UNIVERSITÀ DELLA CALABRIA

UNIVERSITA' DELLA CALABRIA

Dipartimento di Ingegnieria Informatica, Modellistica, Electtronica e Sistemistica

(DIMES)

Dottorato di Ricerca in

Information and Communication Technologies

CICLO

XXXI

TITOLO TESI

Dual Mode Logic-Based Design of Variable-Precision Arithmetic Circuits

Settore Scientifico Disciplinare ING-INF/01

Coordinatore: Ch.mo Prof. Felice Crupi

Firma

Supervisore/Tutor: Ch.mo Prof. Marco Lanuzza

Firma

Dottorando: Paúl Romero

Firma

Contents

Abstract

The ever growing technological progress has an unquestionable impact on our society and, with the recent emergence of innovative technological paradigms, such as Internet of Things (IoT), Artificial Intelligence (AI), Virtual Reality (VR), 5G, Edge Computing, etc, it is expected that it will take a more and more dominant role in the coming decades. Obviously, the full development of all these new technologies requires the design of specialized hardware to faithfully and efficiently implement specific applications and services. In this sense, the demand of electronic circuits and systems with small area, flexible processing capability, high performance, and low energy consumption, has recently become one of the major concerns in different research areas, such as computing, communications, automation, etc.

In this context, this thesis work entitled "DUAL MODE LOGIC-BASED DESIGN OF VARIABLE-PRECISION ARITHMETIC CIRCUITS" aims to provide a contribution in the research of new design solutions for energy-efficient computing platforms, while also keeping high performance. In this regard, several strategies can be explored at different design abstraction levels, from system-level down to device-level. Among these, the design of variable-precision arithmetic circuits is a well-known approach to achieve more energy-efficient computing platforms when dealing with lossy multimedia applications (e.g., audio/video/image processing) where a reduction of the operation precision can be typically tolerated under the acceptable accuracy loss. At the same time, other solutions can be implemented at both circuit- and logic-level. In this regard, a new logic

family, namely Dual Mode Logic (DML), has recently emerged as an alternative design methodology to the existing digital design techniques. It was originally proposed as a combination of CMOS static and dynamic logics to allow on-the-fly controllable switching at the gate level between static and dynamic operation modes according to system requirements, input-driven control, and/or by designer considerations. Such modularity typically offers greater performance/energy trade-off flexibility in the design and optimization of digital circuits, especially for applications with a flexible workload, such as in multi-precision arithmetic circuits.

In this thesis work, the benefits of the DML design approach with respect to the standard CMOS style are first highlighted on a flexible circuit benchmarck, consisting of 10 levels of 11-stage NAND/NOR chains. In this case, the DML implementation takes advantage of its capability that allows operating in a combined (mixed) mode, i.e. working at the same time partly statically and partly dynamically, thus leading to fully exploit the benefits of the two DML operation modes for better energy-performance trade-offs. Then, the flexibility inherently offered by the DML is exploited to design a double-precision (8×8-bit or 16×16-bit) carry-save adder (CSA)-based array multiplier with the aim of demonstrating the potential in combining the two aforementioned design solutions (i.e., multi-precision computing and DML methodology) in the design and optimization of arithmetic circuits. As a matter of fact, the DML dual operation ability is potentially very attractive to efficiently trade performance and energy consumption between the operations at different precisions in on-demand multi-precision digital circuits. This occurs in the proposed DML multiplier working in a mixed operation mode, i.e., by employing the DML static and dynamic mode for lower- and higher-precision operations, respectively. On one hand, the use

of the dynamic mode for higher-precision operations ensures higher performance as compared to its standard static CMOS counterpart (16% gain on average) at the cost of higher energy consumption. On the other hand, such energy penalty is counterbalanced at lower-precision operations for which the static mode is enabled in the DML circuit. Overall, the adoption of the mixed operation mode in the proposed DML multiplier proves to be beneficial to achieve a better performance/energy trade-off with respect to the standard static CMOS implementation and to the cases when using the DML static or dynamic mode for both operations at the two different precisions. When compared to its CMOS counterpart, the proposed DML design operating in the mixed mode exhibits an average improvement of 15% in terms of energy-delay product (EDP) under wide-range supply voltage scaling. Such benefit is maintained over process-voltage-temperature (PVT) variations.

Sommario

Il progresso tecnologico ha un impatto indiscutibile sulla nostra società e con il recente emergere di paradigmi tecnologici innovativi, quali Internet of Things (IoT), Intelligenza Artificiale (IA), Realtà Virtuale (RV), 5G, Elaborazione al Margine, ecc., si prevede che assumerà un ruolo sempre più dominante nei prossimi decenni. Ovviamente, lo sviluppo di tutte queste nuove tecnologie richiede la progettazione di hardware specializzato per implementare efficacemente le applicazioni e i servizi richiesti. In tal senso, la domanda di circuiti e sistemi elettronici compatti ad elevate prestazioni, basso consumo di energia e con capacità di elaborazione flessibile, è diventata recentemente una delle maggiori preoccupazioni in diversi ambiti di ricerca, quali l'informatica, le telecomunicazioni, l'automazione, ecc.

