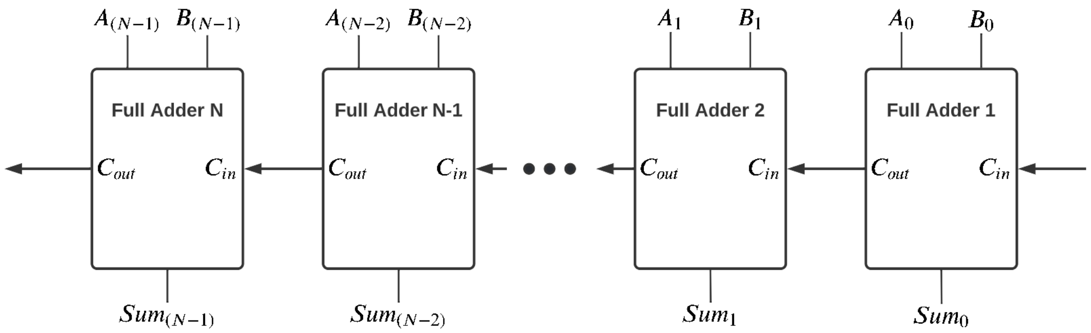


Figure 4 – N-bit Ripple Carry Adder

result, in order to obtain the final result of an addition, the delay of each single stage must be computed. The more bits the RCA has, the longer it takes to perform the operation [39].

2.2.3 Carry-Skip Adder

Just like the RCAs, the Carry-Skip Adder (CSKA) consists of a very simple and regular layout, requiring low layout area and power. However, they differ especially in terms of speed, because the CSKA addresses the RCA's major disadvantage, which is the carry-out dependency of the previous full adder. In general, this problem is partially solved in CSKA by inspecting groups of bits to determine whether or not they have a carry-out [39]. To better understand how this is done in Figure 5 is shown a common implementation of the carry-skip adder, which consists of full adders, a 2-to-1 multiplexer and a logic gate AND. As one can notice, each full adder creates a carry-out signal (Cout), a propagate signal (P) and a partial sum (Sum).

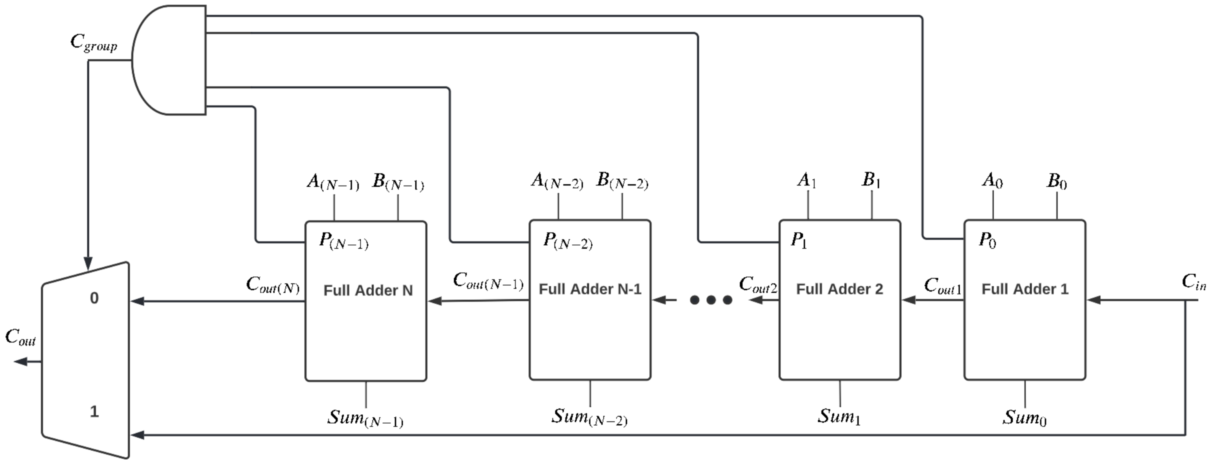


Figure 5 – N-bit Carry-Skip Adder

To further comprehend how those signals are generated and their utility, in Figure 6 is shown the design of a full adder, just like in Figure 3b, but with the intermediate signals "propagate" and "generate" highlighted, which are essential to the operation of the CSKA and other adder structures that will be covered. Briefly, these signals can be used in many adder structures to simplify the addition and predict the overall carry-out with less effort. By using them, for instance, it is possible to determine when the carry-out of each full adder will be generated regardless of the carry-in (Generate=1), when it will be zero (Generate = Propagate = 0) or when the carry-in will be propagated (Propagate=1, Generate=0). Therefore, using these signals allows for the design of faster and more complex adders, as will be shown along this section.

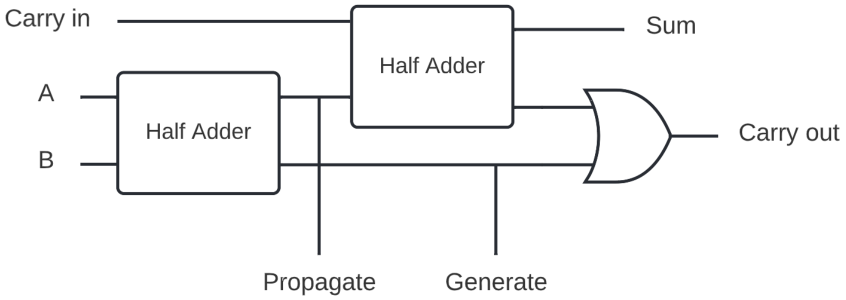


Figure 6 – Full adder with intermediate signals

After detailing the origin and utility of these intermediate signals, it is easier to understand the operation of the CSKA. Basically, when all propagate signals are high the carry-out signal equals the carry-in, which means that the input carry signal is skipping all the full adders. However, when at least one of the propagate signals is zero, this means that the carry-out of the circuit does not depend on the carry-in and, therefore, it must be calculated through logic, just as it is done in RCAs [40].

The main disadvantage of this topology is that carry-skip adder performance is improved with only some of the input bit combinations, in contrast to other faster adders that will be discussed next. This implies that speed increases are not always guaranteed, being only probabilistic. However, this topology is very power efficient and occupy almost the same silicon area as the RCA's [39].

2.2.4 Carry look-ahead adder

The Carry look-ahead adder (CLA) is another architecture based on the RCAs, focused on improving the addition speed by reducing the amount of time required to determine carry bits. To accomplish this, the CLA computes all the carry bits at a time, reducing the time required to calculate the result of the adder's larger-value bits. However, in order to present a higher speed, it is expected that the architecture must be more robust and complex [41]. In general, the CLA consists of full adders, to add and generate the intermediate signals "propagate" and "generate", and a logic circuit, which is responsible for determining the carry. Figure 7 shows a common implementation of the carry look-ahead adder.

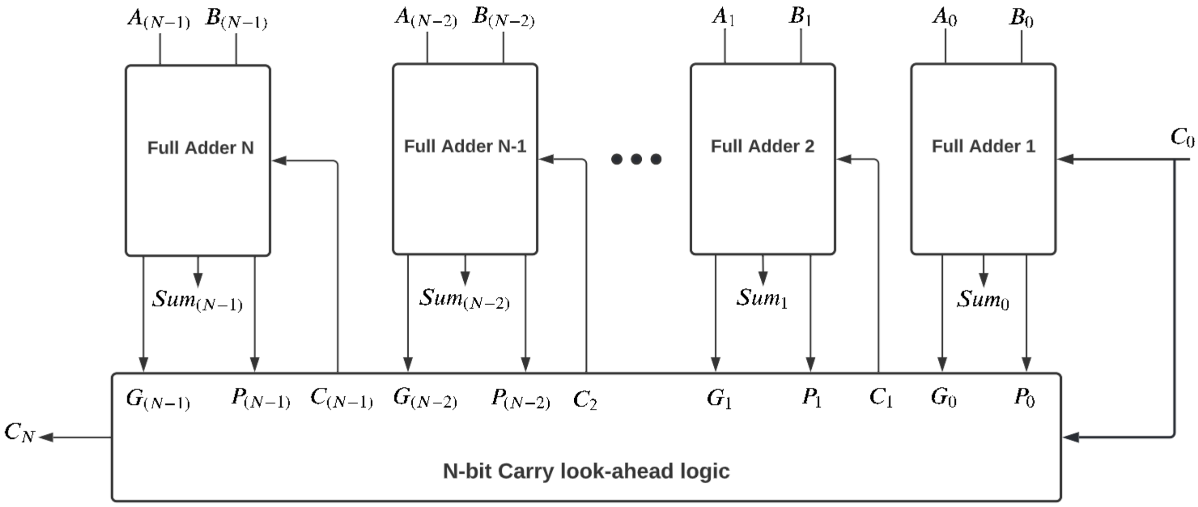


Figure 7 – N-bit Carry-lookahead adder

The working principle of the CLA can be divided into three main steps. First, each full adder generates their respective "propagate" and "generate" signals simultaneously, after being fed with their inputs. Following that, the signals are routed through a logic circuit that determines the carry for each bit. Finally, because each adder has now their two input bits and a carry-in, the complete sum can be performed [42].

As mentioned, the major advantage of the CLA is in terms of speed, due to its fast addition logic and reduced propagation delay. However, as the number of bits increase, the more complex the look-ahead carry logic becomes and the speed gains can be compromised. To get around this problem, it is common to organize CLAs with many bits into smaller groups, so that each one has an optimal size. Also, due to the additional circuitry, the CLA tends to be costlier, occupy more chip area and consume more power [43].

2.2.5 Carry-select adder

The last topology that will be discussed is the carry-select adder (CSLA) which uses redundancy in order to increase the addition speed [44]. The main idea behind this architecture is to perform two additions, for any sum of bits. In the first addition the carry-in is assumed to be zero whereas in the second it is considered one. Later, after the carry-in is known, just one result of these sums is selected [45]. In Figure 8 a widespread design of the carry-select adder is shown.

The CSLA is basically composed of two adder blocks, one for each addition, and a multiplexer, which is responsible for selecting the correct sum, as well as the correct carry-out once the carry-in is known. In Figure 8 the most simple implementation is shown, with each adder block (i.e. group of full adders in each row) being a ripple-carry adder. Still, it should be noted that the CSLA can be designed with any of the adder structures discussed in the preceding sections.

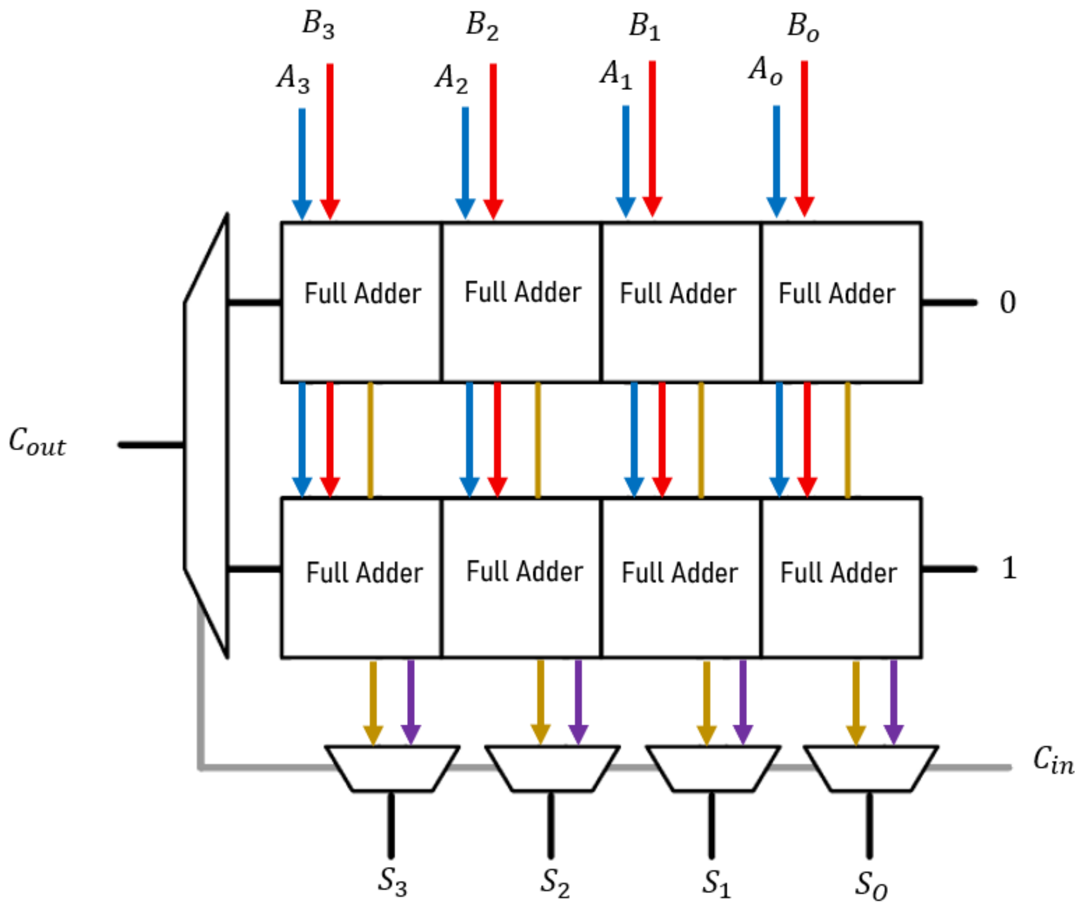


Figure 8 – 4-bit Carry-Select adder

The main advantage of this architecture is in terms of speed, since the delay for an addition is replaced by the delay of a multiplexer, making it faster than the others previously shown. Also, the CSLA has regularity that makes it easier to elaborate the layout. However, when compared to other adder structures, it consumes more power, has a higher amount of logic gates and occupies larger chip area [39].

2.3 Digital Multipliers

Multiplication is one of the basic and most essential operations when it comes to signal processing. In the multiplication process, each bit of the multiplier is multiplied by each bit of the multiplicand, yielding partial products. These unfinished components are then combined to form the final result. Hence, the process of multiplication can be broken down into three steps: generating partial product, reducing partial product and computing final product.

Digital multipliers can be classified into two major types, these being hardware multipliers and software multipliers [46,47]. Software multipliers, traditionally implemented using microprograms, were widely prevalent in older digital systems, mainly due to the challenges associated with implementing hardware multipliers several decades ago. During that time, hardware-based multiplication units were significantly more complex and resource-intensive to design and integrate into digital systems. As a result, software-based approaches, utilizing microprograms, were commonly employed as a feasible alternative for multiplication operations. The major advantage of software multipliers is that they can handle various data types and sizes, allowing for flexibility and versatility. However, due to their sequential nature of execution, they are often slower compared to hardware multiplication [48, 49], which usually have parallel processing capabilities [50].

With the breakthrough of integrated circuits and the demand for high-speed digital systems, however, hardware multipliers started to become more and more viable. They are now essential building blocks in a vast array of digital and high-performance systems, that require fast and efficient multiplication operations to handle the computational demands involved. Furthermore, hardware multipliers have found extensive use in a diverse range of applications, including digital signal processing, cryptography, graphics processing, among others. [51–55].

In the matter of the MAC unit the multiplier is the most critical block since it usually occupies the majority of the chip area and often dictates the speed of the circuit. A wide variety of approaches for implementing the MAC unit multiplication function are possible. In general, the choice is based upon factors such as throughput, latency, area, and design complexity [56, 57].

Some standard digital multiplier designs, which are suitable for VLSI implementation at CMOS level, will be covered along this section. Furthermore, at the end of the chapter, analog multipliers will be briefly discussed, as they are important for understanding the time-domain multiplier, which is a hybrid between analog and digital domains.

2.3.1 Sequential Multiplier

The sequential multiplier is one of the oldest approaches for multiplying two binary numbers and, just as its name suggests, in this type of circuit the process is divided into a few sequential steps. Each step will generate some partial products, which will be added to an accumulated partial sum, and the partial sum will be shifted to align the accumulated sum with the partial products of the following steps. As a result, each step of a sequential multiplication consists of three distinct operations: producing partial products, adding the produced partial products to the accumulated partial sum, and shifting the partial sum [58]. An overview of the sequential multiplier is shown in Figure 9.

A sequential multiplier is typically made up of a register that holds the multiplicand, a shift register that initially holds the multiplier, a shift accumulator that holds the partial

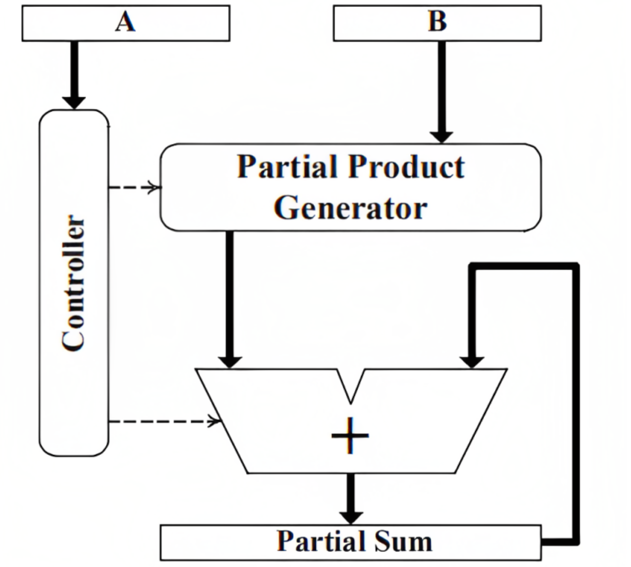


Figure 9 – Sequential Multiplier architecture concept [58]

product, and a shift counter. This type of circuit is very simple but it is also relevant in many architectures, as it is the base of many newly developed multiplication techniques. Sequential multipliers tend to occupy less chip area, but are slower when compared to the other solutions. To increase the speed of multiplication, parallel adder arrays can be used to add partial products, for example.

2.3.2 Array Multiplier

The array multiplier is also a simple architecture, based on repeated addition and shifting procedure. In short, it is a type of a combinational multiplier that can be implemented by arranging Half Adders (HA) and Full Adders (FA) to add the partial products, which are generated by conventional AND logic gates.

Multiplication, of two binary numbers, in array multipliers can be accomplished with a single operation by employing a combinational circuit that generates the product bit at the same time, making it a fast method of multiplying two numbers. As one can observe from Figure 10, which shows a conventional implementation of the array multiplier, the only delay is the time it takes for the signals to propagate through the logic gates.

When compared to other multiplier architectures that are based on the same principle, which will be covered further, it is safe to say that the array multiplier is not a complex circuit, but it tends to be slower and not very efficient, since is its very power consuming due to the high amount of components needed to design it. Also, for this same reason, it occupies a large area, which makes them quite expensive to fabricate [57].

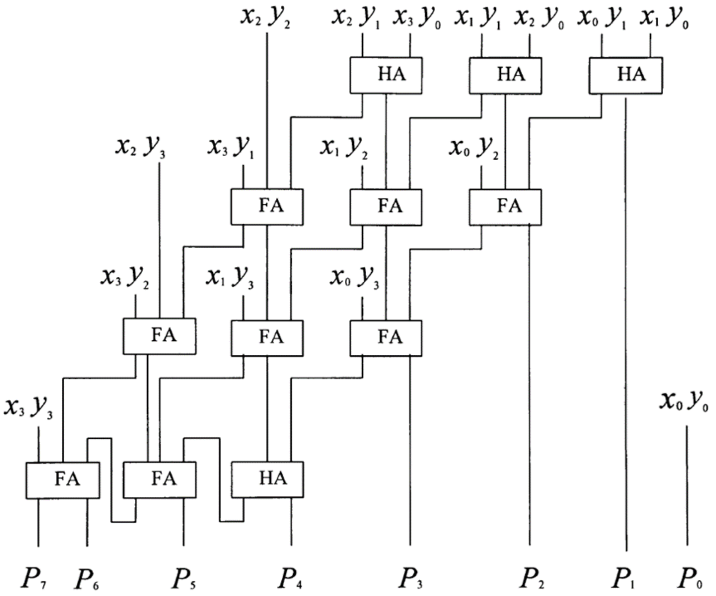


Figure 10 – 4x4 Array Multiplier

2.3.3 Wallace Tree and Dadda Multipliers

The Wallace Tree is a parallel multiplier approach that allows the multiplication of two binary numbers at high speed by using the column compression technique. In this architecture half adders and full adders are employed to sum partial products in stages until only two numbers remain. Briefly, the bit products are formed, the bit product matrix is reduced to a two row matrix where the sum of the row equals the sum of bit products, and the two resulting rows are summed with a fast carry-propagate adder to produce the result [59].

Although the number of the partial products are just the same as in the array multiplier, here they are grouped into sets of three, which makes the multiplier faster, more power efficient and less area consuming. However, because in this architecture the wiring is irregular, the design complexity tends to be higher. The conventional Wallace Tree design is shown in Figure 11.

As one can perceive there are several ways to improve the performance of multipliers. One way is by means of optimization in partial reductions, which can make the multiplier faster and less area/component demanding. The Dadda multiplier, which is basically a refinement of the Wallace tree, do this very well by reducing the number of half and full adders needed to do the multiplication, for instance [61]. Another common way is through modifications in the partial product generation, for example, by using Booth's algorithm, as will be shown in Section 2.3.4.

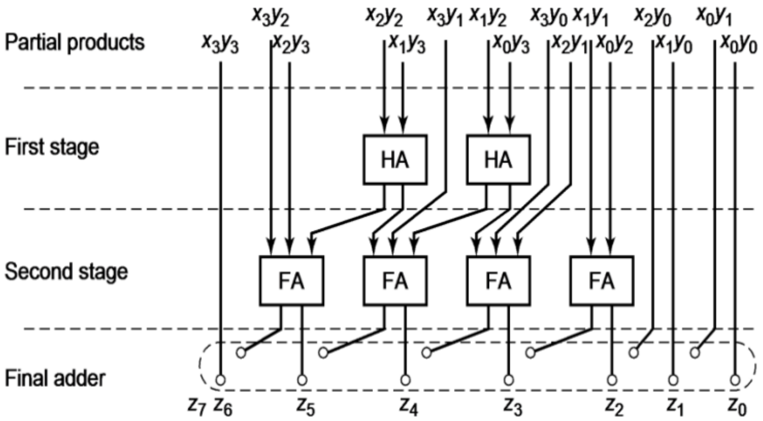


Figure 11 – 4x4 Wallace Tree Multiplier [60]

2.3.4 Booth Multiplier

In the multipliers architectures discussed earlier the partial product generation was done by means of an AND gate, which basically multiply two bits. However, depending on the word length, the multiplier may have a high delay due to the high amount of partial products. With that stated, the Booth Multiplier is a popular implementation to reduce delay by reducing the number of partial products. Also, this multiplier is capable of multiplying two signed binary numbers in two's complement notation, which is another great feature. However, due to the increased complexity, the booth multiplier tends to occupy a larger area and consume more power when compared to the architectures previously shown [62].

The steps of multiplication are quite similar to the other parallel architectures. First, the partial products are generated with the help of the Booth encoder, next they are submitted to a reduction tree and finally the reduced partial products are summed in the final adder. This process is more easily visualized with the help of Figure 12 , which shows the block diagram of a conventional Booth multiplier.

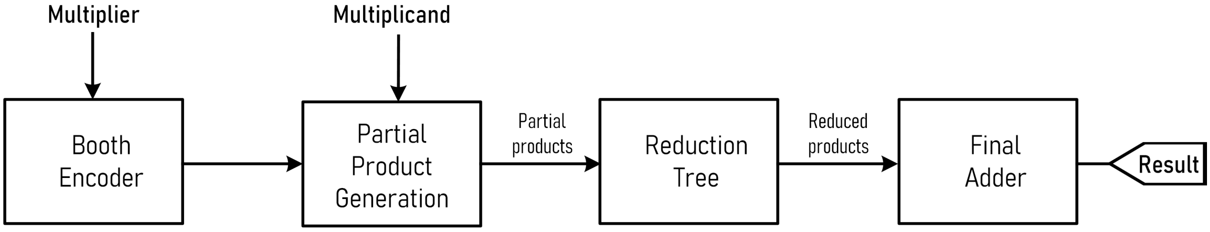
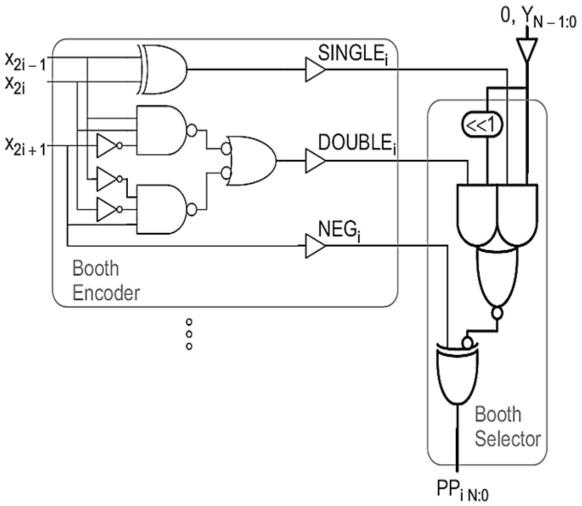


Figure 12 – Conventional Booth Multiplier Diagram

The Booth encoder is what differentiates the Booth multiplier from the other designs previously addressed. This block, as stated, is responsible for the optimized partial product generation. Different schemes can be used to implement this circuit, which are still being developed in order to reduce the encoder power and area consumption, for example [62, 63]. A simple topology of a 3-bit Booth encoder [64] and partial product generator is shown in Figure 13. In this circuit each group of three bits (a pair plus the most significant bit of the previous pair) is encoded and driven across the partial product row using several select lines (SINGLEi, DOUBLEi, and NEGi). The multiplier Y is distributed evenly across all rows. Booth selectors are controlled by the select lines, which choose the appropriate multiple of Y for each partial product.