In questo contesto, questo lavoro di tesi dal titolo "DUAL MODE LOGIC-BASED DESIGN OF VARIABLE-PRECISION ARITHMETIC CIRCUITS" si propone di fornire un contributo nella ricerca di nuove soluzioni progettuali per piattaforme di calcolo ad alta efficienza energetica, mantenendo allo stesso tempo prestazioni elevate. A tal fine, è possibile esplorare diverse strategie ai differenti livelli di astrazione della progettazione, dal livello di sistema fino al livello del dispositivo. Tra le varie strategie possibili, la progettazione di circuiti aritmetici a precisione variabile è un approccio ben noto per ottenere piattaforme di calcolo più efficienti dal punto di vista energetico per applicazioni multimediali (ad esempio, elaborazione audio/video/di immagini) in cui una riduzione della precisione delle operazioni può essere tipicamente tollerata entro un limite accettabile di perdita di accuratezza. Allo stesso tempo, altre soluzioni

progettuali possono essere implementate a livello sia circuitale che logico. A tal riguardo, una nuova famiglia logica, vale a dire la Dual Mode Logic (DML), è recentemente emersa come una metodologia di progettazione alternativa alle tecniche di progettazione digitale esistenti. Questa metodologia è stata originariamente proposta come una combinazione tra le logiche CMOS statiche e dinamiche per consentire una commutazione controllabile a livello di porte logiche tra le modalità operative statiche e dinamiche sulla base di requisiti di sistema, di una strategia di controllo dipendente dai segnali di ingresso e/o di considerazioni del progettista. Tale modularità offre in genere una maggiore flessibilità nella ricerca del miglior compromesso tra prestazioni e consumo di energia durante la progettazione e l'ottimizzazione dei circuiti digitali, in particolare per le applicazioni con un carico di lavoro flessibile, come nei circuiti aritmetici a precisione multipla.

In questo lavoro di tesi, i vantaggi dell'approccio progettuale DML rispetto allo stile standard CMOS sono dapprima evidenziati su un circuito di test flessibile, che consiste in 10 livelli di catene a 11 stadi costituite da porte logiche NAND/NOR. In questo caso, l'implementazione DML beneficia della sua funzionalità intrinseca che consente di operare in modalità combinata (mista), cioè facendo lavorare il circuito in parte in modalità statica e in parte in modalità dinamica, sfruttando così appieno i vantaggi delle due modalità operative DML per ottenere un miglior compromesso tra prestazioni e consumo di energia. Le flessibilità intrinsecamente offerta dalla DML è inoltre sfruttata per progettare un moltiplicatore a doppia precisione (8×8 o 16×16 bit) basato su un array di sommatori di tipo "carry-save" al fine di dimostrare le potenzialità nel combinare le due soluzioni progettuali summenzionate (ovvero il calcolo a precisione multipla e la

tecnica DML) nella progettazione e ottimizzazione di circuiti aritmetici. Difatti, l'abilità di doppia modalità operativa della DML è potenzialmente molto interessante per ricercare il miglior compromesso prestazioni/consumo di energia tra le operazioni a diversa precisione nei circuiti digitali a precisione multipla. Ciò si verifica nel moltiplicatore DML proposto quando esso lavora in una modalità operativa mista, vale a dire impiegando la modalità statica e dinamica rispettivamente per le operazioni a precisione inferiore e superiore. Da un lato, l'uso della modalità dinamica per le operazioni a maggiore precisione garantisce prestazioni più elevate rispetto alla controparte CMOS statica standard (con un guadagno medio del 16%) al costo di un maggiore consumo energetico. D'altra parte, tale penalità energetica è controbilanciata alle operazioni a precisione inferiore per le quali viene impiegata la modalità statica nel circuito DML. Complessivamente, l'adozione della modalità operativa mista nel moltiplicatore DML proposto si rivela vantaggiosa per ottenere un migliore compromesso tra prestazioni e consumo di energia rispetto all'implementazione standard CMOS statica e ai casi in cui si utilizza la modalità dinamica o la modalità statica per entrambe le operazioni alle due diverse precisioni. Rispetto alla sua controparte CMOS, il moltiplicatore DML che opera in modalità mista consente di ottenere un miglioramento medio del 15% in termini di prodotto energia-ritardo in un range ampio di tensioni di alimentazione. Tale vantaggio viene mantenuto anche quando si considerano contemporaneamente variazioni di processo, di tensione di alimentazione e di temperatura.

Chapter 1

1 Introduction

1.1 Towards high-speed and energy-efficient designs in the IoT era

In recent decades the mankind has witnessed a growing technological progress in different areas, such as computing, electronics, communications, automation, etc., with the development of processors, digital systems and computing platforms featuring high performance, small area and low energy consumption. More recently, the upcoming Internet of Things (IoT) era has increased the demand for complex computing on a very small energy budget, thus providing a strong motivation to design high-speed and energy-efficient arithmetic and memory circuits [1]-[3].

Over the last decades, the above requirements have been mainly achieved by scaling both the semiconductor technology and the supply voltage (V_{DD}). In 1965 Gordon Moore predicted that the technology scaling would allow increasing the number of transistors within a chip to the double every 12-24 months at minimum economical cost [4]. This trend has been effectively followed by semiconductor industry increasing to millions the transistor count per chip, as shown in Figure 1.1.

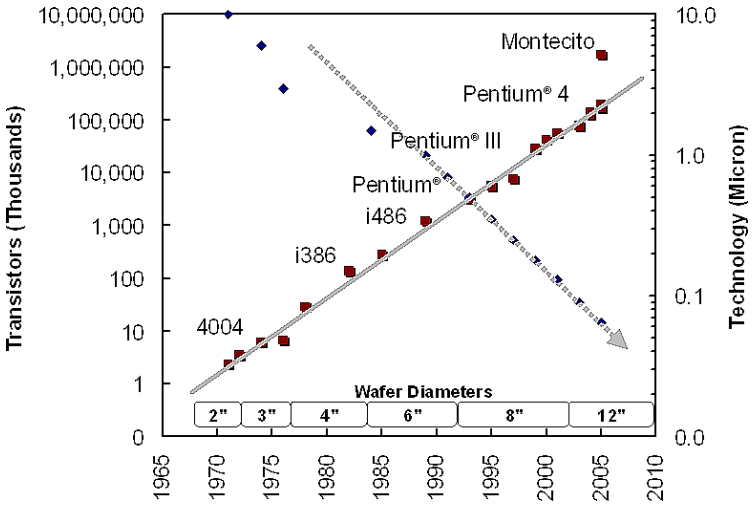

Figure 1.1. Real scaling trends over the years 1975-2010 obeying the Moore's law [5].

In general, the CMOS technology scaling translates into an improvement of transistor and interconnection speed, transistor density, and switching energy consumption. However, in the last years, as technology nodes entered into the deep sub-micron region, leakage power has become a significant issue in VLSI (Very Large Scale Integration) designs, thus limiting the scaling strategy [6]. In this regard, V_{DD} scaling is certainly a very effective lever to reduce power consumption, but at the cost of reduced performance and higher sensitivity to process variability [7].

Therefore, the diminishing benefits from technology scaling, along with (*i*) the growing demand in mobile devices [8], especially in the IoT scenario, (*ii*) the battery technology's inability to provide sufficient low-cost and size solutions for these devices [9], and (*iii*) the contemporary growth in amounts of data processed by computing platforms from mobile devices to data centers and the consequent increase of energy consumption (see Figure 1.2) [10], have motivated researchers to look at new solutions for energy-efficient computing systems, while also keeping high performance.

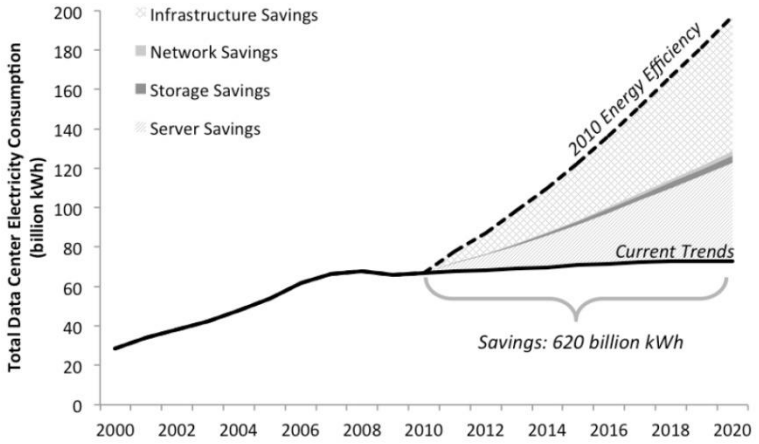

Figure 1.2. Past and projected growth rate of total US data center energy use from 2000 until 2020 [10].

Overall, the design optimization process aimed at achieving a better energy/performance trade-off can occur at various abstraction levels [9], from the device- up to the system-level, as shown in Figure 1.3. The designer can then implement several solutions at different levels. This must be done taking into account that each level typically depends on each others, i.e. a propagation flow exists among design levels. In particular, the requirements/specifications usually propagate from the top to the bottom, whereas the restrictions propagate from the bottom to the top [11].

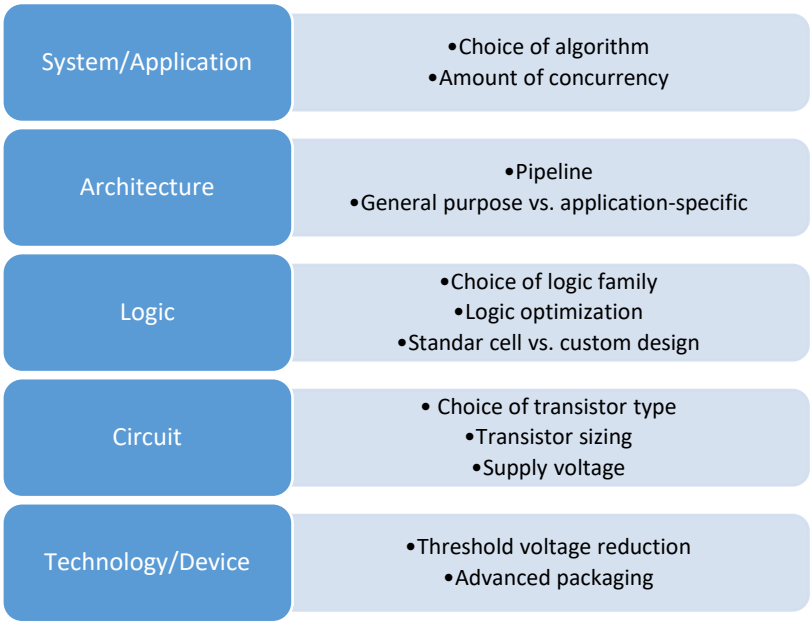

Figure 1.3. The design abstraction levels.