Inputs			Partial Product	Booth Selects		
x_{2i+1}	x_{2i}	x_{2i-1}	PP_i	$SINGLE_i$	$DOUBLE_i$	NEG_i
0	0	0	0	0	0	0
0	0	1	Y	1	0	0
0	1	0	Y	1	0	0
0	1	1	$2Y$	0	1	0
1	0	0	$-2Y$	0	1	1
1	0	1	$-Y$	1	0	1
1	1	0	$-Y$	1	0	1
1	1	1	$-0\ (=0)$	0	0	1

Figure 13 – 3-bit Booth Encoder
[65]

In addition, it should be noted that it is also possible to combine different multiplier designs to increase performance. One common approach, for instance, is by employing the booth encoder for partial product generation and the Wallace tree method for partial product reduction [66].

2.4 Analog Adders and Multipliers

Analog adders and multipliers are another effective way to implement addition and multiplication operations. Despite the fact that their use is becoming increasingly limited, they can be even preferred over digital ones for some applications. Just to name a few, these analog circuits are often employed in variable gain amplifiers [67] adaptive filters, frequency doublers, modulators/demodulators [68, 69], neural networks [70] and fuzzy logic controllers.

Several techniques for implementing the analog operations using either voltage [71] or current signals [72] have been presented. In both cases, the square law of MOS transistors operating in the saturation region serves as the primary starting point for the design of strong inversion multiplier circuits. Addressing the multipliers more specifically, another alternative would be to operate in weak inversion region and rely on the exponential relationship between the gate voltage and drain current in MOS transistors. As a result, the multiplier's power consumption and power supply can be reduced, but the frequency response is compromised [73]. Different approaches can also be employed when designing an analog multiplier, such as translinear principle, exponential cells, square-difference circuits, bulk-driven, gate driven and dynamic threshold MOS, for instance [73].

Regardless of the design method and the transistor's operating mode or whether the signals are voltage or current, analog arithmetic circuits can be categorized into three main groups according to the polarity of the inputs and output [72, 74]. The most simple configuration is the one-quadrant in which all signals can have only one polarity, which is usually positive. In the two-quadrants configuration one of the inputs can assume both positive and negative values, meaning the output is also bipolar. Finally, in the four-quadrants configuration both the inputs and the output of the circuit can assume positive or negative values. This configuration is the most common in the literature and is frequently preferred due to its broader range of potential applications [72, 74].

In contrast to digital multipliers, a particularity of the analog ones is that the resulting signal from a given operation does not usually depend solely on the inputs to be multiplied. Although the output is proportional to the input signals, in analog multipliers it often depends on the transistor construction parameters and the circuit topology, represented by the variable "k", as shown in Eq. (2.3).

$$V_{out} = k \times V_{in1} \times V_{in2} \tag{2.3}$$

For analog adders the behaviour is very similar, as the output is also not only input dependant, as shown in Eq. (2.4).

$$V_{out} = k \times (V_{in1} + V_{in2}) \tag{2.4}$$

In short, analog adders and multipliers are used in a limited number of applications, most of the times, being designed for a specific situation. The great advantage of using analog circuits for performing a given operation on their own is the simplicity of the design, as analog circuits often use fewer devices than corresponding digital ones. For example, a four-quadrant adder can be fabricated from four transistors and a four-quadrant multiplier from nine to seventeen, depending on the required range of operation [75].

Overall, digital signal processing techniques can perform the functions of an analog adder/multiplier better and at a lower cost, with greater reproducibility. A digital solution is less expensive, more effective at low frequencies, and even allows the circuit function to be modified in firmware. As frequency increases, however, the cost of implementing digital solutions rises much faster than the cost of implementing analog ones. Hence, as digital technology advances, the use of analog circuits becomes increasingly relegated to higher-frequency circuits or, as emphasized before, to highly specialized applications.

Chapter 3

Time-domain multiplication and accumulation

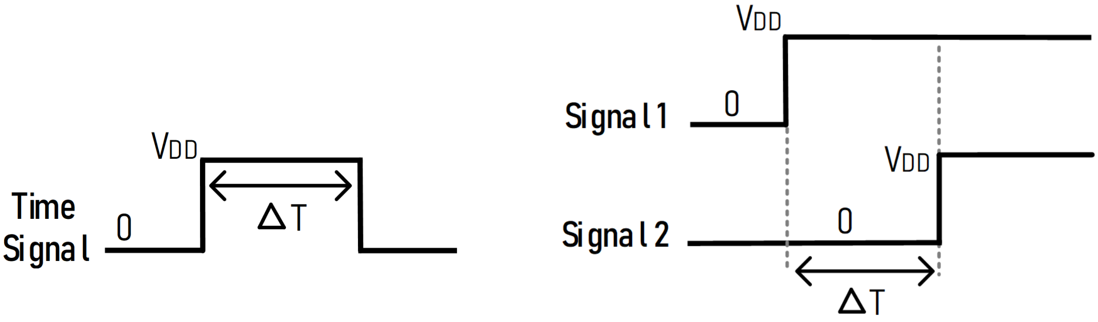
The primary purpose of this chapter is to show the proposed time-domain MAC unit and explain its operation. To accomplish this, the time-domain peculiarities, as well as the primary blocks required for circuit operation, will be addressed first. The most complex component of the MAC unit is the time-domain multiplier, which will be discussed in detail along this chapter. By properly understanding this circuit, it is easier to understand the entire MAC unit operation and its nuances.

3.1 Time-domain

Time-domain can be interpreted as a hybrid between the analog and digital domains. Basically, in the time-domain the amplitude of the analog signal is represented in proportion to the pulse width of time signal or as the time (phase) difference between the occurrences of two digital events [76], as shown in Figure 14. Not only that, but since the time variable has only two largely distinct values it can also be interpreted as a digital one. Thus, the elapsed time can be interpreted as an analog magnitude and the voltage level as a digital one.

Due to the dual nature of the time signals it is feasible to perform analog signal processing in a digital environment, which is particularly beneficial because analog precision and digital advantages can be merged. As it is known, analog circuits do not do so well with scalability, mostly due to supply voltage drops, which affects the dynamic range, speed and resolution of the circuits. However, digital ones tend to benefit from it, presenting more resolution and less delay as transistor size shrinks.

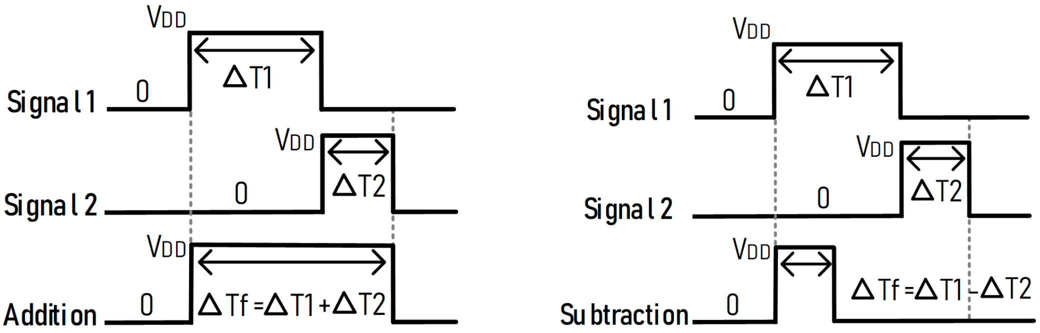
The main element used in the time-domain are the delay cells, which are essentially inverters, or NOT gates. These delay cells could be either technology dependent or can be controlled with the help of a control voltage/current signal. By using them it is possible,



(a) Time variable as a pulse width (b) Time variable as a difference between time pulses

Figure 14 – Time-domain signal representation

for example, to store, add, subtract, multiply and divide time signals. Some operations are shown in Figure 15.



(a) Addition of time signals (b) Subtraction of time signals

Figure 15 – Time-domain operations

Delay cells are primarily responsible for enabling the use of time-domain, as scaling also improves temporal resolution due to the digital nature of these circuits. Consequently, it is conceivable to have analog-like behavior that is also improving with scaling.

3.2 Proposed time-domain multiplier architecture

As discussed in the previous chapter, adders and multipliers can be either analog or digital, each with its own set of advantages and drawbacks. In this work, aiming to reduce the limitations of these circuits due to its signals nature, while emphasizing their unique advantages, a time-domain multiplier was designed. The core principle is to have a circuit that operates in a digital environment but is able to treat information analogically.

The proposed time-domain multiplier is shown in Figure 16 and is the key to the MAC unit topology that will be addressed further. As one can see, it has three main blocks, the time-registers, the gated ring oscillator (GRO) and the bidirectional counter, that will be explained with more detail throughout this chapter.

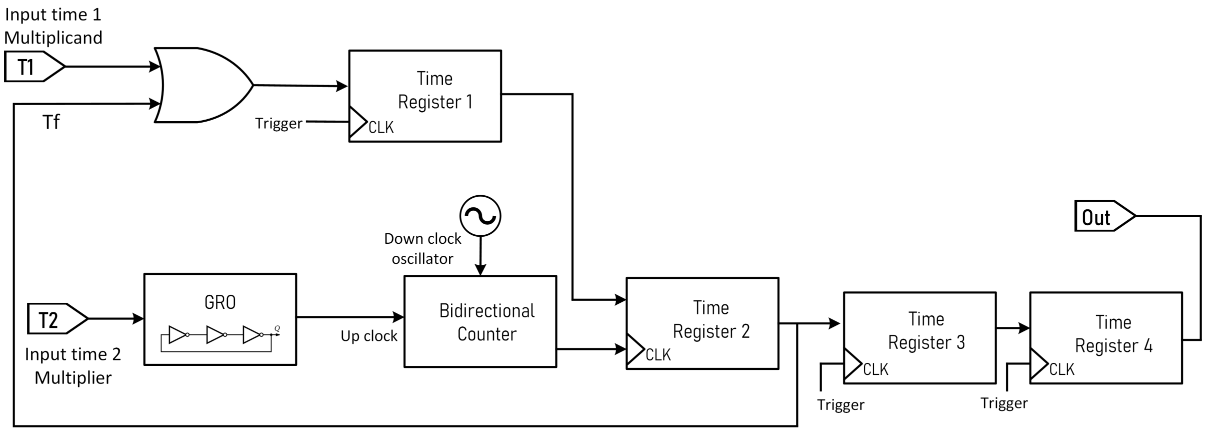


Figure 16 – Proposed Time-Domain Multiplier architecture

The main idea behind this topology is to multiply two time signals, which are the inputs T1 (multiplicand) and T2 (multiplier). Instead of simply multiplying the two inputs, which is not possible in the time-domain, the multiplication is performed by means of successive additions, so basically the first input time T1 is added to itself "T2" times. However, because T2 is a time variable, it needs to be discretized, which is done with the aid of the GRO and the counter. The addition of time signals is done in the time-register, which not only can add two numbers but also store them.

Since it is simpler to understand the entire time-domain multiplier after being familiar with the operation of each block separately, Section 3.3 will go into greater detail regarding the circuit's principle of operation and all of its intricacies. It is worth mentioning that the idea behind the topology conceived was inspired by [14], however the approach presented in this thesis is not only novel but also distinct from existing ones found in the literature.

3.2.1 Time-register

The most important circuit for implementing the time-domain multiplier and MAC unit is the time-register. This circuit's primary function is to store time information and retrieve it when needed [77], but it can also perform additions and subtractions depending on how the inputs are arranged.

To better understand how the circuit works a good analogy is to consider the time-register as a water tank [78], as shown in Figure 17. To fill the water tank the *Enable* signal must be high, which occurs when either the input signal In or the $Trigger$ signals are high. The storage of information happens in the write mode of operation when T_{in} is high and according to the analogy previously made the tank is filled by the amount of pulse-width. When neither In nor $Trigger$ is high, in the hold mode of operation, no water enters or exits the water tank, indicating that the time-register will only retain the information stored. Finally, in the read mode, the stored input T_{in} can be retrieved.

To do so, the *Trigger* signal is set to high and the water tank begins to fill up again until it reaches its full capacity (full scale T_{FS}). When that happens, a *Full* signal flag is set to high, indicating that the operation is over. Hence, the desired information, T_{in}, appears in the output, which is defined as the time difference between the *Trigger* and the *Full* signals. However, at this point, the output signal is still not exactly the input signal, although it contains it.

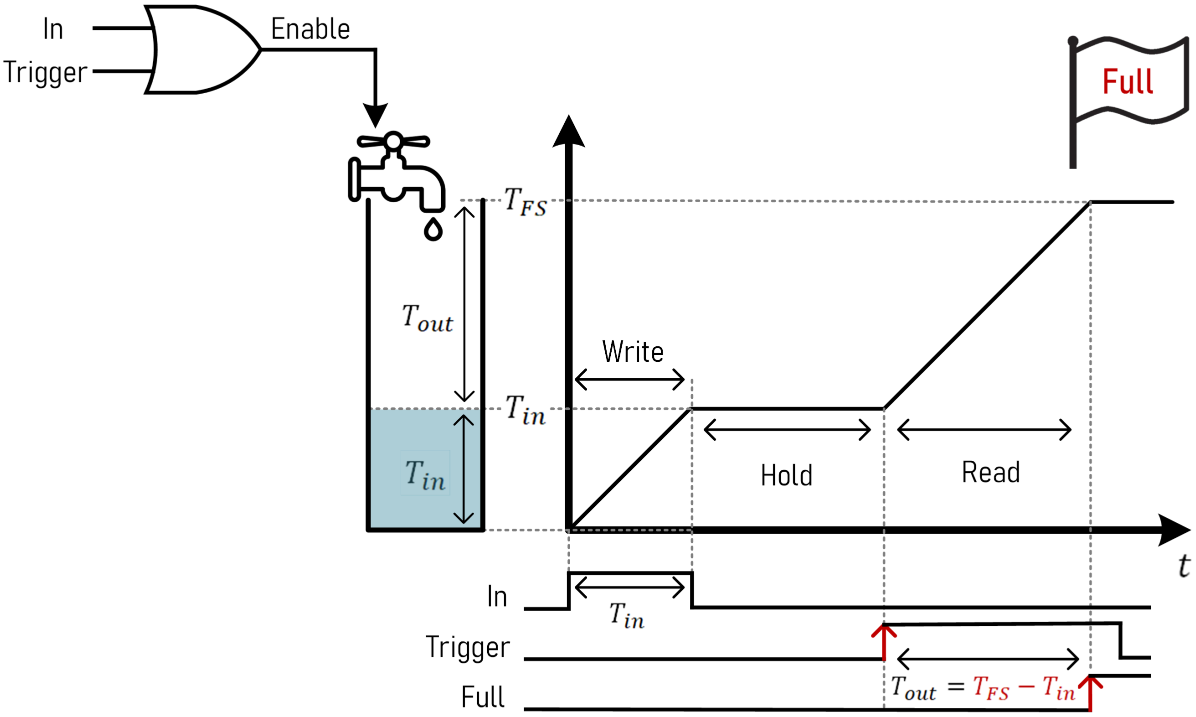


Figure 17 – Time-Register concept diagram

Mathematically, one can note that the output contains the information of the stored input as [79]:

$$T_{out} = T_{FS} - T_{in} \tag{3.1}$$

However, by using only one time-register, the output will be complimentary, since it will also depend on the constructive characteristics of the circuit. To retrieve the true input time, two cascaded time-registers can be used, as shown in Figure 18. This way, the final output is defined as follows:

$$T_{out} = T_{FS} - (T_{FS} - T_{in}) = T_{in} \tag{3.2}$$

In summary, the time-register is a circuit that will receive time pulses as input, storing them until a trigger signal is applied. The output is released after the trigger and is equal to the full scale of the time register minus the input time

The possibilities when using the time-register are numerous. Due to time-domain properties, this circuit can perform additions and subtractions, as shown in Figures 19 and 20, while also storing signals, as long as the inputs and triggers are well synchronized [78].

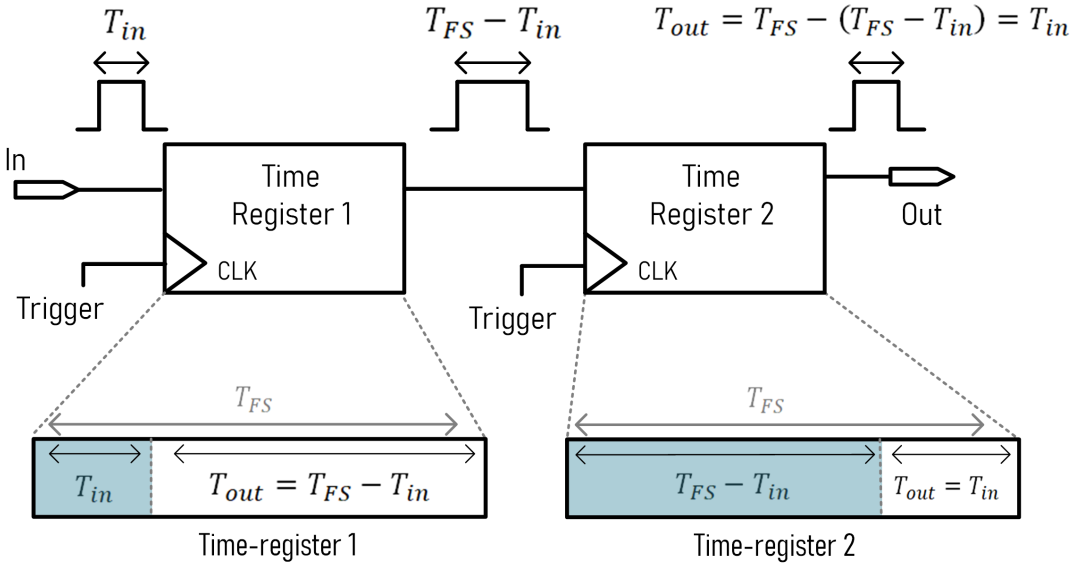


Figure 18 – Cascaded Time-Register for obtaining the true input

Addition, which is an operation used extensively in the time-domain multiplier, can be accomplished by arranging two time-registers as shown in Figure 19. In this time-adder configuration two inputs (T_a and T_b) can be successively sent to the first time-register, which will store first T_a and then T_b. After receiving a trigger signal, the output of the first time-register is released and it is given by $T_{out} = T_{FS} - (T_a + T_b)$. This signal then goes into the input of the second time-register and is released as $T_{out} = T_a + T_b$ thereupon the trigger signal. This means that all inputs are combined and released as a single signal, with pulsewidth equal to the sum of all the stored inputs. Therefore, multiple inputs can be stored using the time-adder configuration as long as they are sequentially fed to the time-register and the sum of them is less than the maximum capacity of the circuit. Upon request, they can all be released as a single signal with a pulse width equal to the sum of all the stored inputs.

By using the same two time-registers and applying the inputs in different parts of the circuit, another possibility is to do time subtraction of two time pulses. As shown in Figure 20 in the time-subtractor configuration one variable goes into the input of the first time-register producing $T_{out} = T_{FS} - T_a$, that goes into the input of the second time-register. The second one also goes into the input of the second time-register. From Eq. (3.1) and Eq. (3.2), it is clear that by doing so, one time pulse is subtracted from the other, and the final output is $T_{out} = T_{FS} - ((T_{FS} - T_a) + T_b) = T_a - T_b$.

The implementation of the time-register is not very complex and can be done by serially connecting delay cells and forming a delay line, as shown in Figure 21 .

There are numerous ways of implementing a delay cell. The time-registers in the proposed MAC unit were implemented using the standard gated delay cell, also known as a current starved inverter, and the skewed gated delay cells [80, 81]. In this work, at

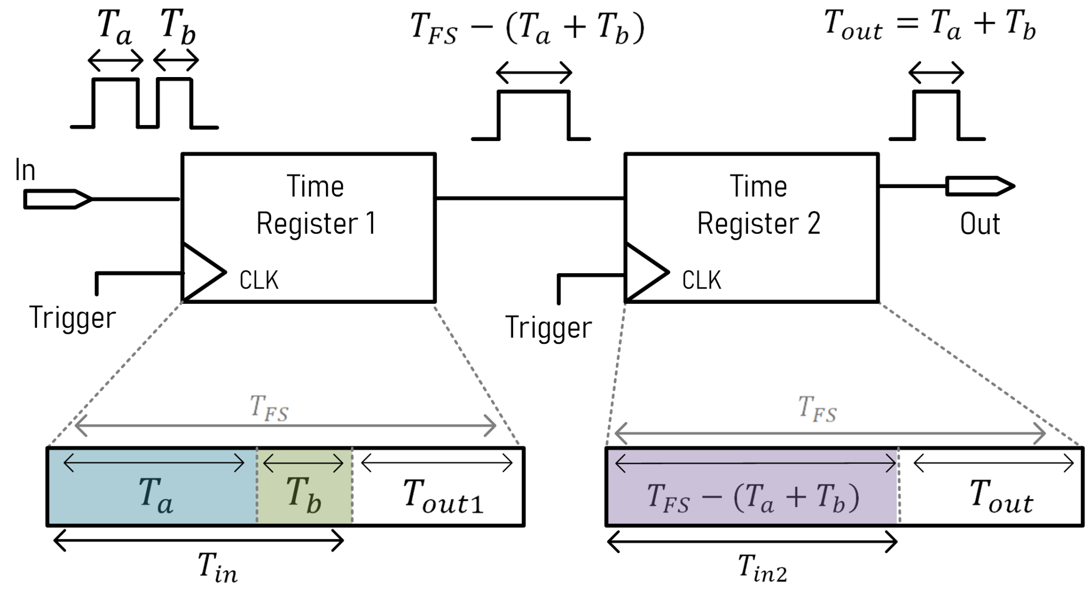


Figure 19 – Time-adder configuration

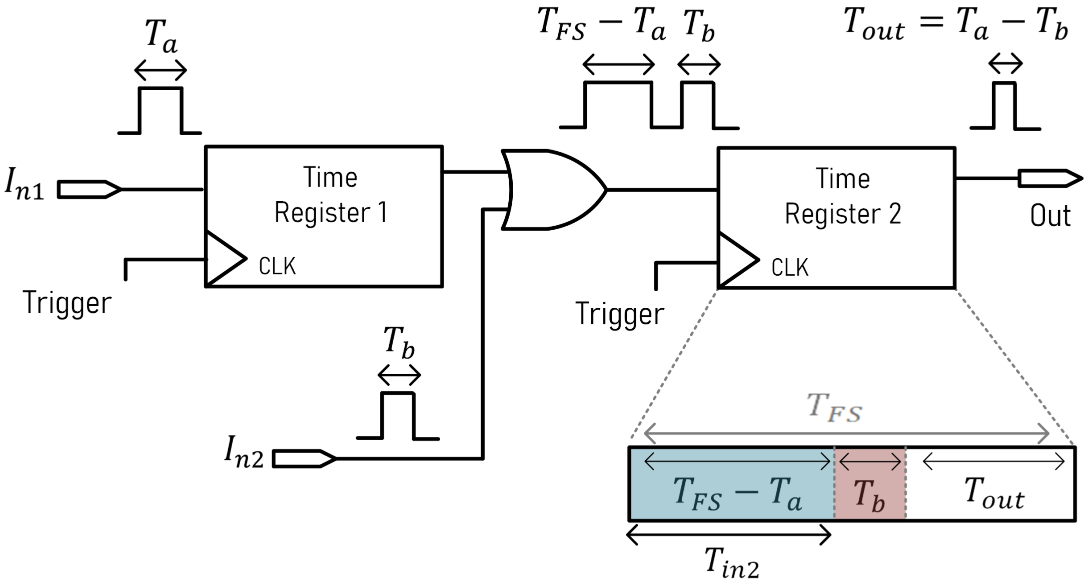


Figure 20 – Time-subtractor configuration

first, the time-register was implemented by using the standard gated delay cell, which schematics is shown in Figure 22. The standard gated delay cell, also known as current starved inverter, is a circuit composed of inverters, which are responsible for generating the delay (M1-M4), and control transistors (M5-M8), to enable or disable its operation. A capacitor can also be placed between the two inverter stages. Although it is not required, employing it reduces the leakage current and increases the delay generated by each cell.

The major advantages of this type of cell are the ease of implementation, the use of few transistors, and the large delay that can be generated in such a simple topology [82]. However, when used in time-adder and time-subtractor configurations, standard delay cells have a relatively high error, i.e. the recovered signal at the output differs slightly

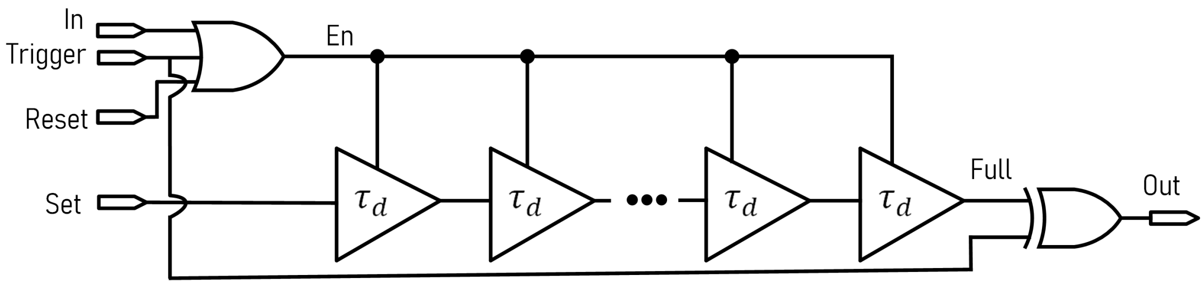


Figure 21 – Time-register convectional implementation

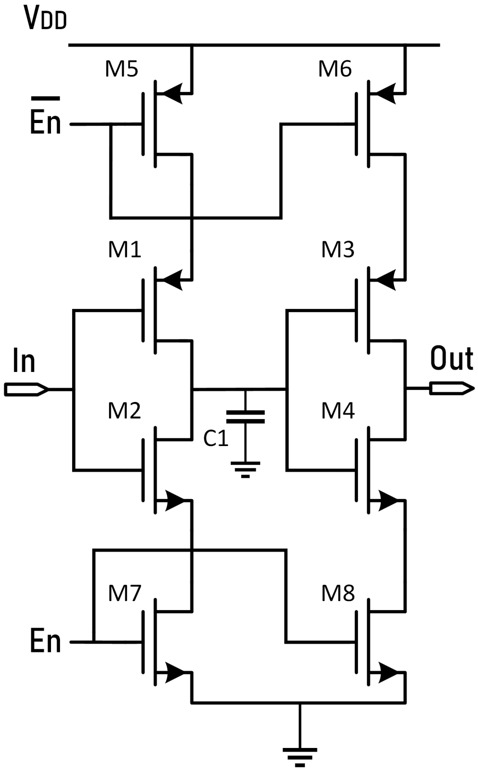


Figure 22 – Standard Gated Delay Cell

from the input signal. This error is the gating skew error caused by the unwanted phase shift when delay cells switch from write/propagation to hold mode [80].