1.2 Performance and power/energy metrics

Performance and power/energy consumption are the main figures-of-merit (FOMs) or quality metrics in VLSI design, along with the cost, the functionality and the robustness [9], [12], [13]. It is worth pointing out that which one of these FOMs is most important depends on the specific application requirements. For instance, the speed is typically the most crucial property in computing servers. On the contrary, for mobile applications such as cell phones, the energy consumption becomes the most prominent FOM [12]. However, the designer usually have to efficiently trade these two FOMs to ensure good performance, while also keeping energy consumption low. For this reason, it is useful to properly define these metrics and how to quantify them [13].

From a system designer perspective, the performance defines the computational load that a digital system can manage. For instance, a microprocessor is often characterized by the number of instructions it can execute per second. This performance metric depends both on the architecture (e.g., the number of instructions the circuit can execute in parallel) and the specific design style of the circuit [12]. When focusing on the circuit design, performance is typically expressed by the duration of the clock period or its frequency. The minimum value of the clock period for a given design is determined by a series of factors, such as the propagation time of the signals through the logic, the time it takes to get data in and out of the registers, and the delay time related to the clock arrival. Generally, the propagation delay (t_p) of a logic gate represents the delay of a signal when passing through that gate. It is typically measured between the 50% transition points of the input and output signals [12]. Because a gate often exhibits different responses for rising and falling input signals, two different delay times has to be defined: the t_{pLH} related to a low (L) to high (H) output transition, and the t_{pHL} related to an H→L output transition. The propagation delay is then defined as the average of these two values, as given by [12]

$$t_p = \frac{t_{pLH} + t_{pHL}}{2} \tag{1.1}$$

In general, the transient response of a gate to an input change is mainly dominated by its output capacitance, although it also depends on the strength (i.e. the on-resistance) of its transistors [12]. Higher performance can be then achieved by keeping the output capacitance small or by decreasing the on-resistance of the transistors. Therefore, t_p is strictly related to the transistor technology and the circuit topology. However, it can depend upon other factors such as the slopes of the input and output signals. In this regard, it is necessary to introduce the rise (t_r) and fall (t_f) times, which are metrics

used to determine how fast a signal transits between two different levels. Generally, t_r and t_f are defined between the 10% and 90% points of the signal. In a digital circuit, they are mainly determined by the strength of the driving gates, and the load presented at nodes, which sums the contributions of the connecting gates (fan-out) and the wiring parasitics [12].

The power consumption of a VLSI design determines how much energy is consumed per operation, and hence how much heat the circuit dissipates. This strongly influences several critical design choices, such as the power-supply capacity, the battery lifetime, supply-line sizing, packaging, cooling requirements, etc. [9], [12]. Therefore, power dissipation is a crucial property that also affects feasibility, cost, and reliability of a design. In the context of high-performance computing platforms, power consumption limits determine the number of transistors/circuits that can be integrated within a single chip, and how fast they can switch. Due to the increasing demand of mobile devices and distributed computing, power/energy limitations impose severe restrictions on the number of computations that can be performed given a minimum time between battery recharges [12].
Depending upon the design issues, different power dissipation metrics must be considered. For instance, the peak power (P_{peak}) is crucial for supply-line sizing, while the average power dissipation (P_{av}) is important when dealing with cooling or battery requirements [12]. P_{peak} and P_{av} are defined as follows

$$P_{peak} = i_{peak} V_{DD} = \max\left[p(t) \right], \quad P_{av} = \frac{1}{T}\int_0^T p(t)\,dt = \frac{V_{DD}}{T}\int_0^T i_{DD}(t)\,dt \qquad (1.2)$$

where $p(t)$ is the instantaneous power, T is the clock period, i_{DD} is the current drawn from the supply voltage over the interval $[0,T]$, and i_{peak} is the maximum value of i_{DD} over that interval.

Typically, power/energy consumption can be decomposed into static and dynamic components. The latter occurs only during switching transitions, owing to the charging/discharging of capacitances and temporary short-circuit current paths between the supply rails. Therefore, it is proportional to the switching frequency, i.e. the higher the number of switching events, the higher the dynamic power consumption. On the other hand, the static consumption is present even when no switching events occur (e.g., during stand-by), owing to the static conductive paths between the supply rails or leakage currents [12]. Therefore, the three dominating components to the total power/energy consumption in a digital circuit are:

- static power (P_{stat}) and energy (E_{stat});
- dynamic power (P_{dyn}) and energy (E_{dyn}) due to charging and discharging of capacitances;
- short-circuit power (P_{sc}) and energy (E_{sc}) due to direct path currents during switching.