An alternative to reduce the error is to use so-called skewed gate delay cells. These cells, shown in Figure 23, reduce the overall skew error by averaging the individual phase error contributions from each delay cell. However, because the delay cells are cascaded not only sequentially, but also taking into account previous stages, this topology has a more complex design and generates less delay than the standard one [78]. In short, a larger area is occupied, but a smaller error can be achieved. Because it takes into account the output of previous stages, the implementation of the time-register when employing skewed delay cells is a little bit different than the one using standard delay cells. In Figure 24, a clipping of the time-register designed with skewed delay cells is shown to illustrate this peculiarity. It is worth noting that if there is no previous stage, the input that should be connected to this stage is left open (e.g. for the first delay cell in the line the inputs In[n-3] and In[n-5] are left open).

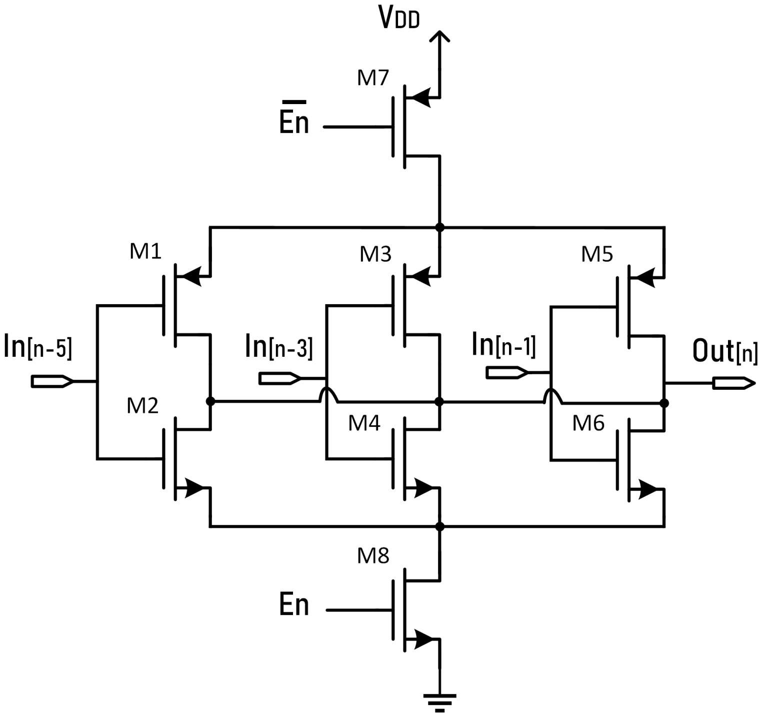


Figure 23 – Skewed Gated Delay Cell

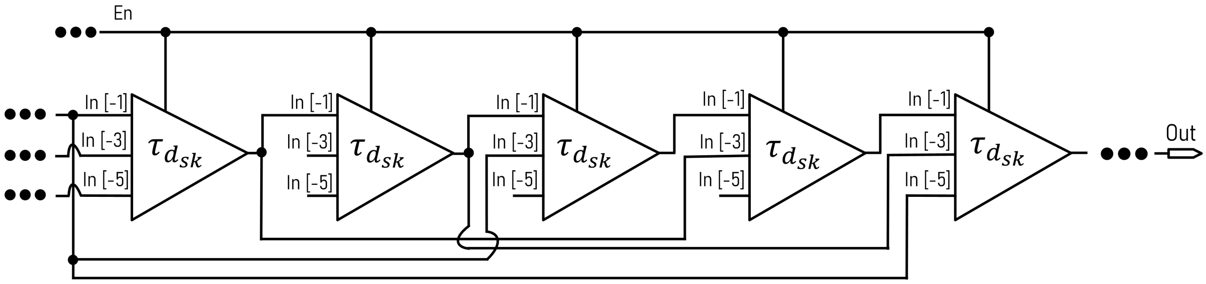


Figure 24 – Time-register with skewed gated delay cells implementation

An important detail is that for both delay cells, the bulk terminal is connected to the circuit's lowest potential for NMOS transistors and the highest for PMOS transistors

After detailing the delay cells, to better explain how the time-register works, at the transistor level, consider that the circuit is a set of delay cells and some logic gates, as depicted in Figure 21. The operation of the circuit is exactly as explained before, but now it is possible to include more details that are extremely important and will help in the understanding of the MAC unit further.

Each delay cell is responsible for generating a delay time equivalent to T_d. This means that the full scale T_{FS} of the time-register can be defined as $N \times T_d$, with N being the amount of delay cells. When either I_n, $Trigger$ or $Reset$ is high, the Enable signal (E_n) turns on and the signal transistors propagate whatever the Set value is, which is done by increasing the phase of the delay line. If none of these signals are high, the phase is simply held. As mentioned before, when *Set* is high, it is possible to be in the Write (I_n high), in the Read ($Trigger$ high) or in the Hold (I_n low, $Trigger$ low) mode of operation. However, if *Set* is low this means that zeros will be propagated in the delay line whenever the *Enable* signal is set to high and all the information previously stored is erased. This is known as the reset mode of operation. In this mode, a signal *Reset* is created and it can enable the propagation of zeros in the delay line. This is useful for allowing the time-register to operate multiple times, since it can store the information, release when needed and then reset, allowing new signals to be stored. Another important aspect to highlight is that the output time is generated by the time difference between the $Trigger$ and the *Full* signals. This can be done by connecting them both into an XOR logic gate [79], so its output will be high after $Trigger$ is released until the full scale is reached (Equation 3.1). To summarize the complete operation of the time-register, in Figure 25 a timing diagram is shown.

3.2.2 Gated ring oscillator

The traditional single-ended ring oscillator is a circuit composed of an odd number of inverters in a ring, whose output oscillates between two voltage levels, usually VDD (logic level = 1) and GND (logic level = 0). As shown in Figure 26, the inverters, or NOT gates, are connected in a chain, with the output of the last inverter feeding back into the first.

In the proposed time-domain multiplier, however, the ring oscillator is used as an artifice to discretize one of the input times (T2). More details will be provided further in Chapter 4. To accomplish this, some changes can be made to the traditional ring oscillator to enable it to operate only when necessary, i.e. when one of the input times (multiplier T2) is high. This can be accomplished by designing a gated ring oscillator (GRO), by adding control transistors to the traditional topology, which will enable or disable the circuit's operation. As a result, whenever the enable (En) signal is high, transitions occur, and when this signal is low, the oscillator's current state freezes. The gated ring oscillator is shown in Figure 27.

The frequency associated to the transitions is defined by Equation 3.3:

$$f = \frac{1}{2 \times n \times \tau_s} \tag{3.3}$$

In which f is frequency of GRO, n is number of stages of delay cells and τ_s is delay associated with each delay cell.

Furthermore, because the main goal of the circuit is to discretize a time pulse, the interest lies more specifically on the number of transitions during a given amount of

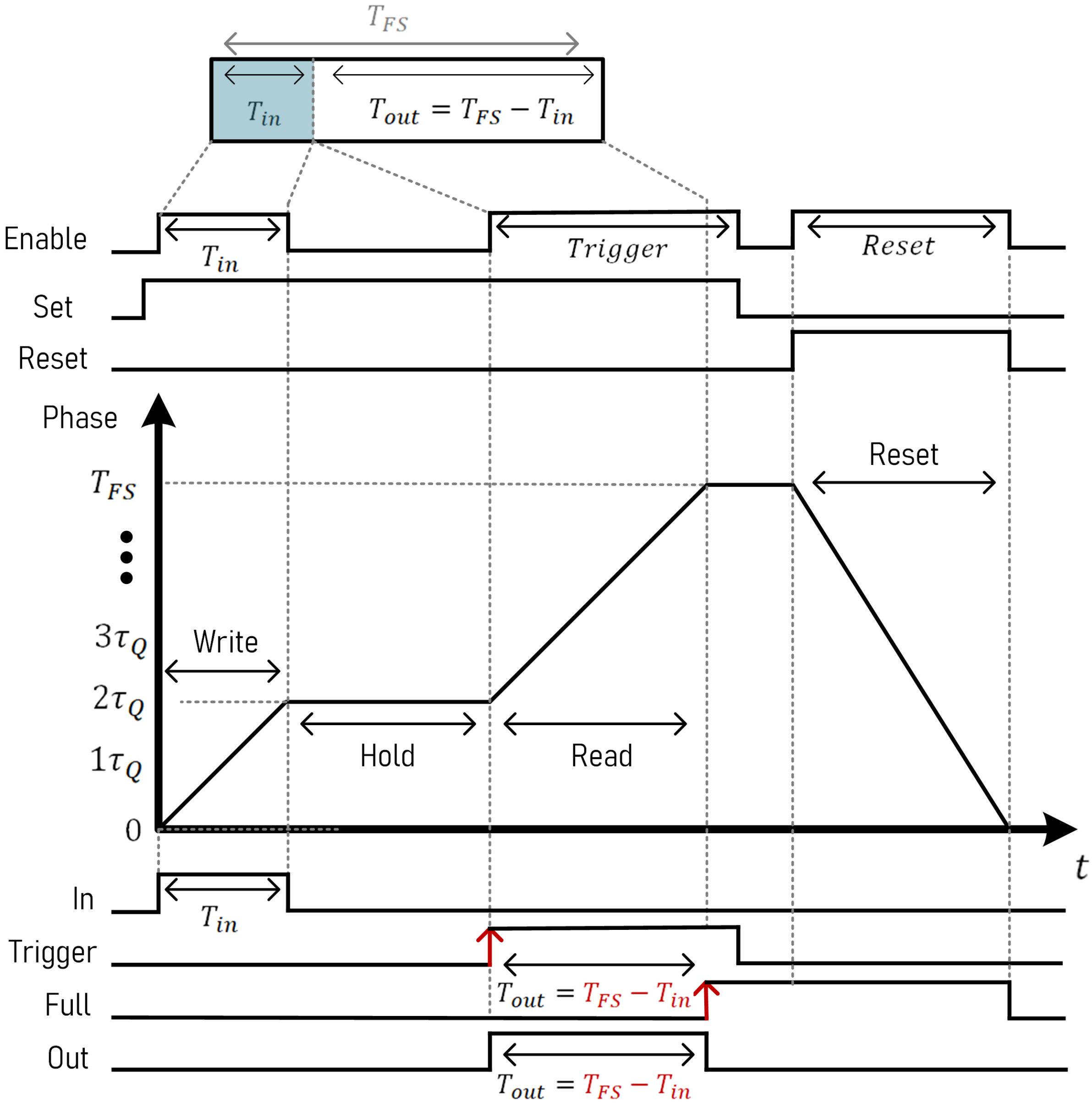


Figure 25 – Time-register timing diagram

time. This parameter can be obtained by taking the inverse of the product of the oscillator frequency, f, and the input time pulse T_{in}, as shown in Eq. (3.4):

$$transitions = \frac{1}{f \times T_{in}} \tag{3.4}$$

Therefore, by using the GRO it is possible to obtain a direct relationship between the input pulse width and the resulting transitions. In Figure 28, this GRO function is illustrated. The larger the pulse of the input time i.e. multiplier T2, the more oscillations can be seen in the output.

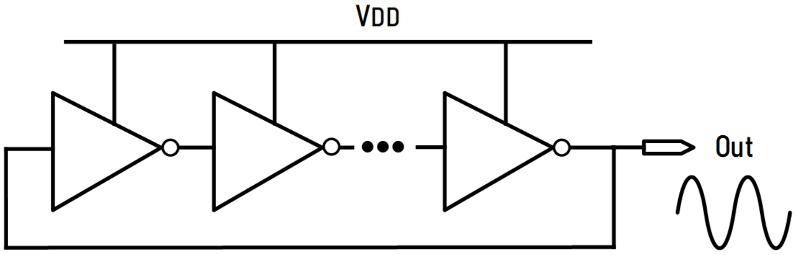


Figure 26 – Traditional single-ended ring oscillator

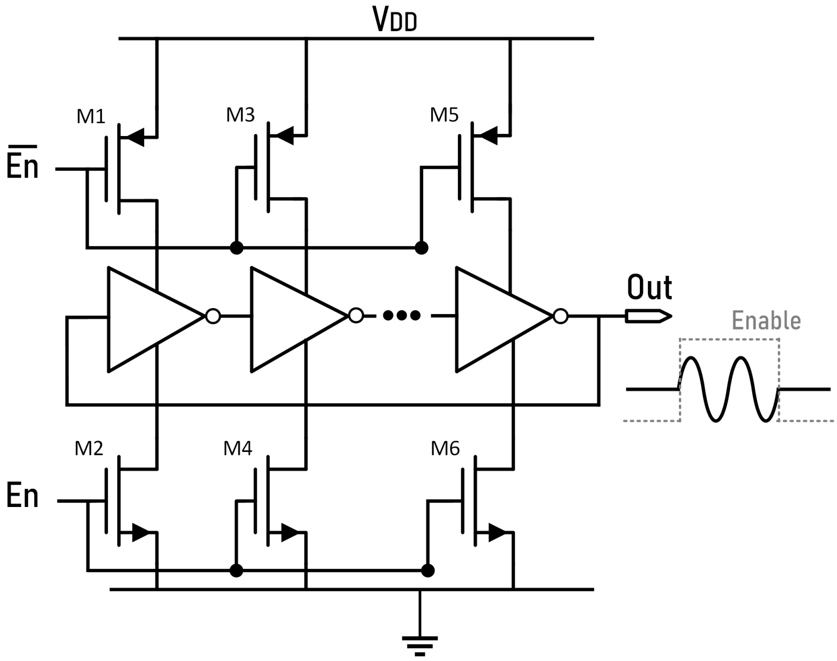


Figure 27 – Gated ring oscillator

3.2.3 Bidirectional counter

The last major block needed to design the time-domain multiplier is the Bidirectional counter, which is a circuit capable of both up counting and down counting. This circuit performs two important functions in the multiplier, counting and synchronization, which will be performed according to its mode of operation. The circuit can be designed by using JK flip flops and logic gates [83], as shown in Figure 29.

In general, the counter has three input terminals, besides VDD and GND, which are Enable (En), Clock (CLK) and Mode. The number of outputs is variable, since the circuit presents the same number of outputs as bits (e.g. a 3-bit counter has 3 outputs). However, since the single-bit outputs are displayed in parallel it is convenient to treat the counter as having a single multi-bit output (count), facilitating its interpretation as a decimal value, as depicted in Figure 29. The enable terminal is responsible for controlling whether the counter will be operating (En=1) or frozen (En=0) in its previous state. The frequency of operation is controlled by the clock input which can be different for each mode

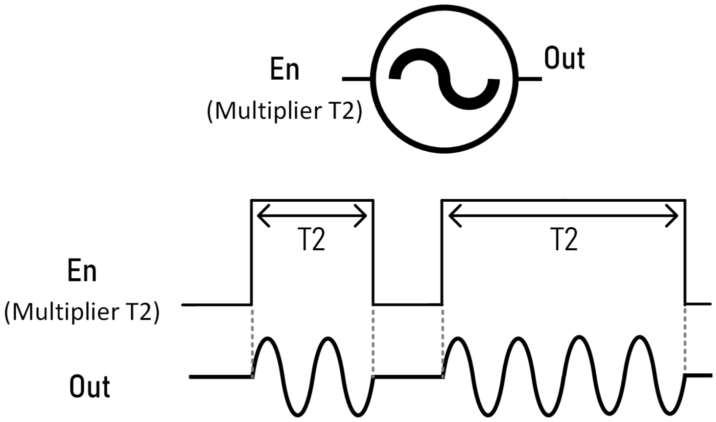


Figure 28 – Gated ring oscillator operation

of operation. As will be discussed later, when inserted into the time-domain multiplier, the bidirectional counter CLK input will assume two distinct frequencies, one referring to the up count mode and one to the down count mode.

The first mode of operation is the up count mode (Mode=1), in which the circuit is responsible for counting the output transitions of the oscillator (Out, from Figure 28), incrementing its value by one after each transition. However, due to the high frequency of the GRO supplying the counter's clock input, the counting process in this mode of operation will be rapid, as it is done at the clock frequency. In order to allow the multiplicand and the multiplier to have the same order of magnitude, the down count is needed to reduce the frequency of the signal and synchronize it with the multiplicand branch of the time-domain multiplier. To do so, when down counting, a clock with much lower frequency is fed into the circuit as presented in Figure 16. Therefore, in the down count mode of operation (Mode=0) the value previously accumulated due to the up count is decremented until it reaches zero. When this occurs, it indicates that the operation is complete, and the bidirectional counter is reset to allow a new operation to be executed.

It is important to highlight that the operational mode is determined by the T2 multiplier, which serves as the enable input of the GRO. When this signal is pulsed, the counter switches to the up count mode, indicated by the Mode input of the counter being set to a high level. However, once the T2 pulse concludes, the GRO ceases to oscillate, resulting in the Mode input transitioning to a low level and initiating the down count mode.

The complete operation of the bidirectional counter can be seen in Figure 30, which exemplifies the behaviour of the GRO and counter's most important signals in a timing diagram. Note that the count signal is given by combining all the output bits of the counter. That is, considering a counter of N bits, like the one shown in Figure 29 for "0...00", count = 0, for "0...01" count = 1, for "0...10", count = 2.

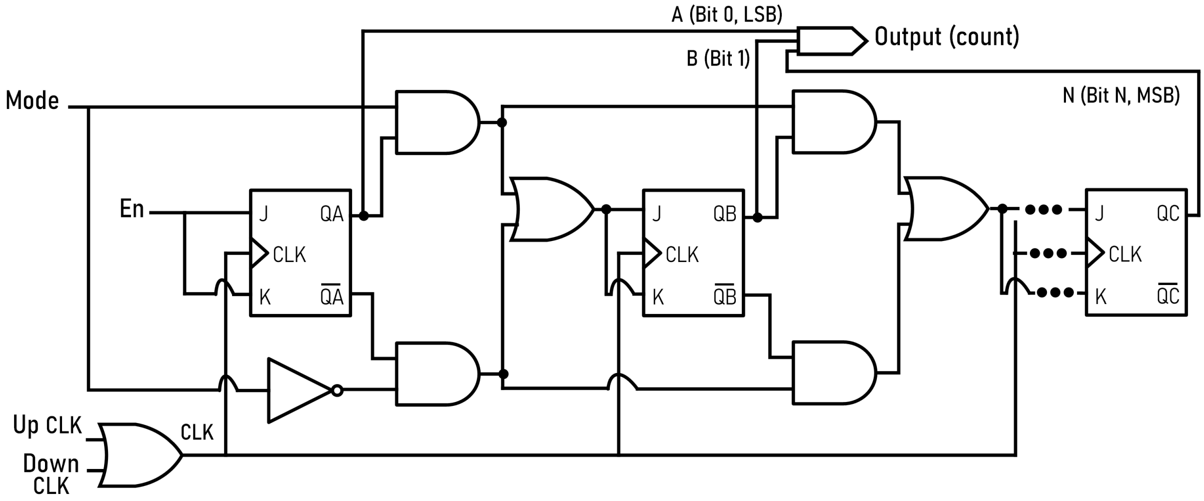


Figure 29 – N-bit multi-frequency bidirectional counter circuit implementation

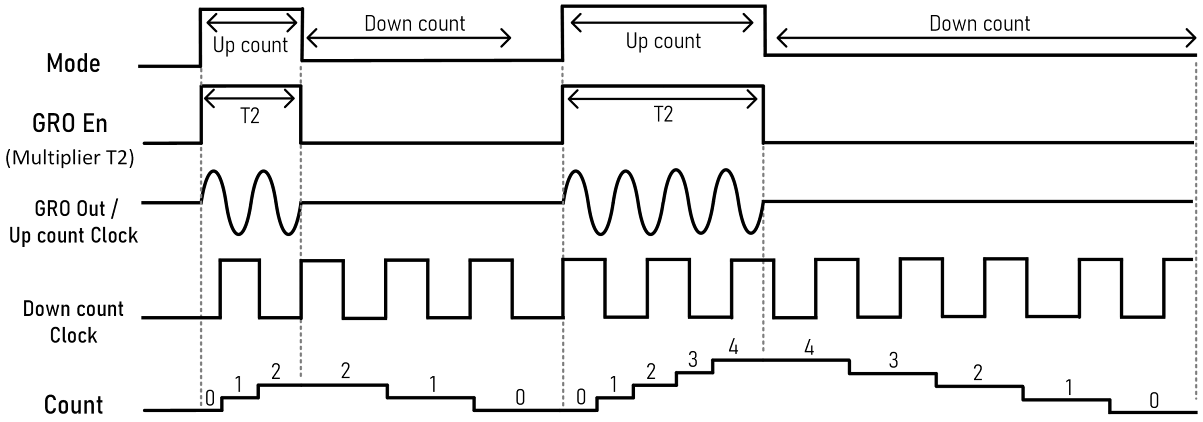


Figure 30 – Bidirectional Counter operation

3.3 Time-domain multiplier operation

After detailing each block present in the time-domain multiplier, finally the complete topology can be presented in Figure 31 , with intermediate signals in each critical part of the circuit being highlighted. The multiplier's operation principle will be thoroughly detailed in this section, allowing readers to easily comprehend the entire MAC unit.

As previously stated, the circuit's core principle is to enter with two time signals, T1 and T2, and obtain an output, also in the time-domain, proportional to the product of the inputs. The entire process is accomplished through successive additions, meaning that the multiplicand T1 will be added repeatedly based on the multiplier T2. The operation of the circuit can be broken down into three main steps: Multiplicand storage, feedback control and final addition.

In the first step, which is called multiplicand storage, the time signal T1 is added to the feedback signal by using two cascade time-registers (Time-register 1 and 2). During the first rising edge of the input signal T1, there would be no feedback signal, causing

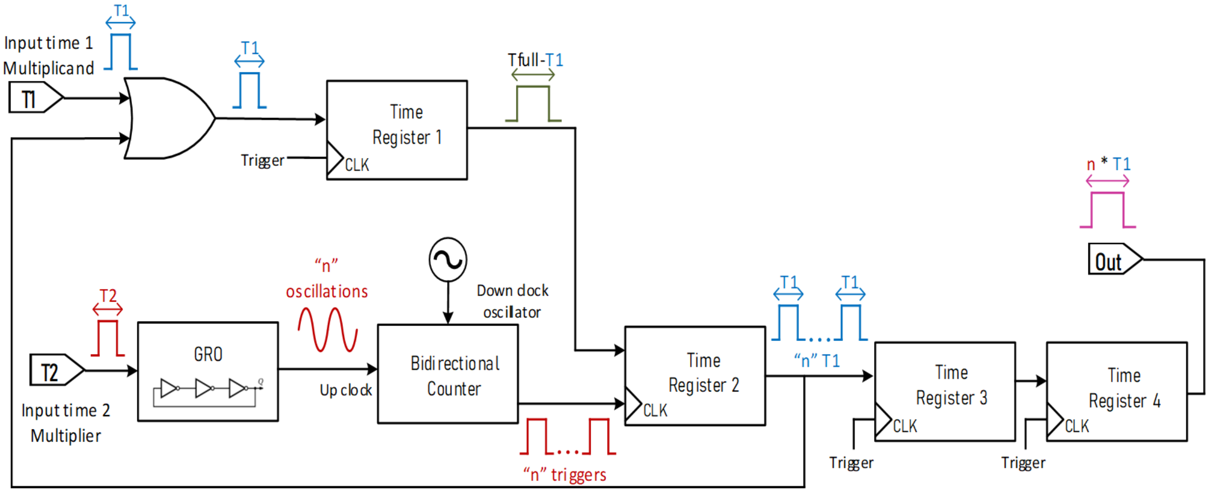


Figure 31 – Proposed Multiplier architecture

itself to be stored in the final time-adder (Time-registers 3 and 4). Besides being stored, the signal T1 also comes back in the feedback loop, which is critical because no new signal is applied to the circuit after the first cycle. This means that in the first cycle signal T1 is being fed to the multiplier, however, from the second cycle on, this very signal T1 is confined in the circuit, being stored and returning as feedback during each cycle. This procedure is illustrated in Figure 32, where a multiplication up to three is represented. The number of times this will occur is determined by input time T2, which controls whether the T1 signal should be returned as a feedback or whether the final addition can be made.

The second step, which is called feedback control, occurs concurrently with the multiplicand storage, but in the multiplier branch. This part of the circuit has the main function of deciding whether signal T1 should be returned as a feedback signal, allowing a new successive addition to be made, or whether multiplication is terminated. In brief, this control is accomplished by means of the pulse width of the input signal T2. As demonstrated repeatedly, the GRO converts the pulse width of T2 into oscillations, which are then registered by the bidirectional counter, in the up count mode. In the down count mode each new addition results in a decrease of one out of the total value stored in the counter. When this value reaches zero, it indicates that no more successive additions are required because the multiplication operation has been completed successfully. The trigger signals that feed Time-Register 2 are thus disabled, and the multiplicand (T1 signal) is no longer returned as feedback. Hence, the larger the T2 pulse, the more oscillations the GRO displays, the higher the count accused by the counter, and finally, the more triggers are available for performing the successive additions. In Figure 33, this stage is shown with more details, also representing a multiplication by three.

Following the completion of the previous two steps, the final step consists of the final addition and is quite simple. As explained above, at each cycle of operation, the T1 input is successively being stored in Time-Register 3, which together with Time-Register

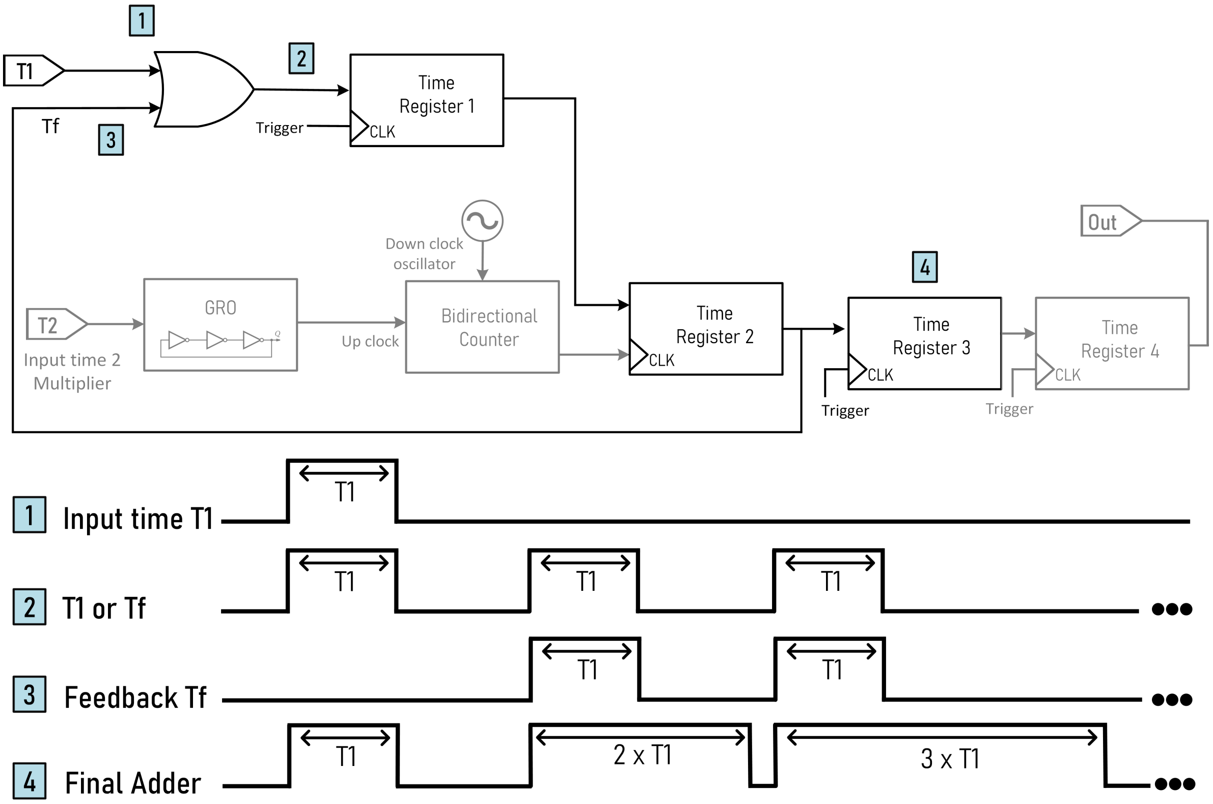


Figure 32 – Multiplicand storage conceptual timing diagram

4, acts as the final adder. With the end of multiplication, i.e., when it is determined that there are no more triggers available to bring the T1 signal back into feedback, no more signals are stored and the final summation can take place. In short, in the final addition T1 is added "n" times, with the number of times being proportional to the pulse width of T2. After this step, the output, which is a time signal proportional to the product of the two input signals, is finally released. It is worth mentioning that the proposed topology is robust to fluctuations in the timing of the control signals. As a result, whether the trigger signal is activated either before or after the intended timing, the circuit continues to function properly.

The complete operation of the multiplier can be seen in Figure 34, which shows the timing diagram of a multiplication by three.

3.4 Complete MAC unit: concept and operation

Following the description of the multiplier's concept and operation, in this section is presented the designed MAC unit, which is the ultimate goal of this thesis. The developed concept, shown as a block diagram in Figure 35, consists in a novel approach for performing the operation of multiply and accumulate in the time-domain.

The designed MAC unit is composed of two larger blocks: the multiplier and the

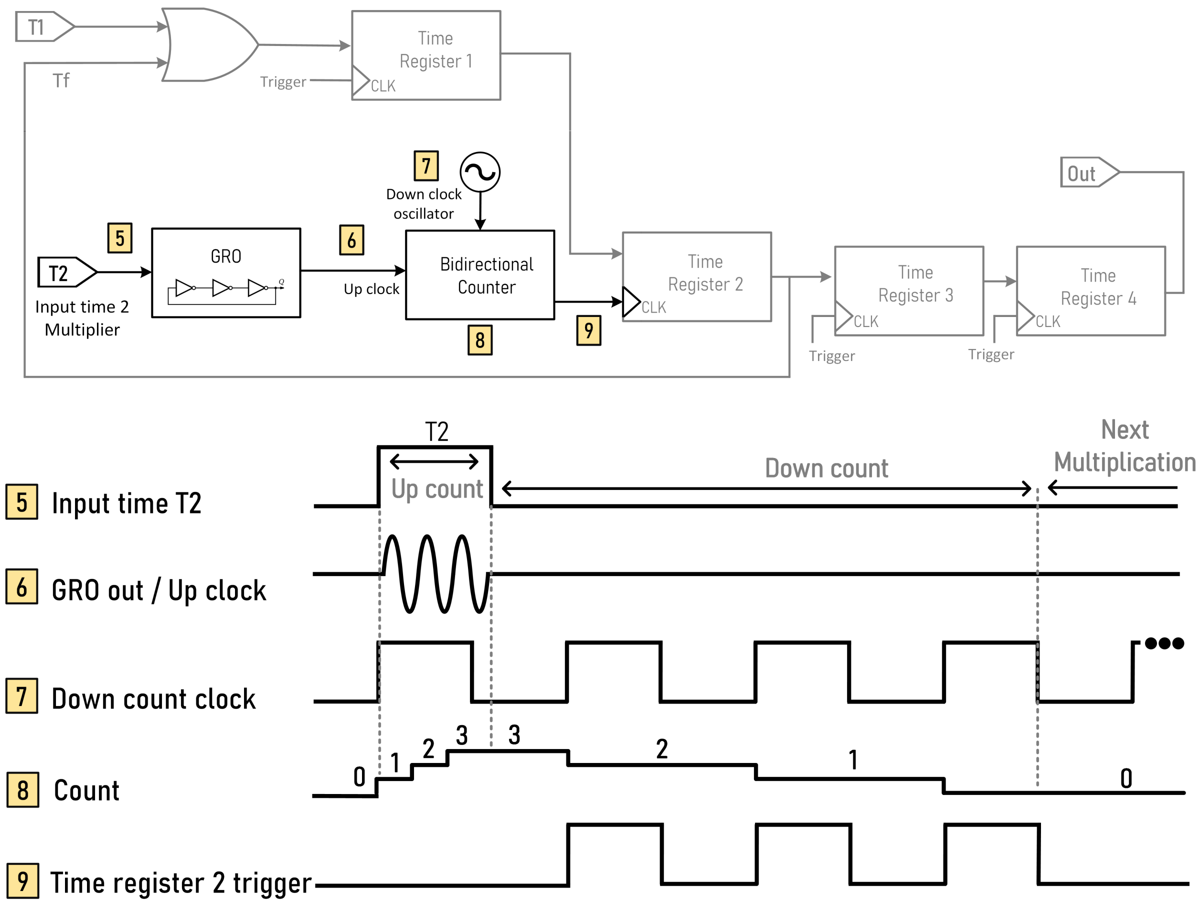


Figure 33 – Feedback control in the multiplier branch conceptual timing diagram

adder and accumulator. Aside from the domain shift, the time-domain MAC unit differs from conventional digital ones, especially in terms of addition and accumulation. In digital MAC units it is customary to do these operations separately, in different blocks. However, due to time-domain properties, it is possible to do both by simply using a time-adder, which is a circuit that can not only add but also store signals. Performing both addition and accumulation in the same circuit is very advantageous, as it allows for the use of fewer components and facilitates the layout routing, consuming less power and taking up less silicon area.

Two versions of the time-register, employing different delay cells, were implemented to be used in unique parts of the circuit because of their different performances, as discussed in Section 3.2.1. For the multiplier the time-registers were designed with skewed gated delay cells (Figure 23). While this cell has less error, it also generates less delay, which limits the ability to store time signals for a given silicon area, or increases the required silicon area for a given time delay. In fact, skewed cells are the best option for the multiplier because, due to the principle of successive additions, the input signals will pass through the multiplier's

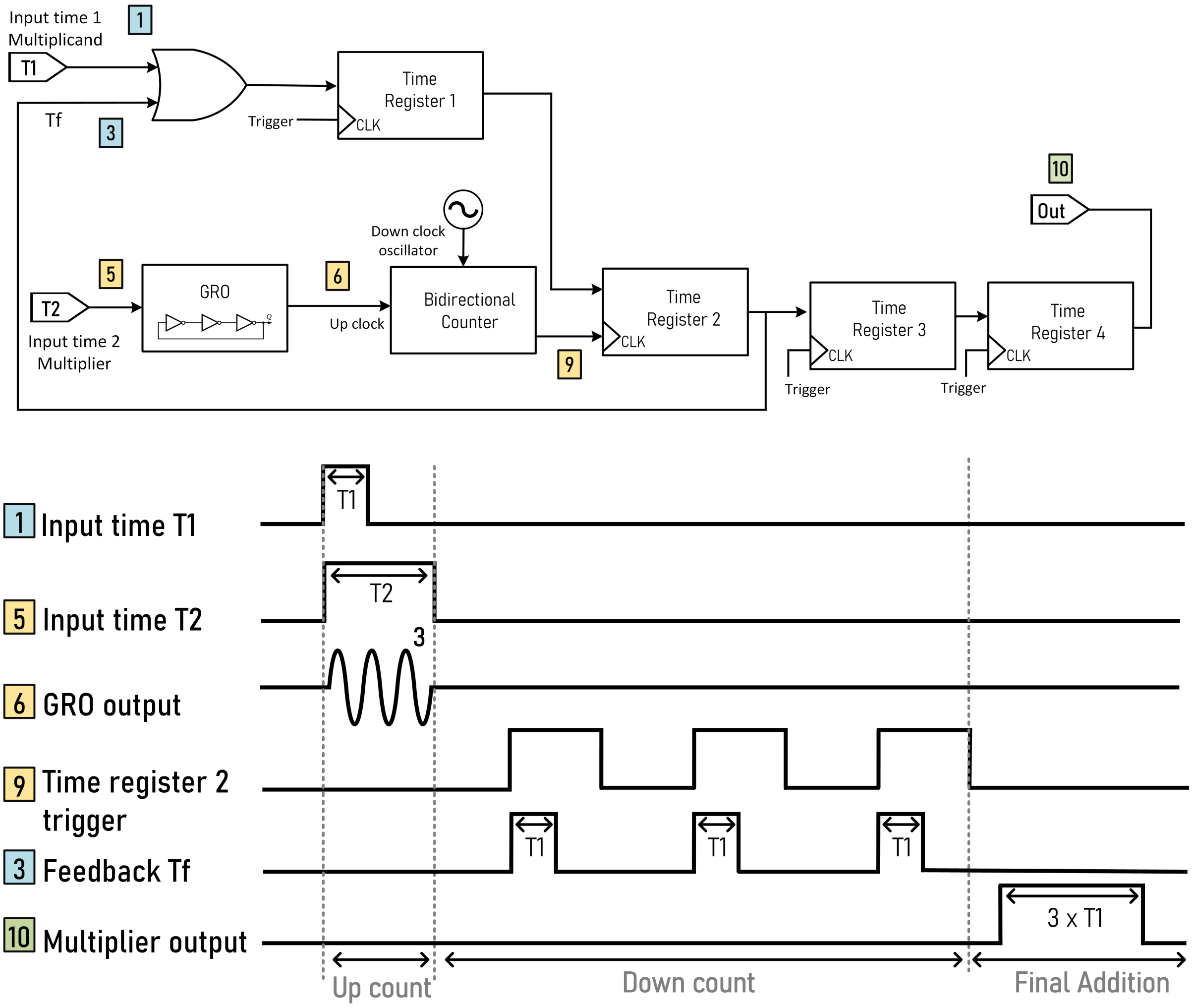


Figure 34 – Complete multiplier conceptual timing diagram

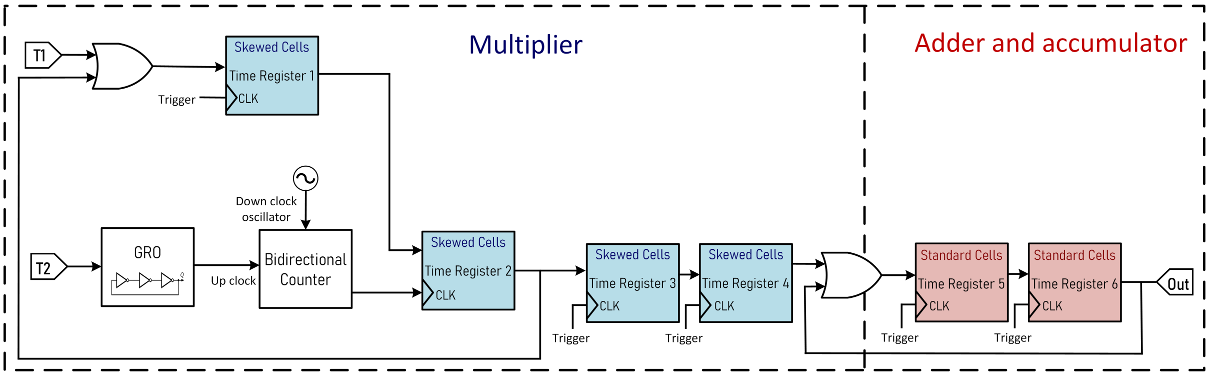


Figure 35 – Conceptual block diagram of the proposed MAC unit

time-registers several times, resulting in an increasing cumulative error. However, the time-registers will be much less active for addition and accumulation operations, being activated only once at the end of each multiplication, resulting in less cumulative error. Furthermore, it is critical that this circuit, in the adder and accumulator block, has adequate storage capacity, so that the MAC unit has a wider range of operation. The multiplier, on the other hand, requires less storage space because it works with narrower time pulses. As a result, for the adder and accumulator, standard gated delay cells (Figure 22), which have slightly higher error but greater storage capacity are appealing.

After explaining the multiplier operation in Section 3.3, it is simpler to detail the functioning of the MAC unit. At each cycle of its operation, the three operations, multiplication, addition, and accumulation are performed. Initially, the product of the two input times, T1 and T2, is computed by successive additions of T1, which are controlled by T2. At the end of the multiplication the result obtained is added with the previously accumulated values, which are basically the sum of previous multiplications. The sum is calculated using the OR gate, whose inputs are the multiplier output and the MAC unit output, as well as Time-Registers 5 and 6. As previously stated, these registers in series serve as adder and accumulator, therefore the sum is performed in Time-register 5 and the result is stored in Time-register 6. The value stored in this last Time-Register can be released at two different times, depending on whether a new cycle is about to begin or the operation is already finished. When the MAC unit starts a fresh cycle of operation, that is, when a new multiplication is being performed, the previously accumulated result is released to be added to the result of the new multiplication. However, when there are no more pending multiplications, the multiplication and accumulation operation is complete, and the final result is released to the circuit's output. It is also worth noting that in the first cycle of the MAC unit operation the accumulated result is null , so the result of the first multiplication is added to zero, implying that it is simply stored.

To illustrate the operation of the MAC unit and facilitate its understanding, a timing diagram is shown in Figure 36. In this diagram two cycles of MAC operation are being represented. In the first one the input time T1 is being multiplied by T2, whose pulse width is equivalent to a multiplication by three. After the multiplication is done, the multiplier output, $3 \times T1$, is added to the previously accumulated values, zero, and them stored. At the same time as the sum and accumulation operation is being performed, a new multiplication is also taking place, ensuring that the MAC unit's throughput is as high as possible. In this example, the second multiplication consists in multiplying the same input time T1 by a T2 pulse equivalent to two. At the end of this second multiplication, the multiplier output, $2 \times T1$, is added with the stored signal in the adder and accumulator, $3 \times T1$, and then stored in Time-Register 6. Since there are no more multiplications to take place, the stored time signal is released as $5 \times T1$ $(2 \times T1 + 3 \times T1)$, marking the end of the second operation cycle and the MAC operation.

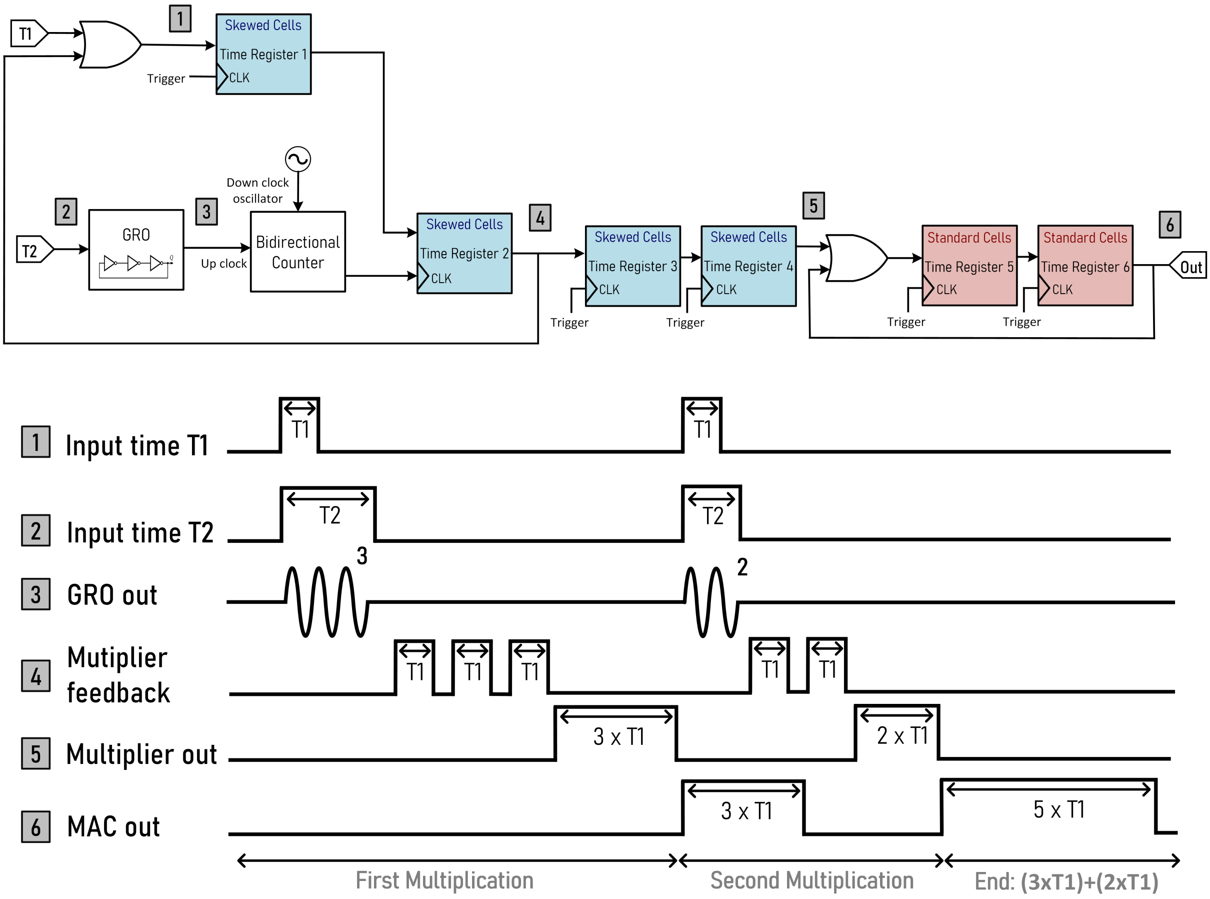


Figure 36 – Proposed MAC unit timing diagram

Chapter 4

Results and analysis

In Chapter 3 a detailed implementation of the time-domain MAC unit along with its sub-blocks was presented. Even though the major contribution of this work is in the proposition of a new concept to perform multiply-accumulate operations solely with time signals, the time-domain MAC unit was designed as a proof of concept. Thereby, the emphasis of this chapter is on displaying the design specifications for all of the circuits that comprise the MAC unit, as well as the corresponding simulation results.

To comprehend the behaviour of the circuit a set of simulations and analysis will be carried out in this chapter. More specifically, simulations aimed at evaluating the linearity and error of the multiplier and MAC unit will be performed, for some range of scenarios and operating points. Process corners and temperature simulations will be conducted in order to demonstrate the robustness of the developed circuit. Furthermore, digital MAC units were also designed to serve as a basis of comparison for the time-domain MAC unit that has been proposed.

The simulations were performed in Cadence Virtuoso: IC Design software, using TSMC's 180-nm CMOS process technology [84]. The main characteristics of this technology are a minimum channel width of 180 nm for the transistor design and an operating voltage of 1.8V.

4.1 Sub Blocks: Schematics and simulations

4.1.1 Gated Ring Oscillator and Bidirectional Counter

The designed gated ring oscillator consists of nine stages and operates at 170 MHz, performing one oscillation every 5.8 ns. As shown in Figure 37, the circuit is made up of inverters, just like typical ring oscillator topologies, and enable transistors, to control whether it should oscillate or not. The specifications for all the GRO components are detailed in Table 1.

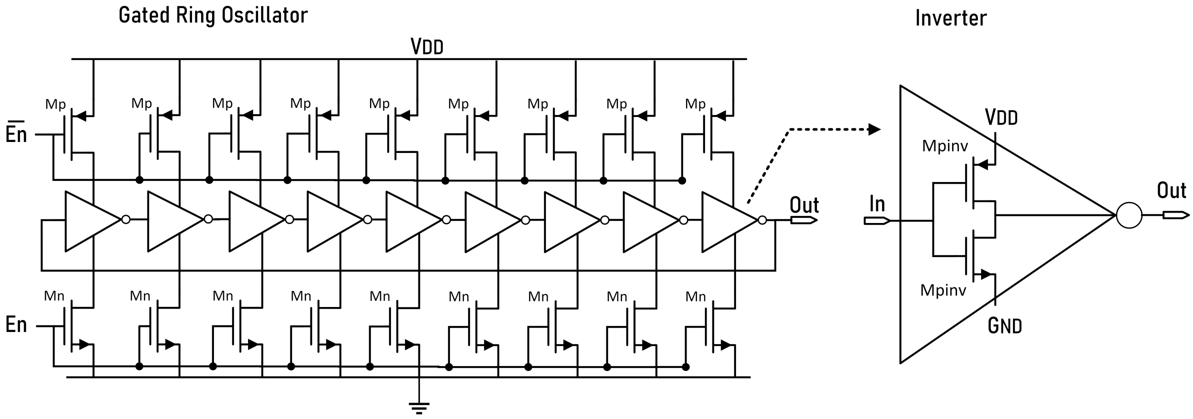


Figure 37 – Gated ring oscillator designed

Table 1 – Gated ring oscillator specification (Refer to Figure 37)

Component	**Specification (W/L) [m]**
M_N	$600\,n/180\,n$
M_P	$1.8\,\mu/180\,n$
M_{Ninv}	$3\,\mu/600\,n$
M_{Pinv}	$7\,\mu/600\,n$

To demonstrate the GRO's behaviour the circuit was simulated for different time pulses as shown in Figure 38. It is worth noting that the number of oscillations are proportional to the enable (En) pulse widths, remembering that, when employed in the MAC unit, the multiplier T2 is the one to be fed to this terminal. The larger the pulse, the more oscillations the circuit will produce. Furthermore, when the enable signal is low, the circuit does not oscillate.

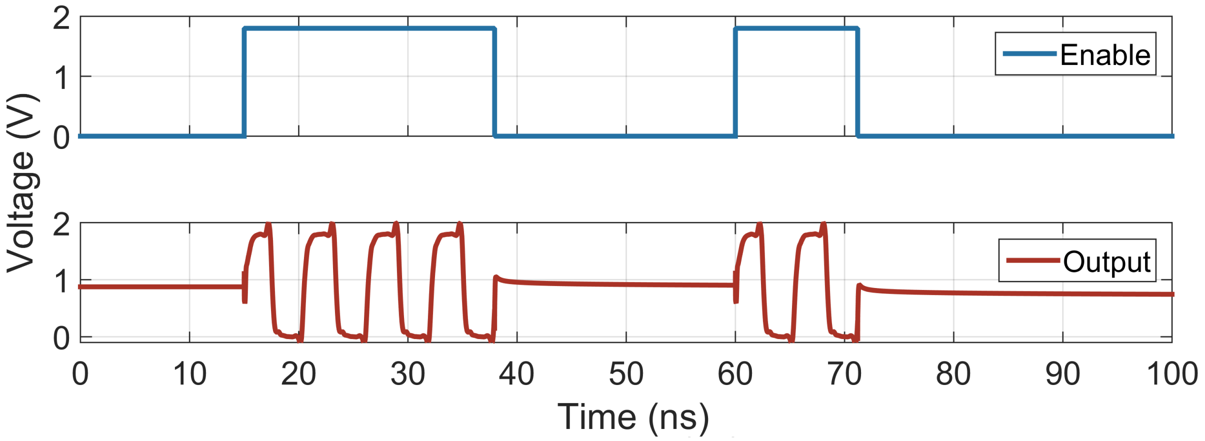
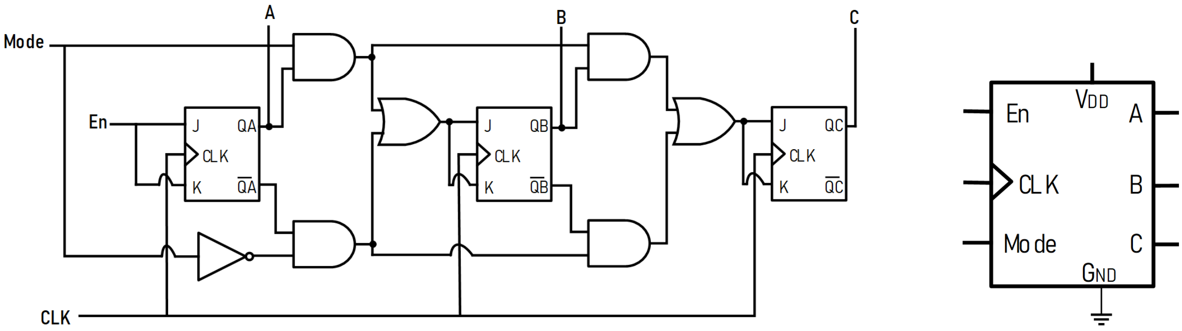


Figure 38 – Gated ring oscillator simulation

To allow oscillation counting, the bidirectional counter is used. It was designed a 3-bit bidirectional counter, which means that the circuit is able to represent eight values, counting from zero to seven. Since the circuit is essentially digital, identical transistors were used in the design ($W_{nmos} = 900\,nm$, $L_{nmos} = 180\,nm$, $W_{pmos} = 2.7\,\mu m$, $L_{pmos} = 180\,nm$). A detailed implementation and a symbol view of the circuit, highlighting its inputs and outputs is shown in Figure 39. It is worth highlighting that in the time-domain MAC unit, the counter's single-bit outputs are combined into a unified multi-bit output, as illustrated in Figure 29. This consolidated output serves as a trigger for time-register 2, enabling its proper operation within the system.