Accordingly, overall we have:

$$P_{total} = P_{stat} + P_{dyn} + P_{sc}, \quad E_{total} = E_{stat} + E_{dyn} + E_{sc} \tag{1.3}$$

The static (or steady-state) power and energy consumption of a circuit are defined by the following expressions, respectively

$$P_{stat} = I_{stat} V_{DD} = I_{leak} V_{DD}, \quad E_{stat} = I_{leak} V_{DD} T \tag{1.4}$$

where I_{stat} is the current that flows between V_{DD} and ground in absence of switching activity. Ideally, the static current of a CMOS inverter is equal to zero, since the NMOS and PMOS devices are never on simultaneously in steady-state operation. In reality, the currents flowing through the reverse-biased diode junctions of the transistors located between the source or drain and the substrate, and the subthreshold conduction of the transitors (i.e. a MOSFET is never completely turn off) translate into a non-zero leakage

current (I_{leak}) and hence into a static power consumption [9], according to (1.4).

As stated above, dynamic power consumption in a digital circuit is mainly associated to charging and discharging of capacitances occurring during switching events. Therefore, it strictly depends on the switching activity of the circuit. Considering a whole switching cycle (consisting of an L→H and an H→L transition) in a simple CMOS inverter, such dynamic power /energy consumption can be given by [12]

$$P_{dyn} = C_L V_{DD}^2 f_{0\to1}, \quad E_{dyn} = C_L V_{DD}^2 \tag{1.5}$$

where C_L is the load capacitance and $f_{0\to1}$ is the switching activity of the circuit, i.e. the frequency of energy-consuming transitions corresponding to L→H output transitions for static CMOS. Considering that the switching activity in a complex circuit depends on several factors (e.g., the nature and the statistics of the input signals, the circuit topology, the function to be implemented, etc.), (1.5) is usually rewritten for a generic logic gate as follows [12]

$$P_{dyn} = C_L V_{DD}^2 \alpha_{0\to1} f = C_{EFF} V_{DD}^2 f, \quad E_{dyn} = C_{EFF} V_{DD}^2 \tag{1.6}$$

where f is the clock frequency, $\alpha_{0\to1}$ is the activity factor, i.e. the probability that a clock event results in a 0→1 event at the output of the gate, and $C_{EFF} = \alpha_{0\to1} C_L$ is the effective capacitance, i.e. the average capacitance switched every clock cycle. Note that technology advances typically result in higher values of switching activity as t_p decreases, and also higher total capacitance onto a chip due to the increased integration density [12].

During switching transitions, another contribution to the power dissipation of a digital circuit is due to the finite slope of the real input signals. Indeed, this causes direct (or short-circuit) current paths between V_{DD} and ground for a short period of time during switching, in which NMOS and PMOS

transistors are conducting simultaneously. Assuming that the resulting current spikes during this short period can be approximated as triangles and considering a CMOS inverter featuring symmetrical rising and falling responses, the short-circuit power dissipation/energy consumption can be given by [12]

$$P_{sc} = t_{sc} V_{DD} I_{peak} f_{0\to1}, \quad E_{sc} = t_{sc} V_{DD} I_{peak} \tag{1.7}$$

where t_{sc} is the time both devices are conducting and I_{peak} represents the peak of the short-circuit current during t_{sc}, which strongly depends on the ratio between input and output slopes [9]. It is worth pointing out that the short-circuit power consumption, as well as the dynamic power consumptions, depends on the switching activity.

Therefore, using (1.3)-(1.7), we can define the total power consumption of a simple CMOS inverter as the sum of its three components:

$$P_{total} = V_{DD} I_{leak} + \left(C_L V_{DD}^2 + t_{sc} V_{DD} I_{leak} \right) f_{0\to1} \tag{1.8}$$

In the past, dynamic power consumption resulting from charging and discharging capacitances was dominant in typical CMOS circuits. More recently, with the introduction of deep sub-micron technologies and the consequent scaling of V_{DD} and threshold voltage of transistors, leakage currents have become more substantial to the point of being the primary source of power dissipation [9].

Performance and power consumption are strictly correlated in a digital circuit because the propagation delay is mainly determined by the speed at which a given amount of energy can be stored on the capacitors. The faster the energy transfer, i.e. the higher the power dissipation, the faster the circuit [12]. The product between power consumption and delay, namely power-delay product (PDP), is considered a quality measure for a digital circuit

[13]. It simply represents the average energy consumed by a circuit per switching event. However, the validity of the PDP as a quality metric in VLSI design is not fully recognized since, for instance, it can be arbitrarily reduced by decreasing the supply voltage. From this perspective, the optimum V_{DD} in a specific circuit would be the lowest possible value that still assures its functionality. But this occurs at the cost of lower performance [12]. For this reason, a more relevant metric is the energy-delay product (EDP), which allows combining performance and energy measurements. EDP is typically used in VLSI design as the ultimate quality metric to find the best energy/performance trade-off [12]. For instance, considering the impact of the supply voltage on the EDP, we can observe that higher V_{DD} translates into lower delay but higher energy consumption, while the opposite occurs for lower V_{DD}. This means that an optimum operation point (i.e., an optimum V_{DD}) should exist. In general and more specifically in the energy-constrained contexts such as the IoT, the design optimization is thus a trade-off process aimed at minimizing energy consumption for a given performance requirement or alternatively at maximizing performance for a given energy budget. As previously stated, this optimization process can occur at different design levels. As shown in Figure 1.4, it takes place in the energy-delay (E-D) plane with the aim of extending as much as possible the optimization space to facilitate the attainment of both delay and energy targets.