(a) Circuit implementation (b) Symbol view

Figure 39 – 3-bit Bidirectional counter

The bidirectional counter operation is shown in Figure 40 . As it can be seen, the circuit counts upwards when the signal mode is high and downwards when it is low. Both are done at the clock frequency. However, as a reminder, the main advantage of this type of circuit is that it allows the up and down counts to be performed at different frequencies. When considering how this circuit works in the time-domain multiplier, the idea is to count the oscillations coming from the GRO at a high frequency, whereas in down count mode the counting is done at lower frequencies.

To better demonstrate the counter operation, the circuit was simulated together with the GRO, as shown in Figure 41. For didactic purposes, Figure 41 assumes a single clock frequency. However, it is important to note that during the complete operation of the time-domain multiplier, two distinct frequencies will be utilized: one for the up count and another for the down count, as will be demonstrated further below. Essentially, these two components will convert a time pulse into a discrete number (with "A" being the least significant bit), which can later be decremented in the frequency of the down count oscillator.

The operation of the GRO together with the bidirectional counter can be seen in Figure 42. First an input pulse is applied to the GRO, which then oscillates throughout its entire width. In that moment, the bidirectional counter is in the up count mode of operation, which means that the GRO oscillations will be counted incrementally. For this simulation, the input time pulse width corresponds to four oscillations, so the counter count up to four (C=1, B=0, A=0) at GRO's frequency. After all oscillations have been

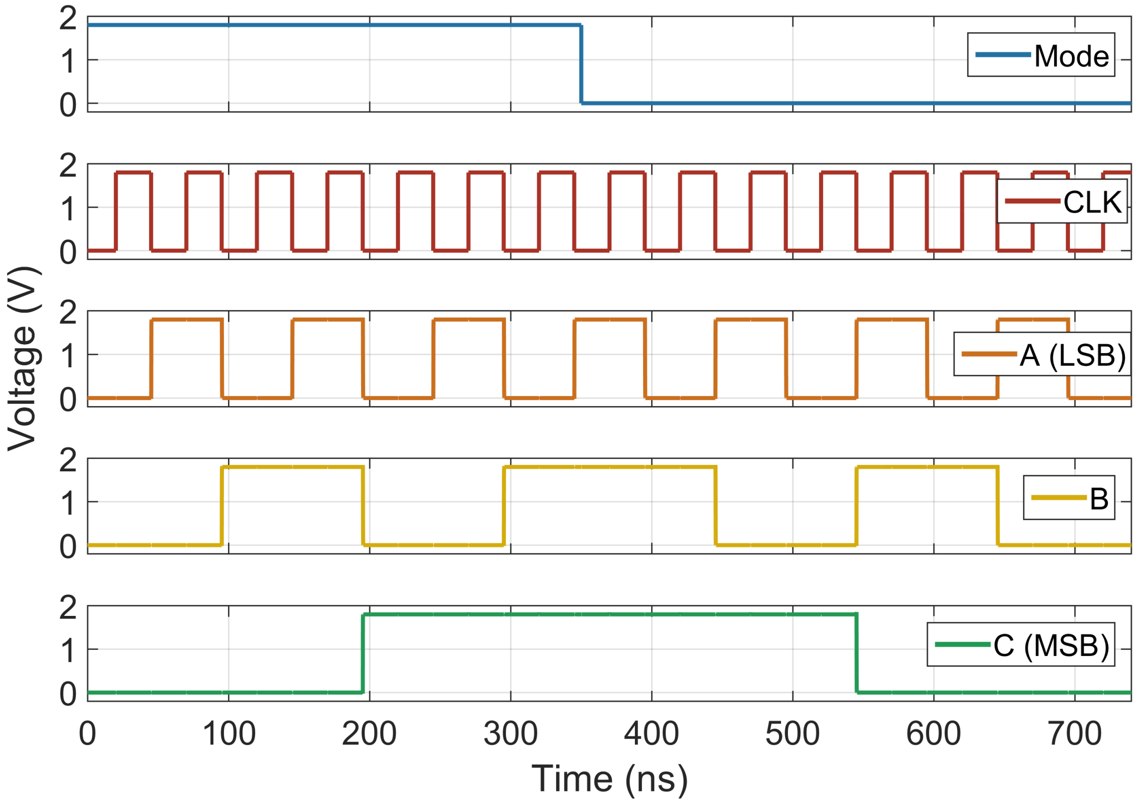


Figure 40 – Bidirectional counter operation

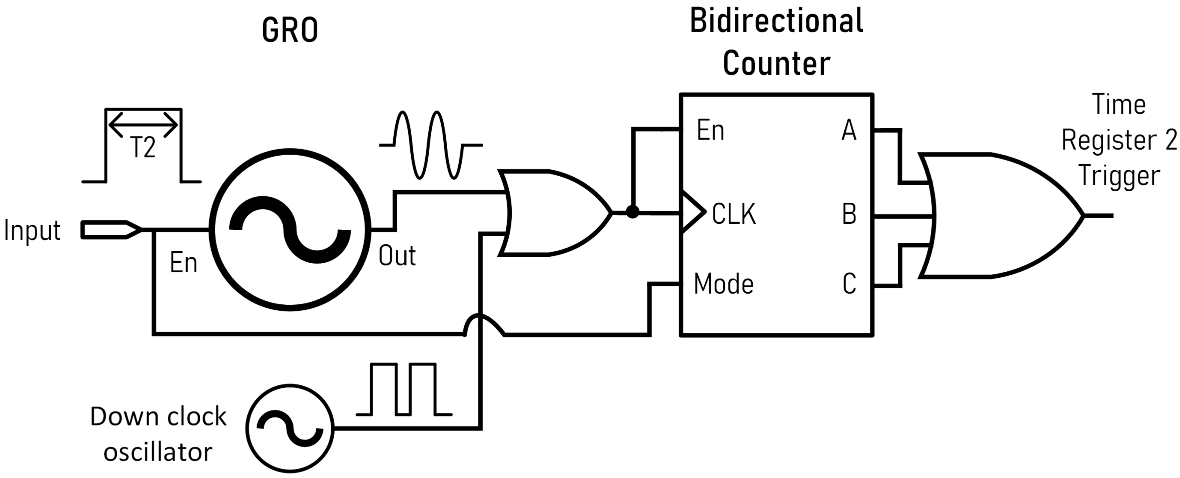


Figure 41 – Simplified GRO and bidirectional counter configuration simulated

counted, the bidirectional counter enters the down count mode of operation and the value stored in the circuit is decremented by one after each clock pulse. The frequency at which this is done is determined by an oscillator exclusively to this mode of operation.

As previously stated, when considering the operation of the GRO and bidirectional counter in the time-domain multiplier block, the down count is done at lower frequencies than the up count so that successive additions can be performed correctly. Note that in this mode of operation, the CLK input of the bidirectional counter is exactly the down clock. For each successive addition performed, the count is decremented by one. When this value reaches zero (C=0, B=0, A=0), it indicates that the multiplication is over.

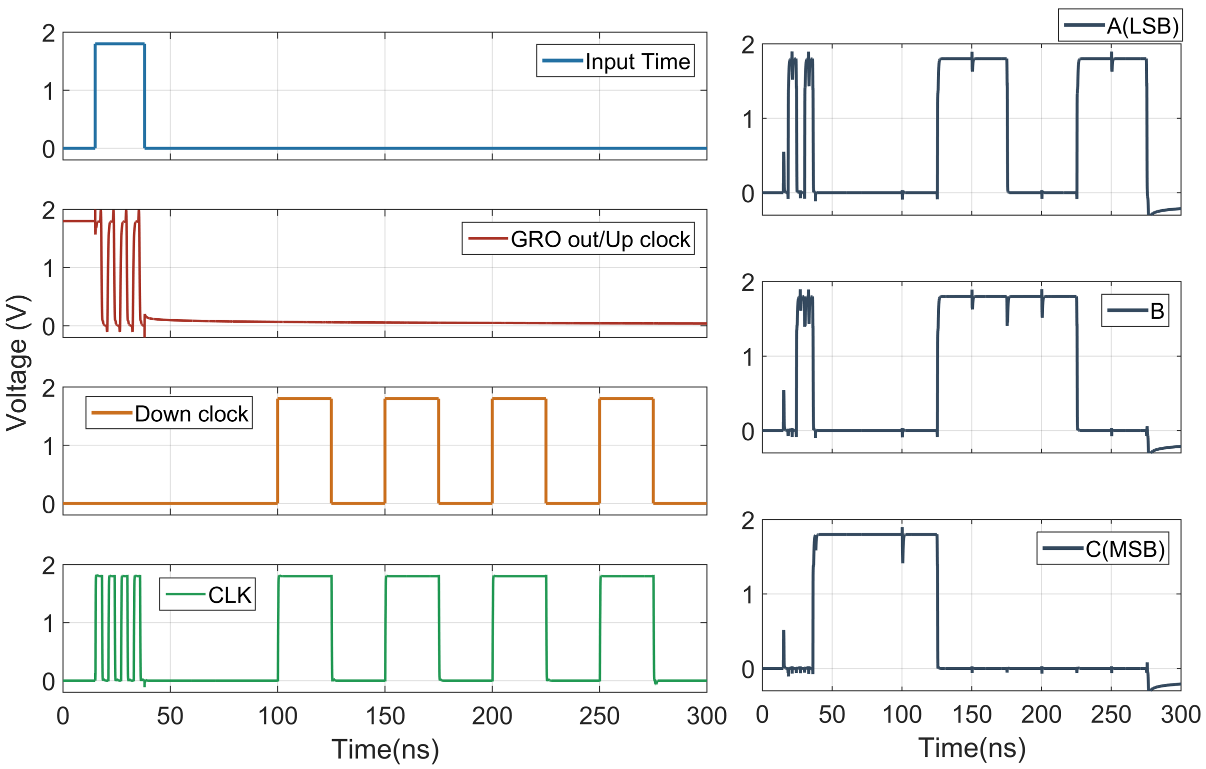


Figure 42 – GRO and bidirectional counter operation

4.1.2 Time-Registers

The core circuit that enables time-register operation is the delay cell. Other than that, logic gates must be used, primarily to control its operation. As previously explained, two different topologies of delay cells and, consequently, of time-registers, were designed to be employed in different blocks of the proposed time-domain MAC unit. The specifications for the time-registers developed, as well as simulation results, will be shown in this section.

All results presented were obtained considering the simulation of the configuration depicted in Figure 43. Note that by using only one time-register, its output is only proportional to its input (Eq. (3.1)). However, as previously stated, by using two of them it is possible to recover the input signal (Eq. (3.2)).

The first time-register was designed by employing standard gated delay cells. Each one of them is able to generate 1.25 ns of delay, occupying 5.5 µm^2. In order to produce larger delay, increasing the capacity of the Adder & Accumulator block, the time-registers designed by using this cell contains sixteen of them, which means that they would be able to store 20 ns. However, between 0 ns and 1 ns, the circuit may not work properly due to very narrow pulses. Thus, the time-register containing such cells works very well from 1 ns

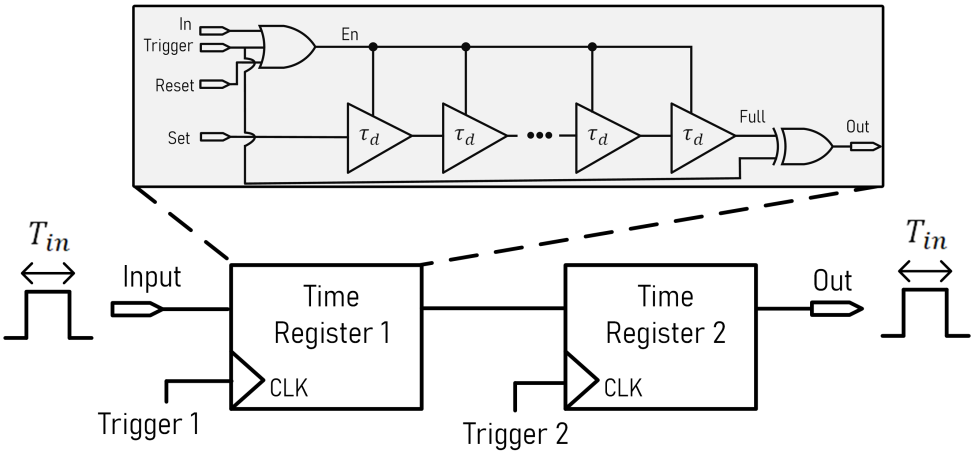


Figure 43 – Cascaded Time-registers configuration simulated

Table 2 – Standard gated delay cell specification (Refer to Figure 22)

Component	Specification (W/L) [m] or capacitance [F]
M1	$1.2\,\mu/180\,n$
M2	$600\,n/180\,n$
M3	$16\,\mu/180\,n$
M4	$8\,\mu/180\,n$
M5,M6	$1.8\,\mu/180\,n$
M7,M8	$600\,n/180\,n$
C1	$200\,f$

to 20 ns, i.e., 19 ns of dynamic range. The specifications of the standard gated delay cell are shown in Table 2.

It is worth noting that, as mentioned earlier, for the standard gated delay cell topology the capacitor is not essential. However, a 200fF capacitor was employed in the design because it allows the leakage current to be reduced. Thus it is possible to generate more delay while reducing the error. Using capacitances larger than this would not be interesting, as it would considerably increase the area occupied by the circuit and would bring negligible benefits. Capacitances smaller than the one used would not be interesting either, since they barely increase the delay and reduce the error of the circuit.

Simulations were performed for the cascaded time-registers designed with standard gated delay cells. First, in Figure 44 a timing diagram containing the behavior and the control signals for both time-registers is shown. This simulation was carried out considering an input signal of 15 ns for demonstration purposes.

The circuit operates exactly as expected and explained in Section 3.2.1. The

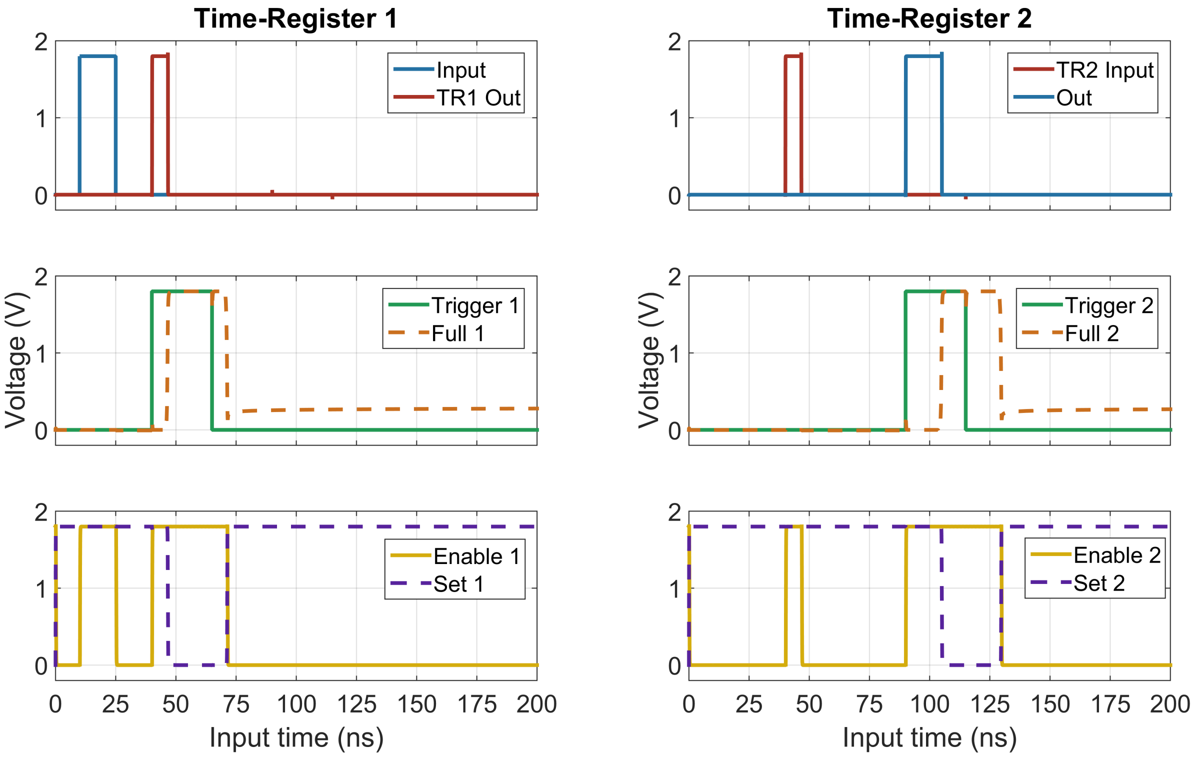


Figure 44 – Time-register with standard gated delay cells timing diagram

output of each time-register is given by the difference between the rising edges of the trigger and the full flag. So if a larger signal is to be stored, the time-register will reach its full capacity sooner when compared to a narrower input signal. This can perfectly be seen in Figure 44, since in Time-Register 1 the input signal has a width of 15 ns, which means that its output is roughly 5 ns ($T_{TR1\,Out} = T_{FS} - T_{Input} = 20\,ns - 15\,ns = 5\,ns$). In Time-Register 2, on the other hand, the input signal is the output of Time-register 1. Its output, which is also the final output of the cascaded time-registers, is approximately the input, 15 ns ($T_{Out} = T_{FS} - T_{TR2\,Input} = 20\,ns - 5\,ns = 15\,ns$). It is also worth noting that when each time-register reaches its maximum capacity, the Set signal goes to zero, which means the circuit is being reset. The full flag is deactivated as soon as this process is completed, indicating that the circuit is ready to operate again. For storing and recovering a time signal the circuit consumes approximately 213 µW.

There is undoubtedly some error associated with the storage of information. The error is defined as the difference between the expected output, which ideally is the input, and the true output obtained after simulation. In Figure 45 the absolute and relative errors are shown, for the circuits' entire operational range.

An interesting characteristic of the time-register is that the absolute error has a periodic behavior. This is because the same holding state recurs every delay cell (for example, if $T1 = 2.4\tau_d$ and $T1 = 5.4\tau_d$, holding occurs in different delay cells but their states are the same). As a result, due to the absolute error pattern, the relative error tends

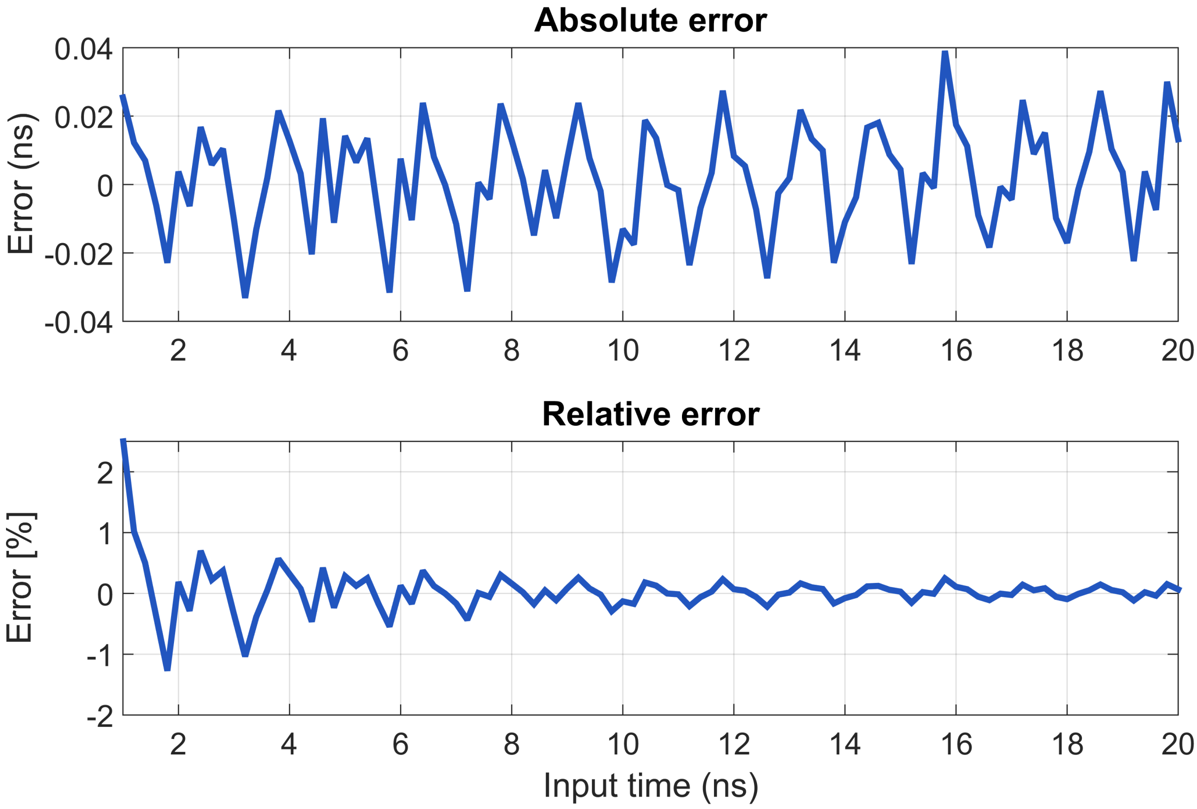


Figure 45 – Time-register with standard gated delay cells simulated error

to decay over the operational range. The maximum error obtained was 39 ps, which is very low, considering the magnitude of the signals the circuit is subjected to. However, in the multiplier operation, the signal is stored and released several times due to successive additions, and the cumulative error tends to increase. To address this issue, new time-registers were designed by employing skewed gated delay cells, a topology known for its low error rate.

The time-registers designed with skewed gated delay cells are the ones used in the Multiplier block. As previously stated, its great advantage is to provide less error, however, its delay cells are capable of generating less delay. Each skewed gated delay cell occupies a silicon area of 43 µm^2, which is almost 9 times more than the area occupied by standard delay cells, and generates roughly 0.55 ns of delay. The time-register designed with those cells is able to store up to 8.7 ns, since sixteen of them were serially connected. The specifications of the skewed gated delay cell are shown in Table 3.

The same simulations were performed for the time-registers designed with skewed gated delay cells. As can be seen in Figure 46, the operation principle is identical to that of time-registers built with standard delay cells. However, the circuit consumes 907 µW, which is 4.3 times more when compared to the ones designed with standard gated delay cells. Also, there are significant differences between the two topologies in terms of error and dynamic range.

Table 3 – Skewed gated delay cell specification (Refer to Figure 23)

Component	**Specification (W/L) [m]**
M1,M3,M5	$16\,\mu/600\,n$
M2,M4,M6	$8\,\mu/600\,n$
M7	$1.8\,\mu/180\,n$
M8	$600\,n/180\,n$

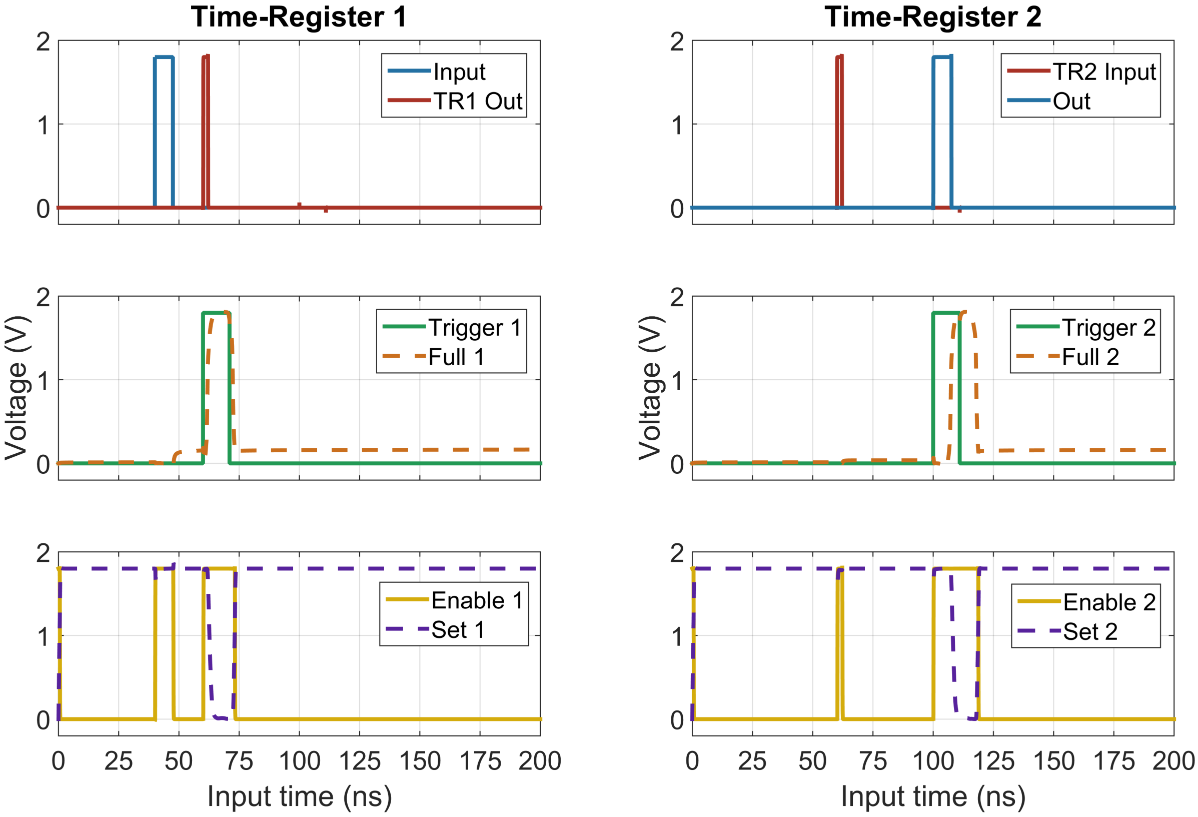


Figure 46 – Time-register with skewed gated delay cells timing diagram

The simulated absolute and relative errors are shown in Figure 47. As it can be see, the maximum error obtained when employing skewed gated delay cells is 24 ps, which is nearly half the error presented by the circuit with standard ones. Also, the error behavior is less sinusoidal.

After presenting the simulation results it is clear that, regardless of the delay cell topology employed, the time-registers can store a time signal and return it when requested. Furthermore, there is a trade-off in relation to some circuit metrics which makes using one delay cell over another more appealing depending on the situation. Time-registers built with standard gated delay cells take up less space, consume less power, and have a wider dynamic range. Time-registers with skewed gated delay cells, on the other hand, operate with less error.

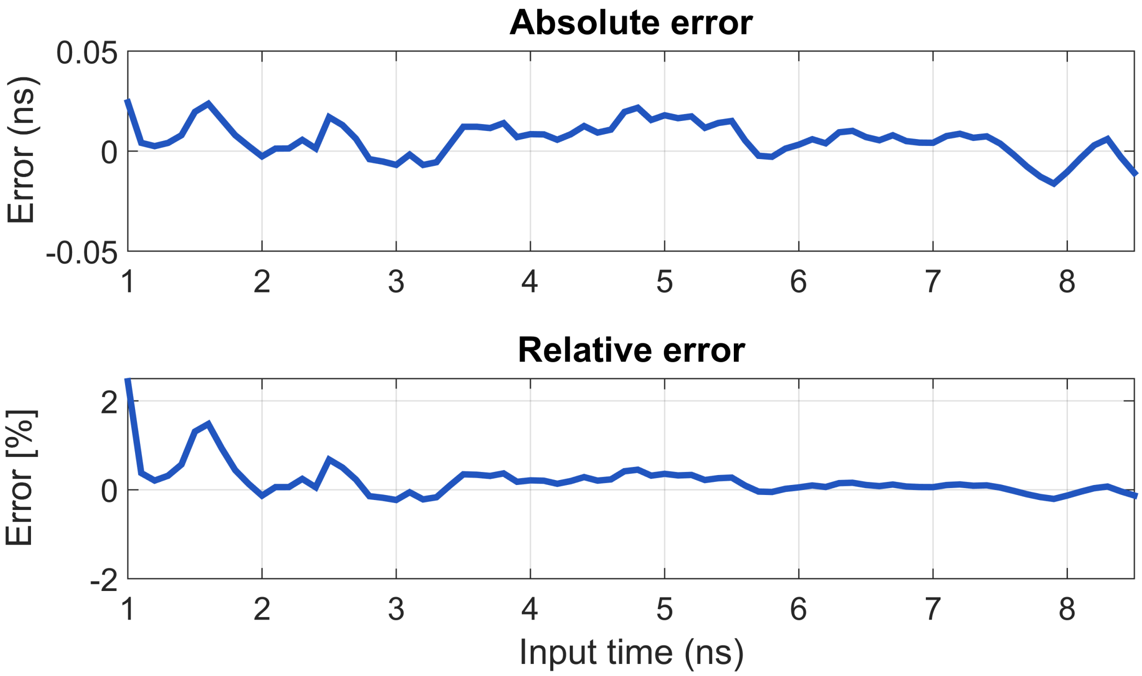


Figure 47 – Time-register with skewed gated delay cells simulated error

4.2 Time-domain multiplier

The designed time-domain multiplier operates fully automatically, being able to perform multiple operations successively. The two time signals to be multiplied, T1 and T2, as well as the trigger signals that serve as clock for the time-registers, must be provided as inputs to this circuit. The output is a time signal proportional to the product of T1 and T2.

The multiplier was designed to perform multiplications up to times four, for T1 ranging from 1 ns up to 8.7 ns, which is enough to demonstrate the proposed approach. As previously stated, the multiplier is obtained by discretizing the second input time T2, which has a range of 3 ns to 23.5 ns. The corresponding time-to-digital conversion is shown in Table 4, where each range of T2 values corresponds to a specific multiplication. This range of values and their respective equivalences are determined by the GRO frequency, since each oscillation corresponds to a new successive addition that must be performed. Hence, for the duration of the T2 pulse, after every oscillation, the multiplicand is incremented by one, and when the pulse is over, the multiplicand is set as the total number of oscillations (e.g., to multiply by two, the pulsewidth of T2 must be between 6.5 and 12.1 ns, which results in two GRO oscillations). This circuit was designed in such a way as to allow the multiplicand T1 and the multiplier T2 to have the same order of magnitude. Naturally, when entering T2 with wider pulse widths i.e., greater than 23.5 ns, multiplication by higher integers is a possibility and can be accomplished by doing small modifications in the circuit. Simply designing a counter with more bits and adjusting the interval between the multiplicand storage (Figure 32) and the final addition (Figure 34) would be sufficient

Table 4 – Input time T2 digital equivalency

Input time T2 [ns]	Corresponding operation
3 - 6.4	x1
6.5 - 12.1	x2
12.2 - 18	x3
18.1 - 23.5	x4

for this. To enter with a higher T1 and increase the range of operation, the solution is to increase the amount of delay cells in time-registers. However, as previously mentioned, the multiplier was constrained to this operational range for proof-of-concept purposes. The designed time-domain multiplier is able to perform 1.9 Moperations/s.

To demonstrate the time-domain multiplier operation Figure 48 shows the circuit's most important signals, considering the input signals T1 = 2 ns and T2 = 20 ns. This T2 pulse width corresponds to a multiplication by four, so in the circuit's final output a signal around 8 ns is to be expected.

There is undoubtedly some error associated with the time-domain multiplier. The error is defined as the difference between the expected output and the true output obtained after simulation. Because more successive additions are performed, the absolute error tends to increase when performing multiplications by greater numbers. However, the relative error remains roughly constant. The absolute error is defined as the difference between the output time obtained in simulation and the expected output, which is the input time multiplied by the corresponding operation:

$$Abs_{error} = simulated\ T_{out} - expected\ T_{out} \tag{4.1}$$

However, more meaningful than the absolute error, is the relative error, defined by Eq. (4.2):

$$Rel_{error}[\%] = \frac{Abs_{error}}{expected\ T_{out}} \times 100\% \tag{4.2}$$

To illustrate this behavior, Figure 49 shows the relative error associated with the time-domain multiplier, for various simulations of the multiplication operation. As can be seen, the error is less than 3% across the entire operational range of the circuit.

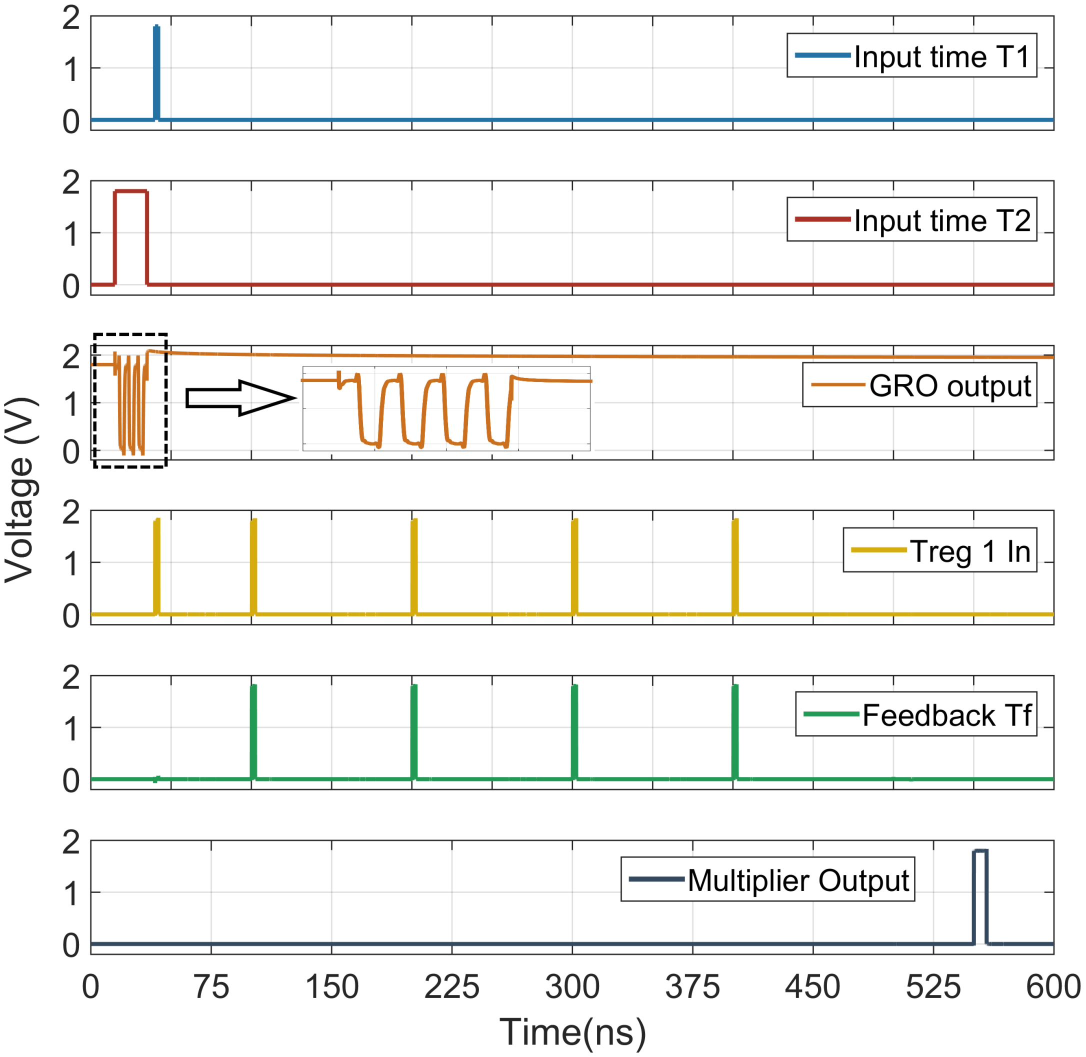


Figure 48 – Multiplier simulation: Operation and key signals (Refer to Figure 31)

4.3 Time-domain MAC unit

After demonstrating the simulation results and the specifications of all the subcircuits that comprise the time-domain MAC unit, this section addresses several relevant results obtained for the proposed circuit. The operation of the MAC unit and its performance metrics, as well as process corners and temperature simulations, to study the circuit under different conditions, will be covered. Finally, conventional architectures of digital MAC units were designed to assist in determining the benefits and drawbacks of performing multiply-accumulate operations in the time-domain.

4.3.1 Operation and Performance Metrics

The time-domain MAC unit, like the multiplier, requires only two input times and trigger signals to function properly. The multiplication, addition, and accumulation operations are completely automatic, and the circuit can run indefinitely, with each

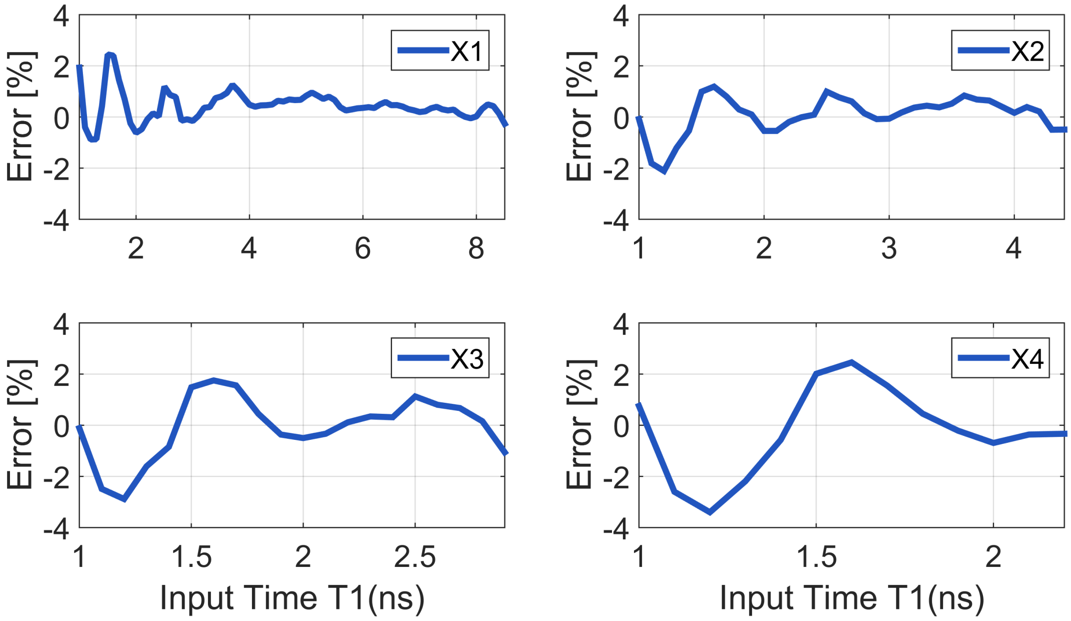


Figure 49 – Simulated relative error associated with multiplication by one, two, three and four

cycle representing one MAC operation. After all MAC operations have been completed, the circuit releases the final result as the output. The partial results are also shown in the circuit's output, after the end of every cycle. To facilitate understanding, Figure 50 illustrates how the operational flow is done in the simulated circuit.

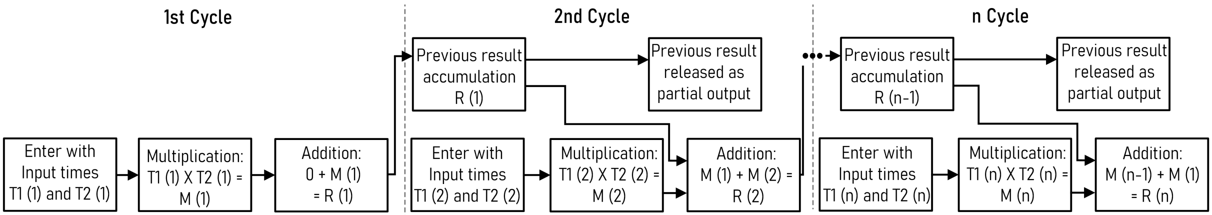

1st Cycle
2nd Cycle
n Cycle
Enter with Input times T1 (1) and T2 (1)
Multiplication: T1 (1) X T2 (1) = M (1)
Addition: 0 + M (1) = R (1)
Previous result accumulation R (1)
Previous result released as partial output
Enter with Input times T1 (2) and T2 (2)
Multiplication: T1 (2) X T2 (2) = M (2)
Addition: M (1) + M (2) = R (2)
Previous result accumulation R (n-1)
Previous result released as partial output
Enter with Input times T1 (n) and T2 (n)
Multiplication: T1 (n) X T2 (n) = M (n)
Addition: M (n-1) + M (1) = R (n)

Figure 50 – MAC unit operational flow

To demonstrate the time-domain MAC unit operation Figure 51 shows the circuit's most important signals, considering the input signals T1 = 2.5 ns and T2 = 15 ns. This T2 pulse width corresponds to a multiplication by three. These signals are fed to the MAC unit twice, resulting in two multiply-accumulate operations or two complete cycles performed by the circuit. After the first cycle the MAC final output is roughly 7.5 ns, since T1 was successively added three times. After the second and final cycle, the output is approximately 15 ns ($TI \times 3 + T1 \times 3 = T1 \times 6$). Note that the result of a previous cycle is displayed in the output of the circuit only during the next cycle. This is because it must be stored and can only be released when it is added to the multiplication result of the next cycle. This behavior is very similar to that observed in digital MAC units, since the result tends to be one clock cycle delayed.

The proposed MAC unit achieves input dynamic range of 19 ns, operating for

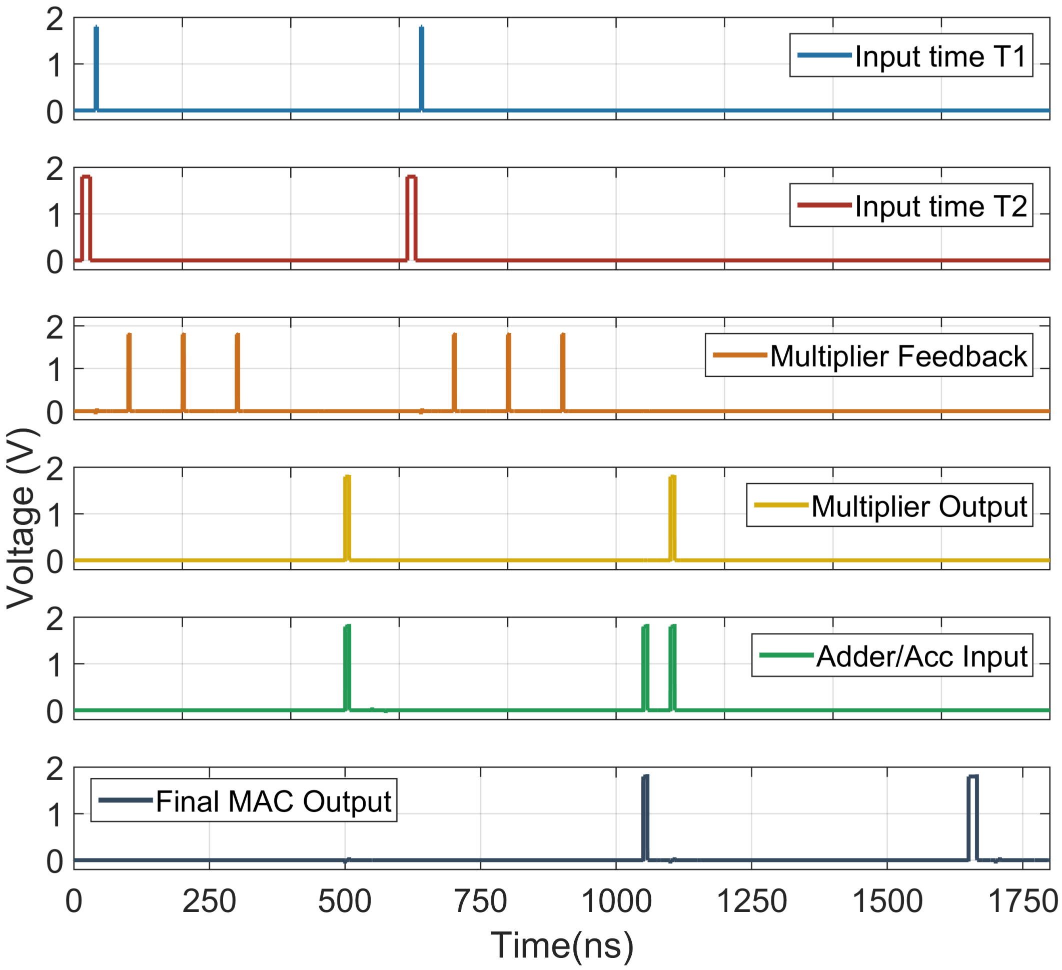


Figure 51 – MAC unit simulation: Operation and key signals (refer to Figure 35)

times ranging from 1 ns up to 20 ns. The circuit average power consumption tends to increase for higher multiplications and in the worst case is 1.72 mW. The total area occupied by the MAC unit is 3167 µm^2. The time-register with skewed gated delay cells is the sub-block that consumes more power and occupies larger area, contributing with roughly 80% for both parameters. The circuit is able to perform 1.67 Moperations/s.

Linearity is another MAC unit metric of interest and has a direct relationship with circuit error. The higher the error, the lower the linearity tends to be, since the error has oscillatory behavior. To evaluate the error and determine the linearity of the circuit, T2 was assigned a time signal in the lower operating range, corresponding to a multiplication by one, and a sweep for various T1 pulse widths (input time T1) was performed, considering a single MAC operation. It is important to remember that the MAC unit range is wider than the multiplier range, since the adder and accumulator block was designed with standard gated delay cells. As a result, for this simulation, the multiplier's maximum capacity was assumed, because only one multiply-accumulate operation is being performed. As shown in Figure 52, the time-domain MAC unit has a very linear behaviour, with the simulated results being quite close to the expected. The simulated coefficient of determination, R^2, is over 0.99 for all the MAC unit operational range.

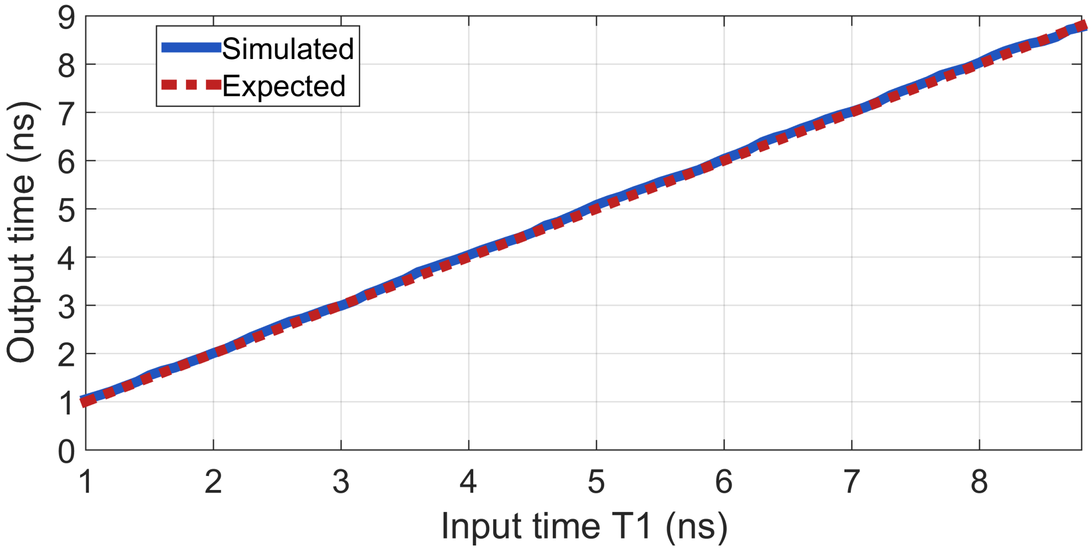


Figure 52 – MAC unit simulation: Linearity

The absolute and relative error are presented in Figure 53. An interesting feature of the circuit is that the absolute error has a sinusoidal behavior. This is because of the time-registers, as the same holding state recurs every delay cell (for example, if $T1 = 2.4\tau_d$ and $T1 = 5.4\tau_d$, holding occurs in different delay cells but their states are the same). As a result, due to the absolute error pattern, the relative error tends to decay over the operational range. It is worth noting that the time-registers account for the majority of the error. To reduce the relative error it is possible to apply an offset signal, causing the absolute error to oscillate around 0 ns, instead of 0.04 ns. Another possibility would be to design the time-domain MAC unit by using newer technologies, with smaller technology nodes, in order to reduce the absolute and, therefore, the relative error.

For the sake of clarity, the reproduction of all combinations of operations in the entire operating range is not shown in this section. Nevertheless, it is important to highlight that the simulated error is not worsened for different operations. The maximum simulated error for different multiplication operations is shown in Figure 54 for a single multiply-accumulate cycle, taking into account the circuits' entire dynamic range. As can be seen, the maximum error occurs when multiplying by one, and it is less than 5%.

A performance summary for the designed time-domain MAC unit is shown in Table 5.