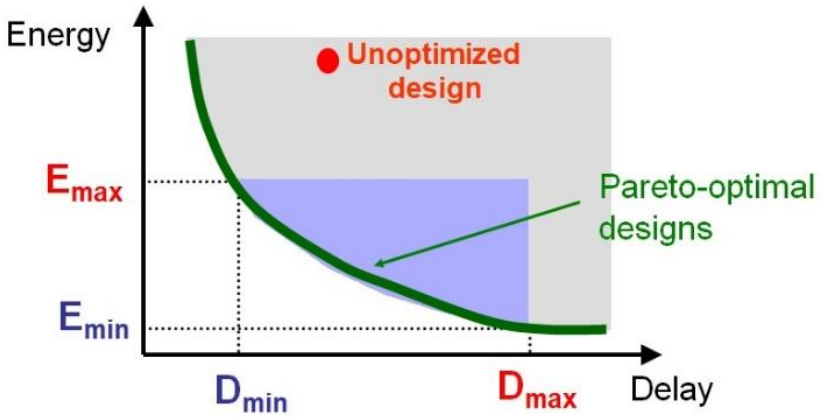

Figure 1.4. Energy-delay design space [9].

1.3 Purpose and organization of this work

As stated above, energy consumption represents a major design constraint in today computing systems. Moreover, the IoT paradigm has recently increased the demand for complex digital signal processing (DSP), multimedia systems and portable devices with flexible processing ability, high performance, small area, and low energy consumption [1]-[3]. In this regard, several solutions and strategies can be explored at different design abstraction levels. Among these, the design of variable-precision arithmetic units is a well-known approach to achieve more energy-efficient computing platforms [14]-[16]. This is particularly suitable for lossy multimedia applications (e.g., audio/video/image processing) where reducing the precision of arithmetic operations can be tolerated under the acceptable accuracy loss [16]. The design of multi-precision arithmetic units featuring an optimal energy/performance trade-off can be also facilitated by making appropriate choices at both circuit- and logic-level. To this purpose, a new logic family, namely Dual Mode Logic (DML), has been very recently

proposed with the aim of providing on-the-fly controllable switching at the gate level between static and dynamic operation modes [17]-[24]. This dual modularity typically allows wider energy/performance trade-off flexibility in the design and optimization of digital circuits [18].

In this context, the main purpose of this thesis work is to demonstrate the potential in combining the two aforementioned design strategies (i.e., multi-precision computing and DML design approach) in the design and optimization of arithmetic circuits (e.g., adders, multipliers, etc.). As a matter of fact, the flexibility inherently offered by DML is potentially very attractive to efficiently trade performance and energy consumption between the operations at different precisions in on-demand variable-precision digital circuits. In particular, as case study, this work mainly focuses on the design of a double-precision (8×8-bit or 16×16-bit) carry-save adder (CSA)-based array multiplier by exploiting the DML design approach [25]. For comparison purpose, the same benchmark has been also designed by using the standard static CMOS logic. All the circuits reported in this thesis have been implemented in a commercial 1.2V 65-nm low-power CMOS technology and characterized by means of circuit simulations in a commercial computer-aided circuit design tool, such as Cadence Virtuoso environment.

The rest of this thesis is structured as follows. Chapter 2 provides a brief overview of the most common CMOS logic design styles and it introduces the DML family. Chapter 3 first compares the DML design approach with the standard static CMOS style on a flexible circuit benchmark consisting of 10 levels of 11-stage NAND/NOR chains. Then, it describes in detail the proposed DML implementation of a double-precision (8×8-bit or 16×16-bit)

CSA-based multiplier, whose simulation results are provided and discussed over a wide range of process-voltage-temperature (PVT) variations in comparison with its static CMOS counterpart. Finally, Chapter 4 draws the main conclusions of this thesis work.

Chapter 2

2 Survey of Logic Families

Several circuit styles (or logic families) can be used for the implementation of a given logic function. Since each of these styles inherently holds its own advantages and drawbacks, the designer has to make an appropriate choice depending on the specific application and its requirements in terms of area, speed, energy consumption, robustness and reliability.

This chapter first provides a brief review of the most common design styles used to implement logic gates, blocks and digital circuits, such as the complementary (or standard) static CMOS and the dynamic CMOS. Both of these logic styles serve as basic building blocks for a novel and very promising logic design style, namely Dual Mode Logic (DML), whose

operating principle and main characteristics are then presented and discussed. The chapter ends by providing some guidelines to be used for the selection of the appropriate logic style when designing a digital circuits addressed to a specific application.