4.3.2 PVT simulations

In order to have an estimate of the impact of fabrication process, temperature and supply voltage on MAC unit operation, simulations were performed. Again, for all simulations along this section, T2 was assigned a time signal in the lower operating range,

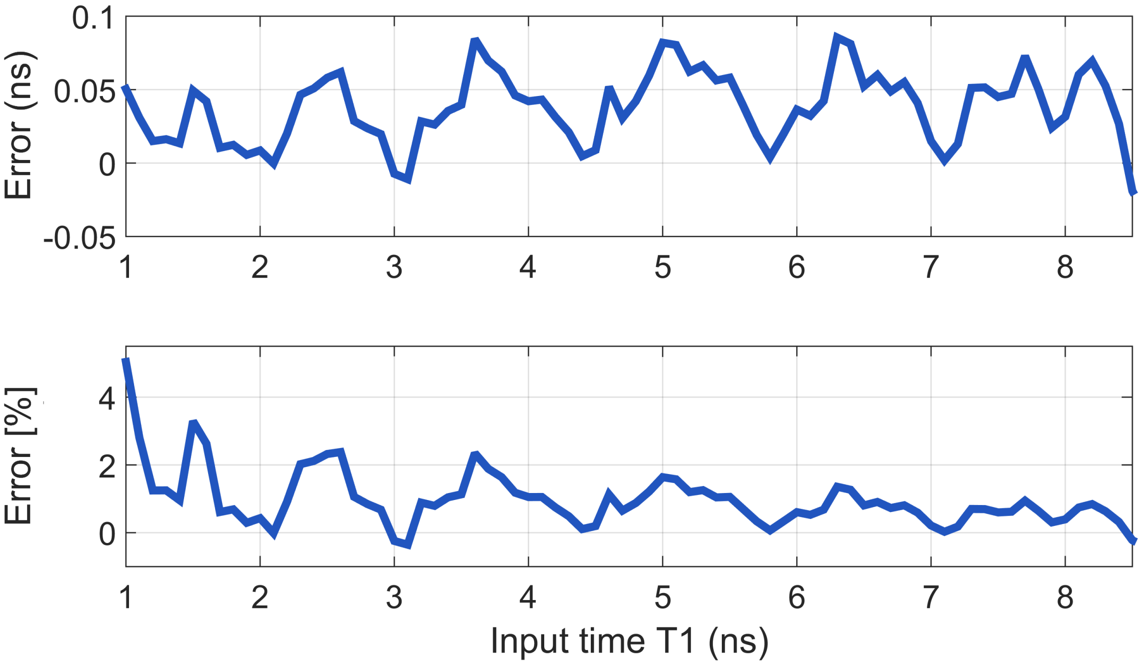


Figure 53 – MAC unit simulation: Absolute and relative error

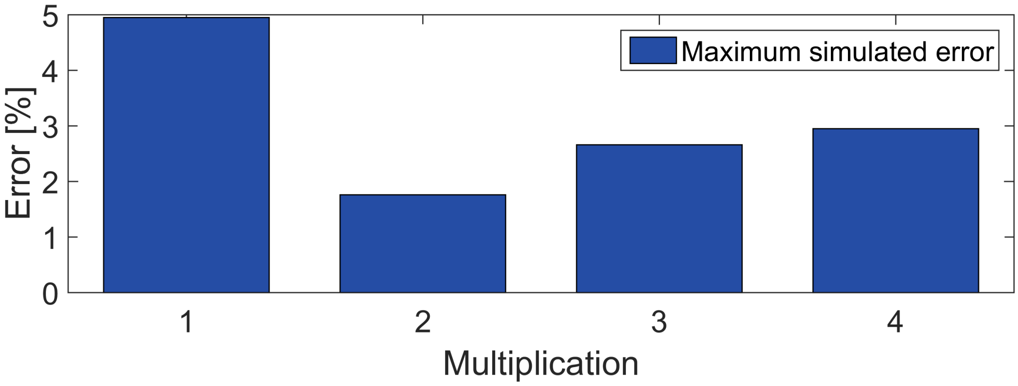


Figure 54 – Maximum simulated error obtained for various multiplications in a single MAC operation

and a sweep for various T1 pulse widths (input time) was performed, considering a single MAC operation. As it will be discussed below, although the MAC handles fabrication process, temperature and supply voltage variations well, its linear operating range tends to shift a bit according to the corner considered. Consequently, minor changes in the DR may occur.

The simulation of the MAC unit under process corners variation is shown in Figure 55. The circuit was simulated for the five possible corners: typical-typical (TT), fast-fast (FF), slow-slow (SS), fast-slow (FS), and slow-fast (SF). As can be seen, all process corners have similar magnitude errors. However, the error for the FF corner is generally smaller, remaining below 1% for almost the entire operational range. The SS corner, on the other hand, has the highest error, staying above 1% for larger T1 pulse widths, indicating a slightly higher absolute error when compared to the others. It is also

Table 5 – Time-domain MAC unit performance summary

Technology	180 nm
Supply voltage	1.8 V
Throughput	1.67 Moperations/s
Linear dynamic range	19 ns
Linearity by R^2	0.99
Power	1.72 mW
Area	3167 µm^2
Number of gates	193

worth noting that some corners, such as TT and SS, allow for larger input pulse widths, whereas others have a narrower upper limit. The FF corner has the shortest operational range, with the multiplier block only capable of processing input times of up to 7.5 ns. This is to be expected, as the delay cells are faster, resulting in less delay and a reduced working range for the time-registers.

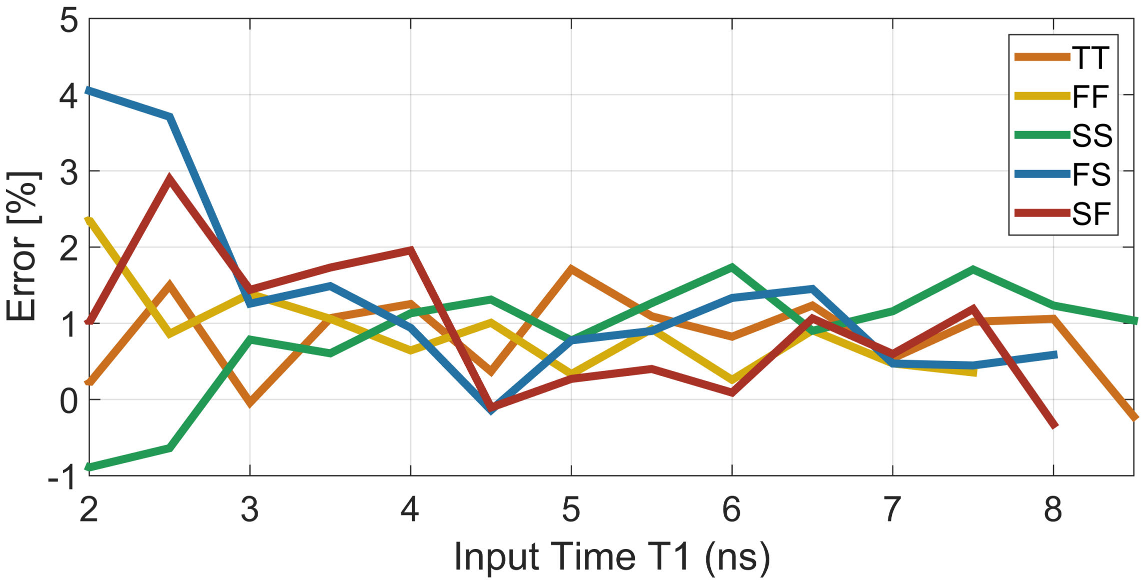


Figure 55 – MAC unit simulation: Process corners

Simulations of the time-domain MAC unit under temperature variation, in the commercial temperature range (0° to 80 °C), were also carried out and can be seen in Figure 56. The circuit works properly across the commercial temperature range. One might observe that the higher the temperature the higher the error tends to be and vice versa (i.e. lower temperatures, lower error). However, the dynamic range is reduced for lower temperatures because of the same reasons as for corner FF.

Finally, the MAC unit was simulated considering a ±10% supply variation as

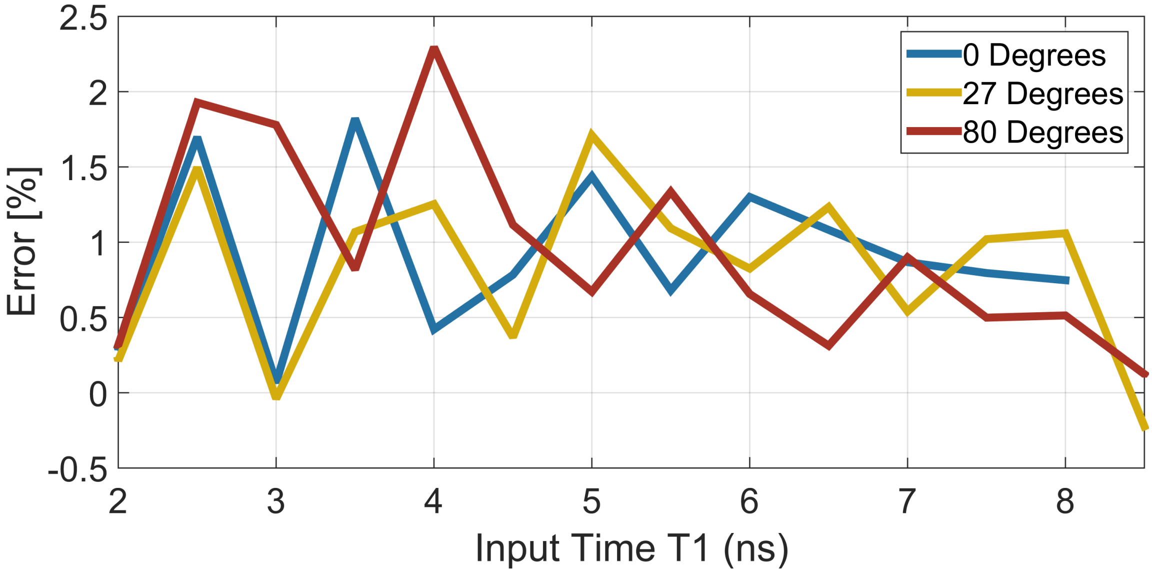


Figure 56 – MAC unit simulation: Temperature variation

shown in Figure 57. In both cases, the circuit can still perform MAC operations without experiencing significant performance degradation. It is worth noting that the circuit tends to present higher errors when employing a lower power supply and smaller ones for higher voltages. Moreover, at lower voltages, the dynamic range slightly increases because the time-registers are able to generate more delay. However, as V_{DD} increases, it tends to decrease since the circuit is faster.

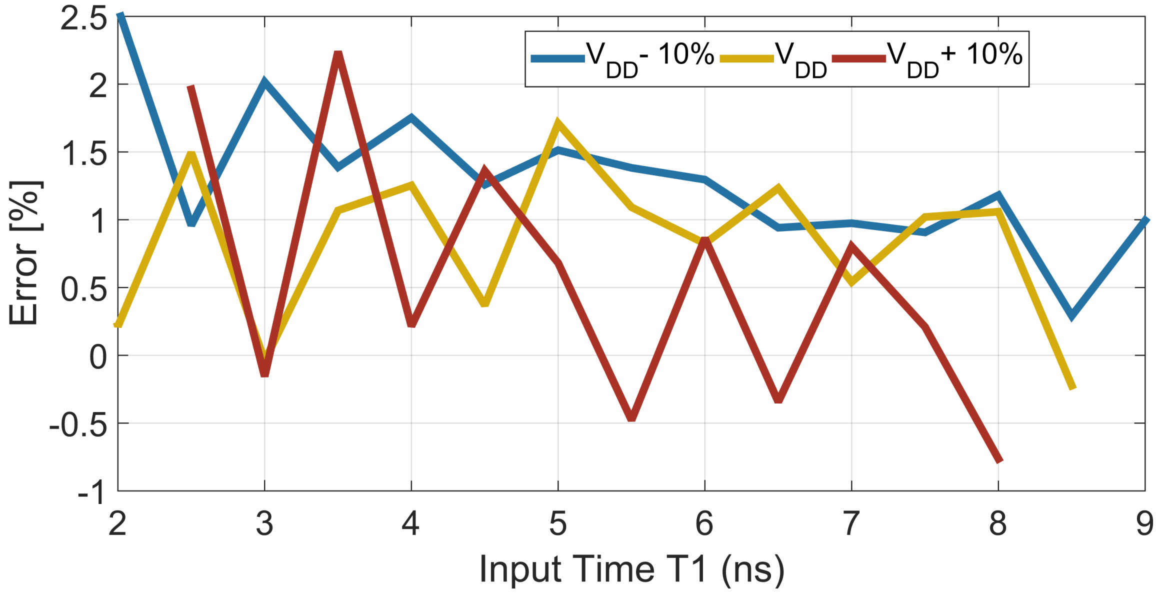


Figure 57 – MAC unit simulation: Supply variation.

4.3.3 Dynamic range impact

A critical point in circuit design is related to the scalability of parameters such as area, power and speed, that is, how these parameters change as the dimensions of the

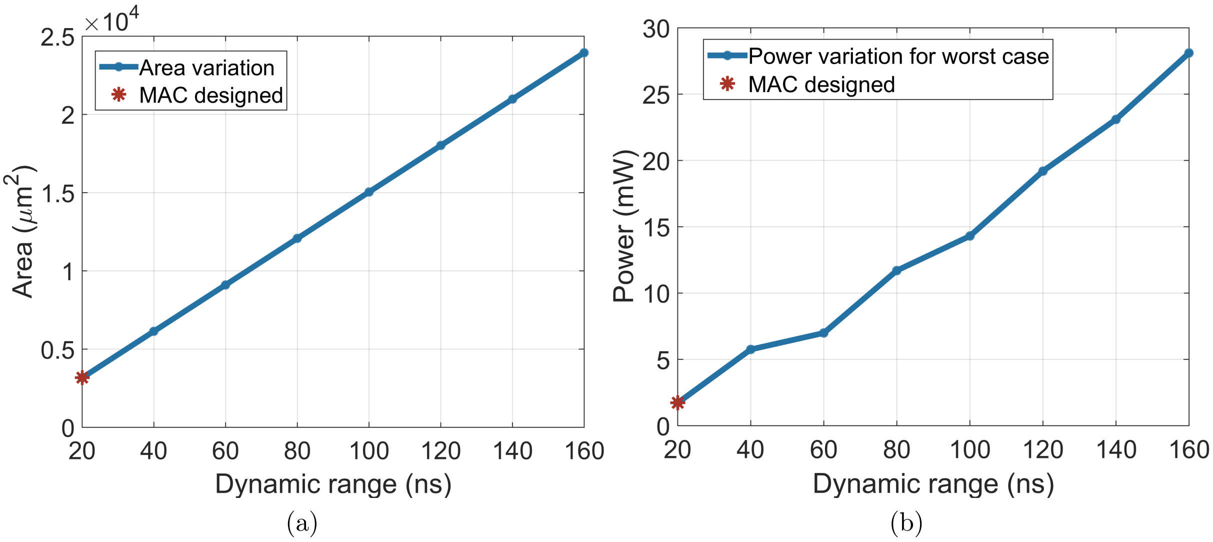


Figure 58 – Comparison of simulation results of increasing time-domain MAC unit dynamic range. (a) Area occupied. (b) Power consumption

device increase. In this subsection, it will be discussed how variations in the dynamic range affects the performance of the proposed MAC unit. Due to the fact that information in the time-domain is contained in time pulses, it is evident that depending on the application, a wider or a narrower dynamic range may be required. Increasing the dynamic range, for example, may be beneficial because it allows for longer and larger operations, but it increases the area occupied and power consumed by the MAC unit, since the time-registers must be designed with more delay cells. When working with narrower dynamic ranges, however, it is possible to gain speed by performing more operations in a given time because the information must be transmitted by using signals with narrower pulse width.

To investigate the above trade-off, the time-domain MAC unit was redesigned for different dynamic ranges. The results obtained can be seen in Figure 58. As shown in Figure 58a, the estimated schematic area of the time-domain MAC unit increases linearly with the dynamic range. This is due to the fact that in the proposed topology, the dynamic range is primarily related to the number of delay cells, which is also the component that occupies the most area. Thus, for instance, to double the dynamic range it is only necessary to double the number of delay cells in the time-registers.

Following the same logic, power consumption was also estimated for the redesigned time-domain MAC units. When increasing the dynamic range the proposed circuit also consumes more power, as shown in Figure 58b. It tends to increase almost linearly with dynamic range.

As a result, it is clear that the proposed MAC unit topology can be easily redesigned to meet the needs of various applications, without causing the power consumption and occupied area to change dramatically to the point that using the circuit becomes unfeasible.

Another trade-off to consider when increasing the dynamic range is the speed of each operation. As expected, increasing the dynamic range increases the time taken for performing a multiply-accumulate operation proportionally. This is because all of the operations in the circuit are performed with time signals. As a result, if the MAC unit is designed to work with signals of greater magnitude, the operations take longer to complete, lowering the circuit's throughput.

4.3.4 Digital MAC units

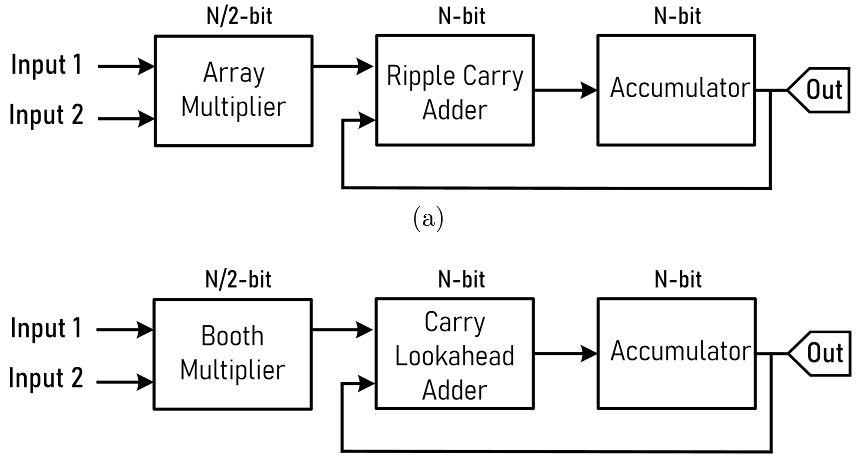
Since the proposed fully time-domain MAC unit is the first one in literature to the best of authors´ knowledge, it is not possible to compare it to similar ones. However, this section presents a comparison of our approach to the traditional digital MAC units, not to make a one-a-one comparison, but to highlight its advantages while also addressing its limitations.

Two distinct MAC units were designed in VHDL hardware description language and are available to the readers through the link [85]. These circuits were synthesized using Genus Synthesis Solution from Cadence. The same 180-nm CMOS technology used for the time-domain MAC unit was employed. The first MAC unit consists of an array multiplier and a ripple carry adder (RCA), which are one of the most simple designs for those circuits. This MAC unit, whose block diagram is shown in Figure 59a consumes less power and area, however, due to its simplicity, it performs poorly and cannot multiply negative numbers. Both sub-circuits can be designed by using half-adders and full-adders and some AND gates, for partial product generation. The second MAC unit designed, on the other hand, is a bit more complex and optimized in terms of performance, containing a Booth multiplier and a carry lookahead adder (CLA) as shown in 59b. However, to have better performance it tends to consume more power and area [86, 87]. The digital adders and multipliers architectures employed in the digital MAC units were presented in detail in Sections 2.2 and 2.3.

First, both MAC units were simulated and they are operating just as expected. The simulations are shown in Figure 60 and were done in Intel Quartus Prime.

To assess the silicon area and power consumption of the digital MAC units, simulations were performed for both architectures, and these parameters were extracted while taking into account various bit numbers. The goal is to demonstrate how they change as the number of bits increases, allowing the reader to contrast the performance of the proposed time-domain MAC with the performance of digital ones. The same operating frequency presented by the time-domain MAC unit was used for the simulation of the designed digital MACs.

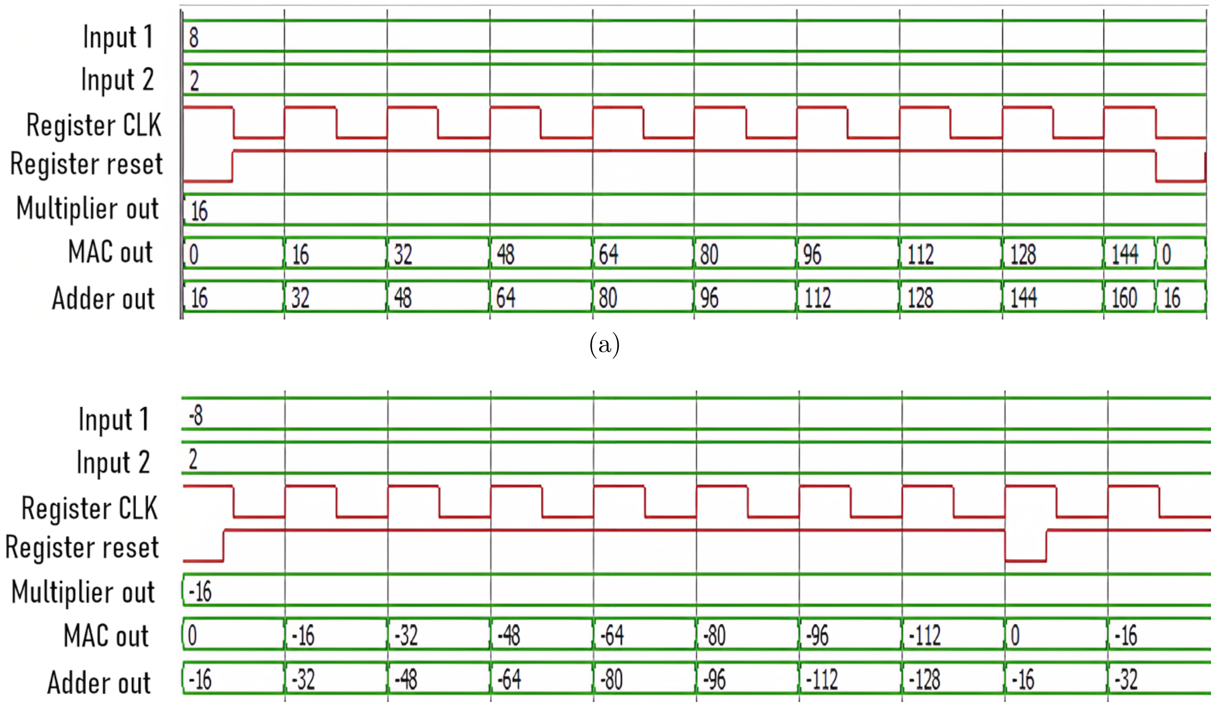
Both digital MAC units were simulated and their respective area for various bits



(a)

(b)

Figure 59 – Block diagram of proposed digital MAC units. (a) Array Multiplier + RCA. (b) Booth Multiplier + CLA [86] [87].

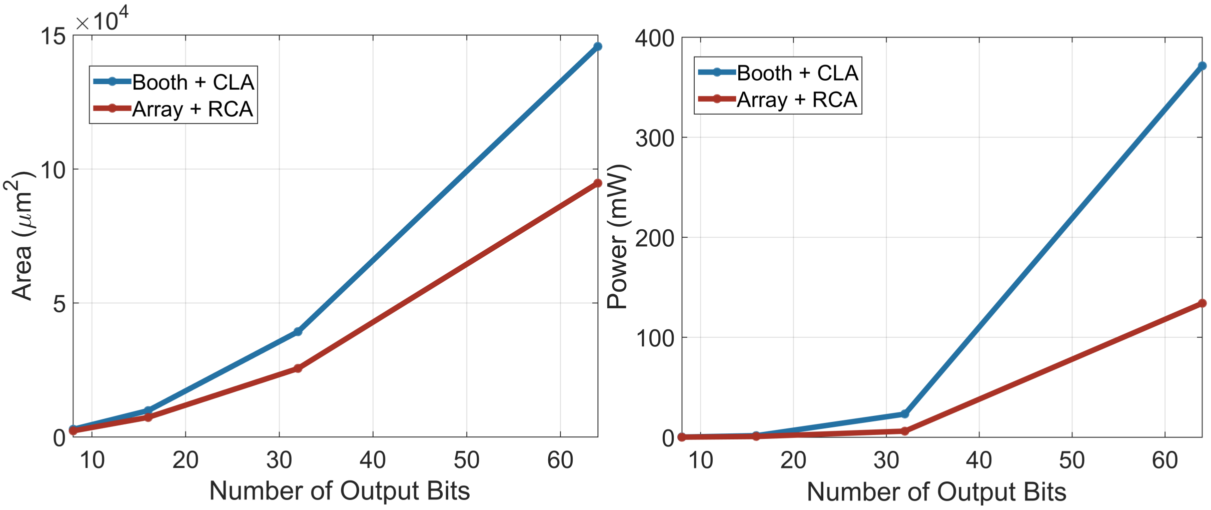


(a)

(b)

Figure 60 – Digital MAC units simulation. (a) Array Multiplier + RCA. (b) Booth Multiplier + CLA.

is show in Figure 61a. The area occupied when using a booth multiplier and a CLA is higher, as expected. Considering the 64-bit simulation (two 32-bit inputs, 64-bits output), this MAC unit occupies 0.145 mm^2 , whereas the MAC with array multiplier and RCA takes up 0.095 mm^2. This is 3.7 times greater than the area occupied by the same MACs designed for 32-bit. Also, it is interesting to note that the area occupied by digital MAC units increases significantly when the number of bits increases, in an almost exponential relationship.



(a) (b)

Figure 61 – Comparison of results obtained with Genus Synthesis Solution estimates. (a) Area occupied. (b) Power consumption.

Simulations aimed at estimating the power consumption were also carried out. In digital MAC units power consumption increases, even more steeply than area, for greater number of bits, as shown in Figure 61b. 64-bits conventional digital MACs (two 32-bits inputs, 64-bits output), for example, can consume up to 371 mW which is 1500% higher than the power consumed by 32-bits ones.

The time to execute an operation is another parameter of interest. Digital MAC units synthesized in the same technology node typically have a much smaller delay when compared to the proposed time-domain solution, ranging from a few to dozens of nanoseconds, depending on the architecture and the number of bits [88, 89]. As a result, due of the way the arithmetic operations are performed in the designed time-domain MAC unit, it is safe to conclude that it will always be slower than digital ones, at least for the technological node used in this work. To perform faster operations in the time-domain, a possible solution would be working with narrower signals, which is possible by using smaller technology nodes.

4.3.5 Overview and comparison

As previously stated, because of the novelty of the MAC unit designed, it is not straightforward to fairly compare the time-domain circuit with other domain solutions. Notwithstanding, Table 6 compares this work to MAC units developed both in analog and digital domains that were designed for applications similar to those intended by this work.

Performance comparison with recent works reveals that even by designing the MAC unit with an older technology and with a higher supply voltage, the proposed circuit tends to occupy less area than analog and digital solutions while exhibiting energy

Table 6 – Performance summary and comparison with recent works

References	This work	JETCAS'18 [90]	JSSC'19 [91]	BioCAS'19 [92]	IEETC'20 [55]	ISCAS'19 [93]	ICEIC'19 [94]	ICETET'09 [56]
Technology	180 nm CMOS	65 nm CMOS	55 nm CMOS	Submicron Pseudo-CMOS	40 nm LP CMOS	28 nm FD-SOI CMOS	28 nm FD-SOI CMOS	180 nm CMOS
Domain	Fully Time	Time (back-gate-driven)	Time mixed-signal	Analog Current – Time encoding	Analog/Digital Hybrid	Digital	Digital	Digital
Voltage Supply (V)	1.8	0.7	0.4 - 1	-	1	1 - 1.8	1	1.8
Active area (mm^2)	0.0031	0.04	3.4	-	0.04	0.0035	0.0029	3.15
Speed (MACs/s)	1.67 M	21.6 G	-	6.667 k	7.55 M	769 M	3.7 G	83.3 M
Power (W)	1.72 m	-	690 µ	136.6 µ	1.71 m	2.48 m	2.25 m	50.26 m
FoM (ops^{-1}/W)	1.94 G	9 T	3.12 T	2.343 G	8.83 G	0.62 T	3.28 T	3.31 G
Energy/MAC (J/MAC)	1.03 n	0.042 p	-	20.64 n	0.226 n	3.22 p	0.6 p	0.6 n
Number of gates	193	-	-	-	-	7360	6130	3200

consumption of the same order of magnitude. It is also important to emphasize that the proposed circuit needs far fewer gates than all works, which is another major advantage and a great indication that it is very efficient in terms of area and power. However, as explained throughout this section, the main disadvantage of the proposed work is in terms of speed, since operations tends to be performed slower than most solutions found in the literature.

Chapter 5

Conclusion

5.1 Final Remarks

This master thesis presented a fully time-domain MAC unit. The circuit is capable of consecutively multiplying two input time pulses and add them to previously stored. Not only that, but also a time-domain multiplier was designed, which is an extremely important circuit when it comes to signal processing. Since the development of time-domain circuits is an emerging research field, several subsequent projects are enabled by this master thesis and the blocks designed in this work can be employed in a wide range of applications. More than just a new circuit design, this work presented a new concept for performing multiply-accumulate operations. This preliminary investigation has already shown that utilizing the time-domain to perform addition, multiplication and accumulation is feasible and can be very beneficial, especially in low-power and low-area applications.

While the proposed MAC unit is a novel approach for performing multiply-accumulate operations in the time-domain, there exists limitation and room for improvement. First, it should be noted that the time-domain MAC unit performs poorly when compared to digital ones in terms of operations per second. This is primarily due to the sequential (rather than parallel) nature of the designed circuit, which means that the order of magnitude of the signals being handled in the circuit is directly associated with its speed. As the circuit operates with signals in the range of nanoseconds, it is expected to have a reduced speed compared to digital circuits. To allow faster operation, an alternative would be to operate with narrower signals, on the order of picoseconds. However, to do so, the circuit must be designed in newer technologies with smaller and faster nodes, thus improving the minimal resolution. Second, by using smaller nodes, the circuit error can be reduced even further. Nevertheless, the simulated errors obtained with the proposed approach are not that high (under 2% for almost the entire operational range) and do not pose a problem, especially for internet of things and machine learning applications. Finally, it should also be noted that parameters such as power consumption an area can also be reduced since the MAC unit was not optimized in terms of those. Because the designed MAC contains mostly digital blocks, lowering the power supply voltage could be one solution to reduce power consumption. Working with narrower input times, as previously mentioned, would also be an option to reduce the area occupied and consume less power.

5.2 Future Works

This master thesis paves the way for many future works of varying complexity and domain areas. The following are some of the possible ramifications of this work:

- Improve the comparison of the proposed MAC unit with other MAC units: As previously stated, because the proposed topology is novel, comparing it to digital MAC units is not straightforward. Since this work's preliminary comparison is superficial and limited it would be interesting to establish a figure of merit that allows a fair assessment in terms of accuracy and throughput.

- Optimization of the proposed circuit: As previously stated the major contribution of this master thesis is in proposing a new concept of MAC unit. So there is still a lot of room for optimization of the circuit in terms of area and power consumption. One possibility would be to reduce the supply voltage, which is currently very high (1.8V). Since in the time-domain the circuits are essentially digital, it would be possible to operate with much lower voltages, which would certainly reduce these parameters.

- Design in different technology nodes: Another possibility is to redesign the time-domain MAC unit in different technology nodes, especially in newer ones. Aside from allowing for lower power consumption and silicon area, using newer process nodes enables the operation with narrower signals. Because the pulse width of the input signals in the circuit is directly related to the throughput, using smaller nodes allows for more operations to be performed in the same time interval. Furthermore, it would be interesting to test the error behavior when performing multiply-accumulate operations with signals of narrower pulse widths.

- Employ the developed circuits in time-domain applications: In this work, only the circuits responsible for performing the operations of addition, multiplication, and accumulation were proposed. One possibility would be to use the time-domain multiplier and/or the MAC unit developed here to perform these operations for a specific application. They could, for example, be used to implement neural network activation functions in the time-domain or to process data from time mode temperature sensors [95].

References

[1] E. P. DeBenedictis, “It’s time to redefine moore’s law again,” *Computer*, vol. 50, no. 2, pp. 72–75, 2017.

[2] H. J. Levinson, “The lithographer’s dilemma: shrinking without breaking the bank,” in *29th European Mask and Lithography Conference*, vol. 8886. SPIE, 2013, p. 888602.

[3] T. Heil, M. Waldow, R. Capelli, H. Schneider, L. Ahmels, F. Tu, J. Schöneberg, and H. Marbach, “Pushing the limits of euv mask repair: addressing sub-10 nm defects with the next generation e-beam-based mask repair tool,” *Journal of Micro/Nanopatterning, Materials, and Metrology*, vol. 20, no. 3, pp. 031 013–031 013, 2021.

[4] K. Asada, T. Nakura, T. Iizuka, and M. Ikeda, “Time-domain approach for analog circuits in deep sub-micron lsi,” *IEICE Electronics Express*, vol. 15, no. 6, pp. 20 182 001–20 182 001, 2018.

[5] O. Panetas-Felouris and S. Vlassis, “A 3rd-order fir filter implementation based on time-mode signal processing,” *Electronics*, vol. 11, no. 6, p. 902, 2022.

[6] F. Yuan, *CMOS current-mode circuits for data communications.* Springer Science & Business Media, 2007.

[7] F. Yuan , *CMOS time-mode circuits and systems: fundamentals and applications.* CRC Press, 2018.

[8] G. W. Roberts and M. Ali-Bakhshian, “Time-domain analog signal processing techniques,” in *Proceedings of the 21st annual symposium on Integrated circuits and system design*, 2008, pp. 7–7.

[9] A. Ameri, “Time-mode reconstruction iir filters for sigma-delta phase modulation applications,” Master’s thesis, McGill University, 2011.

[10] C. Taillefer, “Analog-to-digital conversion via time-mode signal processing,” Ph.D. dissertation, McGill University, 2007.

[11] M. Ali-Bakhshian, “Digital processing of analog information adopting time-mode signal processing,” Ph.D. dissertation, McGill University, 2013.

[12] S. Henzler and S. Henzler, *Time-to-digital converter basics.* Springer, 2010.

[13] O. Akgun and J. Mei, "An energy efficient time-mode digit classification neural network implementation," *Philosophical Transactions of the Royal Society A*, vol. 378, no. 2164, p. 20190163, 2020.

[14] S. Parmar, "Fully digital time domain multiplier," Master's thesis, Dalhousie University, 2017.

[15] R. Krishna, A. K. Mal, and R. Mahapatra, "Cmos time-mode smart temperature sensor using programmable temperature compensation devices and $\Delta\Sigma$ time-to-digital converter," *Analog Integrated Circuits and Signal Processing*, vol. 102, no. 1, pp. 97–109, 2020.

[16] S. U. Rehman, M. M. Khafaji, C. Carta, and F. Ellinger, "A 25-gb/s 270-mw time-to-digital converter-based 8× oversampling input-delayed data-receiver in 45-nm soi cmos," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 11, pp. 3720–3733, 2018.

[17] Z. Gao, J. He, M. Fritz, J. Gong, Y. Shen, Z. Zong, P. Chen, G. Spalink, B. Eitel, K. Yamamoto *et al.*, "A 2.6-to-4.1 ghz fractional-n digital pll based on a time-mode arithmetic unit achieving-249.4 db fom and-59dbc fractional spurs," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 65. IEEE, 2022, pp. 380–382.

[18] S. Uenohara and K. Aihara, "A 18.7 tops/w mixed-signal spiking neural network processor with 8-bit synaptic weight on-chip learning that operates in the continuous-time domain," *IEEE Access*, vol. 10, pp. 48 338–48 348, 2022.

[19] P. S. Locatelli, D. M. Colombo, and K. El-Sankary, "Time-domain multiply–accumulate unit," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 31, no. 6, pp. 762–775, 2023.

[20] M. Masadeh, O. Hasan, and S. Tahar, "Input-conscious approximate multiply-accumulate (mac) unit for energy-efficiency," *IEEE Access*, vol. 7, pp. 147 129–147 142, 2019.

[21] G. Devic, M. France-Pillois, J. Salles, G. Sassatelli, and A. Gamatié, "Highly-adaptive mixed-precision mac unit for smart and low-power edge computing," in *2021 19th IEEE International New Circuits and Systems Conference (NEWCAS).* IEEE, 2021, pp. 1–4.

[22] J. Song, Y. Cho, J.-S. Park, J.-W. Jang, S. Lee, J.-H. Song, J.-G. Lee, and I. Kang, "7.1 an 11.5 tops/w 1024-mac butterfly structure dual-core sparsity-aware neural

processing unit in 8nm flagship mobile soc," in *2019 IEEE International Solid-State Circuits Conference-(ISSCC).* IEEE, 2019, pp. 130–132.

[23] U. Ramadass, J. Ponnian, and V. Kumar, "A new recursive shared segmented split multiply-accumulate unit for high speed digital signal processing applications," in *2016 International Electronics Symposium (IES).* IEEE, 2016, pp. 203–208.

[24] C. Narendra and K. R. Kumar, "Low power mac architecture for dsp applications," in *International Conference on Circuits, Communication, Control and Computing.* IEEE, 2014, pp. 404–407.

[25] P. Khan and R. S. Mishra, "Comparative analysis of different algorithm for design of high-speed multiplier accumulator unit (mac)," *Indian Journal of science and technology*, vol. 9, no. 8, 2016.

[26] A. Navaneetha and K. Bikshalu, "Finfet based comparison analysis of power and delay of adder topologies," *Materials Today: Proceedings*, vol. 46, pp. 3723–3729, 2021.

[27] T. Francis, T. Joseph, and J. K. Antony, "Modified mac unit for low power high speed dsp application using multipler with bypassing technique and optimized adders," in *2013 fourth international conference on computing, communications and networking technologies (ICCCNT).* IEEE, 2013, pp. 1–4.

[28] A. Parameswar, H. Hara, and T. Sakurai, "A swing restored pass-transistor logic-based multiply and accumulate circuit for multimedia applications," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 6, pp. 804–809, 1996.

[29] F. Lu and H. Samueli, "A 200 mhz cmos pipelined multiplier-accumulator using a quasi-domino dynamic full-adder cell design," *IEEE Journal of Solid-State Circuits*, vol. 28, no. 2, pp. 123–132, 1993.

[30] C.-H. Chang, J. Gu, and M. Zhang, "A review of 0.18-μm full adder performances for tree structured arithmetic circuits," *IEEE Transactions on very large scale integration (VLSI) systems*, vol. 13, no. 6, pp. 686–695, 2005.

[31] M. Lapedus, "Big trouble at 3nm," 2021. [Online]. Available: https://semiengineering.com/big-trouble-at-3nm/

[32] P. Girard, "Low power testing of vlsi circuits: Problems and solutions," in *Proceedings IEEE 2000 First International Symposium on Quality Electronic Design (Cat. No. PR00525).* IEEE, 2000, pp. 173–179.

[33] R. Pawar and D. Shriramwar, "Review on multiply-accumulate unit," *Int J Eng Res Appl*, vol. 7, p. 09, 2017.

[34] N. J. Babu and R. Sarma, "A novel low power multiply–accumulate (mac) unit design for fixed point signed numbers," in *Artificial Intelligence and Evolutionary Computations in Engineering Systems.* Springer, 2016, pp. 675–690.

[35] V. Camus, L. Mei, C. Enz, and M. Verhelst, "Review and benchmarking of precision-scalable multiply-accumulate unit architectures for embedded neural-network processing," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 4, pp. 697–711, 2019.

[36] N. Drego, A. Chandrakasan, and D. Boning, "All-digital circuits for measurement of spatial variation in digital circuits," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 3, pp. 640–651, 2010.

[37] R. O. Julio, L. B. Soares, E. Costa, and S. Bampi, "Energy-efficient gaussian filter for image processing using approximate adder circuits," in *2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS).* IEEE, 2015, pp. 450–453.

[38] N. Nair, S. Kaur, and H. Singh, "All-optical ripple carry adder based on soa-mzi configuration at 100 gb/s," *Optik*, vol. 231, p. 166325, 2021.

[39] V. Vijay, M. Sreevani, E. M. Rekha, K. Moses, C. S. Pittala, K. S. Shaik, C. Koteshwaramma, R. J. Sai, and R. R. Vallabhuni, "A review on n-bit ripple-carry adder, carry-select adder and carry-skip adder," *Journal of VLSI circuits and systems*, vol. 4, no. 01, pp. 27–32, 2022.

[40] K. Bagyalakshmi and M. Karpagam, "Performance enhancement of efficient process based on carry-skip adder for iot applications," *Microprocessors and Microsystems*, vol. 76, p. 103101, 2020.

[41] M. Hasan, P. Biswas, M. S. Alam, H. U. Zaman, M. Hossain, and S. Islam, "High speed and ultra low power design of carry-out bit of 4-bit carry look-ahead adder," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT).* IEEE, 2019, pp. 1–5.

[42] M. Hasan, M. S. Islam, and M. R. Ahmed, "Performance improvement of 4-bit static cmos carry look-ahead adder using modified circuits for carry propagate and generate terms," *Science Journal of Circuits, Systems and Signal Processing*, vol. 8, no. 2, pp. 76–81, 2019.

[43] P. Balasubramanian and N. E. Mastorakis, "High-speed and energy-efficient carry look-ahead adder," *Journal of Low Power Electronics and Applications*, vol. 12, no. 3, p. 46, 2022.

[44] M. Hasan, M. S. Hossain, A. H. Siddique, M. Hossain, H. U. Zaman, and S. Islam, "A high-speed 4-bit carry look-ahead architecture as a building block for wide word-length carry-select adder," *Microelectronics Journal*, vol. 109, p. 104992, 2021.

[45] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*. Pearson, 2002.

[46] D. Zoni, A. Galimberti, and W. Fornaciari, "Flexible and scalable fpga-oriented design of multipliers for large binary polynomials," *IEEE Access*, vol. 8, pp. 75 809–75 821, 2020.

[47] Y. Doröz, E. Öztürk, and B. Sunar, "Evaluating the hardware performance of a million-bit multiplier," in *2013 Euromicro Conference on Digital System Design*. IEEE, 2013, pp. 955–962.

[48] B. Baldwin, R. R. Goundar, M. Hamilton, and W. P. Marnane, "Co-ecc scalar multiplications for hardware, software and hardware–software co-design on embedded systems," *Journal of Cryptographic Engineering*, vol. 2, no. 4, pp. 221–240, 2012.

[49] P. Yiannacouras, J. G. Steffan, and J. Rose, "Exploration and customization of fpga-based soft processors," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, pp. 266–277, 2007.

[50] L. D. Pyeatt, *Modern assembly language programming with the ARM processor*. Newnes, 2016.

[51] B. Moons, R. Uytterhoeven, W. Dehaene, and M. Verhelst, "14.5 envision: A 0.26-to-10tops/w subword-parallel dynamic-voltage-accuracy-frequency-scalable convolutional neural network processor in 28nm fdsoi," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2017, pp. 246–247.

[52] M. Chang, S. D. Spetalnick, B. Crafton, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, and A. Raychowdhury, "A 40nm 60.64 tops/w ecc-capable compute-in-memory/digital 2.25 mb/768kb rram/sram system with embedded cortex m3 microprocessor for edge recommendation systems," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 65. IEEE, 2022, pp. 1–3.

[53] C. Uthaya Kumar and S. Kamalraj, "Ambient intelligence architecture of mrpm context based 12-tap further desensitized half band fir filter for eeg signal," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 4, pp. 1459–1466, 2020.

[54] A. C. Mert, E. Karabulut, E. Öztürk, E. Savaş, M. Becchi, and A. Aysu, "A flexible and scalable ntt hardware: Applications from homomorphically encrypted deep learning to post-quantum cryptography," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020, pp. 346–351.

[55] K.-H. Park *et al.*, "Design of analog and digital hybrid mac circuit for artificial neural networks," in *2019 International Conference on Electronics, Information, and Communication (ICEIC).* IEEE, 2019, pp. 1–3.

[56] S. Shanthala, C. Raj, and S. Kulkarni, "Design and vlsi implementation of pipelined multiply accumulate unit," in *2009 Second International Conference on Emerging Trends in Engineering & Technology.* IEEE, 2009, pp. 381–386.

[57] S. Vaidya and D. Dandekar, "Delay-power performance comparison of multipliers in vlsi circuit design," *International Journal of Computer Networks & Communications (IJCNC)*, vol. 2, no. 4, pp. 47–56, 2010.

[58] N. Honarmand, M. R. Javaheri, N. Sedaghati-Mokhtari, and A. Afzali-Kusha, "Power efficient sequential multiplication using pre-computation," in *2006 IEEE International Symposium on Circuits and Systems.* IEEE, 2006, pp. 4–pp.

[59] M. Janveja and V. Niranjan, "High performance wallace tree multiplier using improved adder," *ICTACT journal on Microelectronics*, vol. 3, no. 01, pp. 370–374, 2017.

[60] K. S. Monica, D. Anuradha, S. H. Rasheed, and B. Shereesha, "Vlsi implementation of wallace tree multiplier using ladner-fischer adder," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 1, 2021.

[61] K. Bickerstaff, E. E. Swartzlander, and M. J. Schulte, "Analysis of column compression multipliers," in *Proceedings 15th IEEE Symposium on Computer Arithmetic. ARITH-15 2001.* IEEE, 2001, pp. 33–39.

[62] N. V. V. K. Boppana, J. Kommareddy, and S. Ren, "Low-cost and high-performance 8× 8 booth multiplier," *Circuits, Systems, and Signal Processing*, vol. 38, no. 9, pp. 4357–4368, 2019.

[63] B. Mukherjee and A. Ghosal, "Design and analysis of a low power high-performance gdi based radix 4 multiplier using modified booth wallace algorithm," in *2018 IEEE Electron Devices Kolkata Conference (EDKCON).* IEEE, 2018, pp. 247–251.

[64] A. Inoue, R. Ohe, S. Kashiwakura, S. Mitarai, T. Tsuru, T. Izawa, and G. Goto, "A 4.1 ns compact 54/spl times/54 b multiplier utilizing sign select booth encoders," in *1997 IEEE International Solids-State Circuits Conference. Digest of Technical Papers.* IEEE, 1997, pp. 416–417.

[65] N. H. Weste and D. Harris, *CMOS VLSI design: a circuits and systems perspective.* Pearson Education India, 2015.

[66] N. Kumar, M. Bansal, and N. Kumar, "Vlsi architecture of pipelined booth wallace mac unit," *International Journal of Computer Applications*, vol. 57, no. 11, 2012.

[67] J. Petrzela and R. Sotner, “Binary memory implemented by using variable gain amplifiers with multipliers,” *IEEE Access*, vol. 8, pp. 197 276–197 286, 2020.

[68] G. Zamora-Mejia, A. Diaz-Armendariz, H. Santiago-Ramirez, J. M. Rocha-Perez, C. A. Gracios-Marin, and A. Diaz-Sanchez, “Gate and bulk-driven four-quadrant cmos analog multiplier,” *Circuits, Systems, and Signal Processing*, vol. 38, no. 4, pp. 1547–1560, 2019.

[69] R. Sotner, L. Polak, J. Jerabek, J. Petrzela, and V. Kledrowetz, “Analog multipliers-based double output voltage phase detector for low-frequency demodulation of frequency modulated signals,” *IEEE Access*, vol. 9, pp. 93 062–93 078, 2021.

[70] J. Zhu, Y. Huang, Z. Yang, X. Tang, and T. T. Ye, “Analog implementation of reconfigurable convolutional neural network kernels,” in *2019 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS).* IEEE, 2019, pp. 265–268.

[71] A. Diaz-Sanchez, J. C. Mateus-Ardila, G. Zamora-Mejia, A. Diaz-Armendariz, J. M. Rocha-Perez, and L. A. Moreno-Coria, “A four quadrant high-speed cmos analog multiplier based on the flipped voltage follower cell,” *AEU-International Journal of Electronics and Communications*, vol. 130, p. 153582, 2021.

[72] R. B. dos Santos, G. A. Souza, and L. A. Faria, “A novel four-quadrant/one-quadrant multiplier circuit,” *AEU-International Journal of Electronics and Communications*, vol. 138, p. 153865, 2021.

[73] J. M. Rocha-Perez, G. Zamora-Mejia, A. Diaz-Armendariz, A. I. Bautista-Castillo, A. Diaz-Sanchez, and J. Ramirez-Angulo, “A compact four quadrant cmos analog multiplier,” *AEU-International Journal of Electronics and Communications*, vol. 108, pp. 53–61, 2019.

[74] R. Nägele, J. Finkbeiner, M. Grözing, and M. Berroth, “Design of an energy efficient analog two-quadrant multiplier cell operating in weak inversion,” in *2022 20th IEEE Interregional NEWCAS Conference (NEWCAS).* IEEE, 2022, pp. 5–9.

[75] C. Mead and M. Ismail, *Analog VLSI implementation of neural systems.* Springer Science & Business Media, 1989, vol. 80.

[76] A. Sayal, S. T. Nibhanupudi, S. Fathima, and J. P. Kulkarni, “A 12.08-tops/w all-digital time-domain cnn engine using bi-directional memory delay lines for energy efficient edge computing,” *IEEE Journal of Solid-State Circuits*, vol. 55, no. 1, pp. 60–75, 2019.

[77] O. Panetas-Felouris and S. Vlassis, “A time-domain z- 1 circuit with digital calibration,” *Journal of Low Power Electronics and Applications*, vol. 12, no. 1, p. 3, 2022.

[78] K. Kim, W. Yu, and S. Cho, "A 9 bit, 1.12 ps resolution 2.5 b/stage pipelined time-to-digital converter in 65 nm cmos using time-register," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 4, pp. 1007–1016, 2014.

[79] M. Santos, "Projeto de um registrador de tempo cmos," B.S. Thesis, Federal University of Minas Gerais, 2022.

[80] M. Z. Straayer, "Noise shaping techniques for analog and time to digital converters using voltage controlled oscillators," Ph.D. dissertation, Massachusetts Institute of Technology, Department of Electrical Engineering . . . , 2008.

[81] K. Kim, Y.-H. Kim, W. Yu, and S. Cho, "A 7 bit, 3.75 ps resolution two-step time-to-digital converter in 65 nm cmos using pulse-train time amplifier," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 4, pp. 1009–1017, 2013.

[82] G. Chen, D. C. Bui, X. Yu, M. Z. Islam, A. Kobayashi, and K. Niitsu, "A 72-nw 440-mv time register using stacked-nmos-switched gated delay cell in biomedical applications," in *2020 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*. IEEE, 2020, pp. 220–223.

[83] R. Tokheim, *Digital Electronics: Principles and Applications*. McGraw Hill, 2013.

[84] "Tsmc 180-nm process technology," accessed: 2022-08-11. [Online]. Available: https://www.tsmc.com/english/dedicatedFoundry/technology/logic/l_018micron

[85] P. Locatelli, "Digital mac units - vhdl," 2022. [Online]. Available: https://github.com/pedroslo/Digital-MAC-unit

[86] S. Immareddy and A. Sundaramoorthy, "A survey paper on design and implementation of multipliers for digital system applications," *Artificial Intelligence Review*, pp. 1–29, 2022.

[87] P. Balasubramanian, C. Dang, D. L. Maskell, and K. Prasad, "Approximate ripple carry and carry lookahead adders—a comparative analysis," in *2017 IEEE 30th International Conference on Microelectronics (MIEL)*. IEEE, 2017, pp. 299–304.

[88] S. Deepak and B. J. Kailath, "Optimized mac unit design," in *2012 IEEE international conference on electron devices and solid state circuit (EDSSC)*. IEEE, 2012, pp. 1–4.

[89] B. Harish, M. Rukmini, and K. Sivani, "Design of mac unit for digital filters in signal processing and communication," *International Journal of Speech Technology*, vol. 25, no. 3, pp. 561–565, 2022.

[90] S. Gopal *et al.*, "A spatial multi-bit sub-1-v time-domain matrix multiplier interface for approximate computing in 65-nm cmos," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 3, pp. 506–518, 2018.

[91] A. Amaravati *et al.*, “A 55-nm, 1.0–0.4 v, 1.25-pj/mac time-domain mixed-signal neuromorphic accelerator with stochastic synapses for reinforcement learning in autonomous mobile robots,” *IEEE Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 75–87, 2018.

[92] M. Douthwaite *et al.*, “A time-domain current-mode mac engine for analogue neural networks in flexible electronics,” in *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2019, pp. 1–4.

[93] H. Zhang, J. He, and S.-B. Ko, “Efficient posit multiply-accumulate unit generator for deep learning applications,” in *2019 IEEE international symposium on circuits and systems (ISCAS)*. IEEE, 2019, pp. 1–5.

[94] H. Zhang, D. Chen, and S.-B. Ko, “New flexible multiple-precision multiply-accumulate unit for deep neural network training and inference,” *IEEE Transactions on Computers*, vol. 69, no. 1, pp. 26–38, 2019.

[95] Y.-J. An, K. Ryu, D.-H. Jung, S.-H. Woo, and S.-O. Jung, “An energy efficient time-domain temperature sensor for low-power on-chip thermal management,” *IEEE sensors journal*, vol. 14, no. 1, pp. 104–110, 2013.