2.1 Complementary static CMOS logic

The most widely used logic family is the complementary static CMOS. Along with other circuit styles, such as ratioed logic, i.e. pseudo-NMOS and Differential Cascode Voltage Switch Logic (DCVSL) [26]-[28], and pass-transistor logic [29]-[31], such logic family belongs to the category of *static* (or *non-clocked*) circuits in which at every point in time each gate output is connected to either V_{DD} or ground through a low-resistance path, thus assuming at all times the value of the implemented logic function, except during the switching transitions. On the contrary, *dynamic* (or *clocked*) circuits are based on the temporary storage of signal values on the capacitance of high-impedance circuit nodes [12].

As shown in Figure 2.1, a generic standard static CMOS gate is based on the combination of two complementary networks: (*i*) the pull-up network (PUP) composed by PMOS devices, and (*ii*) the pull-down network (PDN) composed by NMOS devices. Note that all the inputs are distributed to both networks. The task accomplished by the PUN is to provide the connection between the output node and V_{DD} anytime the output of the implemented logic function has to be equal to "logic 1" on the basis of the inputs. Likewise, the PDN allows the connection between the output node and V_{SS} (typically the ground) when the output of the implemented logic function has to be "logic 0" on the basis of the inputs. Obviously, to properly

implement the given logic function, the two networks have to be built such that they are mutually exclusive, i.e. one and only one of the networks is active in steady state, thus always resulting into a *low-impedance* output node in steady state [12]. It is worth pointing that the complementary static CMOS gates are intrinsically inverting, thus requiring an additional inverter stage to implement non-inverting logic functions. Typically, NMOS and PMOS transistors are uniformly sized in complementary CMOS gates to have matching characteristics between PDN and PUN (i.e., same t_{pLH} and t_{pHL}), except for large fan-in gates for which a progressive transistor sizing approach is usually adopted [12].

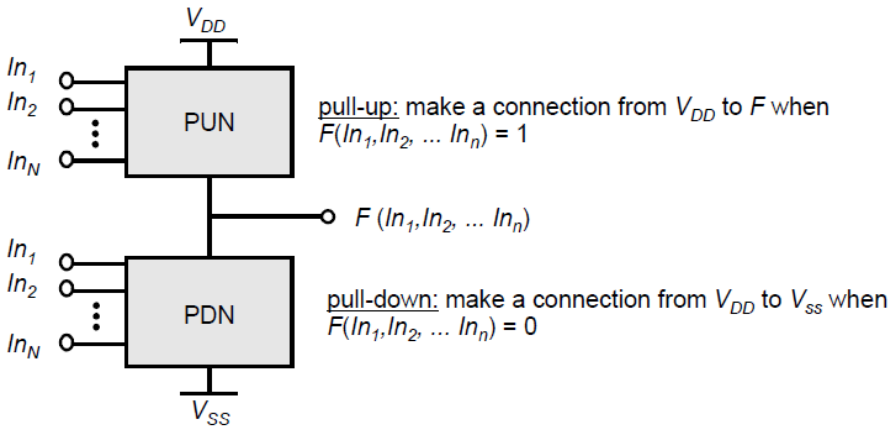

Figure 2.1. Basic scheme of an N-input standard static CMOS gate [12].

The advantages offered by the standard static CMOS design methodology are well known. Among these, we can mention the easiness of design, the robustness to noise, high scalability with technology and voltage, and until recent CMOS processes, almost no static power consumption. Conversely, the main drawback concerns the large amount of transistors required to implement a logic function (2*N* transistors for an *N*-input logic gate), which

translates into large input and output capacitances and hence reduced performance [32]. Moreover, as stated in Chapter 1, the advent of nanoscaled CMOS technologies has coincided with a dramatic increase of static leakage currents, thus leading to a no longer negligible contribution of the static power consumption.

2.2 Dynamic CMOS logic

As mentioned in the previous subsection, the complementary static CMOS requires $2N$ transistors to implement a logic gate with a fan-in of N. Different approaches were proposed to reduce the number of transistors required to implement a given logic function, including alternative static styles such as pseudo-NMOS and pass-transistor logics. For instance, the pseudo-NMOS logic style requires only $N + 1$ transistors to implement an N-input logic gate, but at the expense of higher static power dissipation [12]. An alternative logic style, namely dynamic CMOS, was proposed to achieve similar results in terms of area occupation, while avoiding static power consumption and also ensuring higher performance with respect to static CMOS circuits.

The basic scheme of an *n*-type (or pre-charge) dynamic CMOS logic gate is shown in Figure 2.2. The operation of a dynamic gate is divided into two phases, i.e. the pre-charge phase and the evaluation phase, orchestrated by a clock signal (CLK). To this purpose, as shown in Figure 2.2, a clocked PMOS pre-charge transistor (M_p) and a clocked NMOS evaluation (footer) transistor (M_e) are used in the PUN and at the bottom of the PDN, respectively, while the PDN is built as in the complementary static CMOS. In this way, during the pre-charge phase (i.e. when CLK = 0), the output

node is pre-charged to V_{DD} through the PMOS transistor M_p, while the NMOS transistor M_e is off, thus disabling the PDN path and hence preventing any static power consumption during this period. Conversely, during the evaluation phase (i.e. when CLK = 1), the precharge transistor M_p is off, and the evaluation transistor M_e is turned on. Thus, the output the output node is conditionally discharged to ground depending on the inputs and the logic function implemented by the PDN. Whether the input values are such that the PDN is enabled, then a low-resistance path is established between the output node and ground to discharge the output node. On the contrary, if the inputs do not enable the PDN, the pre-charged value remains

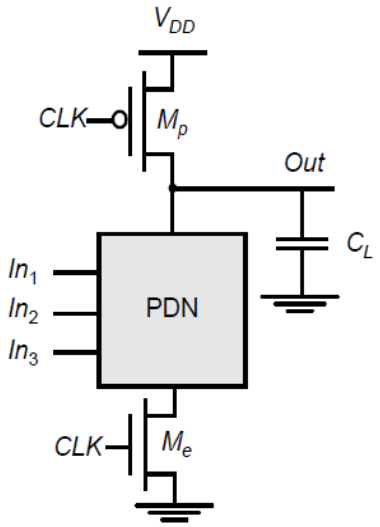

Figure 2.2. Basic scheme of an N-input (n-type) dynamic CMOS gate [12].

stored on the capacitance of the output node. Accordingly, the gate inputs can lead at most to one output transition during each evaluation phase. It is worth noting that, when the PDN is disabled during the evaluation phase, the output is in the high-impedance state. This represents the main difference with respect to static CMOS circuits, where a low-resistance path

between the output and V_{DD} or ground always exists. Therefore, in an *n*-type dynamic CMOS gate the logic function is implemented by the PDN composed by NMOS devices. In a similar way, we can build a *p*-type (or pre-discharge) dynamic CMOS gate by using a clocked NMOS pre-discharge transistor in place of the PDN and a clocked PMOS evaluation (header) transistor at the top of the PUN, which is built in that case as in the complementary static CMOS to implement the given logic function.

In general, when compared to the complementary static CMOS, the main advantages of the dynamic CMOS style are the smaller area occupation (N + 2 transistors versus 2N for an N-input logic gate) and the faster switching speeds owing to the reduced output capacitance (attributed to both the lower number of transistors per gate and the single-transistor load per fan-in) and the higher drive strength during the evaluation phase. Note also that *n*-type (*p*-type) dynamic CMOS gates can be constructed without the clocked footer (header) evaluation transistor, thus reducing both the area occupation (only N + 1 transistors for an N fan-in gate) and the clock load. Moreover, ideally, no static current path ever exists between V_{DD} and ground. Instead, the main drawback concerns the overall power dissipation, which can be significantly higher as compared to a static logic gate, mainly due to the clock power and the higher switching activity [12]. In addition, the dynamic CMOS logic style presents several other issues, including the charge leakage, the charge sharing, the capacitive coupling, and the clock feedthrough, which make this design approach particularly tricky [12]. These limitations are typically intensified with technology and voltage scaling, and under process and temperature variations.

Another important issue that complicates the design of dynamic CMOS circuits regards the cascading of dynamic gates of the same topology (i.e.

using the same clock signal) when implementing more complex logic structures. This problem is illustrated in Figure 2.3, where two *n*-type dynamic inverters are cascaded. During the precharge phase (i.e., CLK = 0), the outputs of both inverters are pre-charged to V_{DD}. Assuming that the input of the first inverter (In) makes a 0→1, as shown in Figure 2.3, on the rising edge of CLK, the output of the first inverter (Out_1) starts to discharge. The output of the second inverter (Out_2) should remain in the pre-charged state since its expected value is 1 due to the 1→0 transition of Out_1 during the

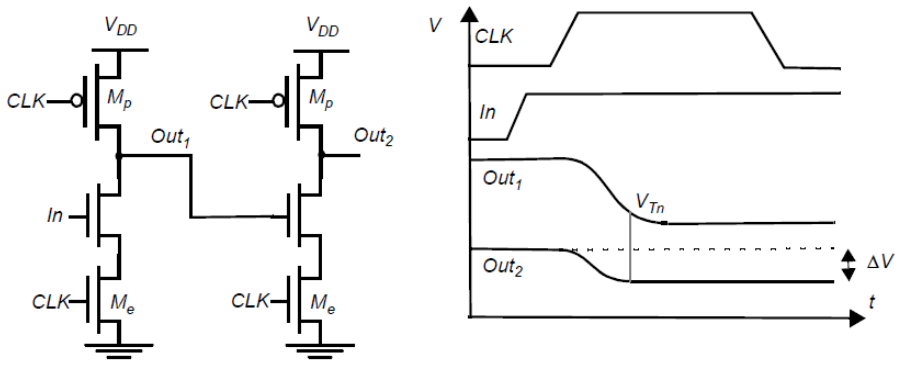

Figure 2.3. Cascading issue for n-type dynamic CMOS blocks [12].

evaluation phase. However, owing to the finite propagation delay for the input to discharge Out_1 to ground, Out_2 also starts to discharge. As a consequence, there is a conducting path between Out_2 and ground and hence some charge is lost at Out_2 as long as Out_1 exceeds the threshold voltage of the NMOS transistor of the second inverter. This conducting path is disabled only when Out_1 reaches the NMOS threshold voltage, thus turning off the transistor. This behavior leads Out_2 to an intermediate voltage level, which is not correct. Moreover, the correct voltage level on Out_2 will not be recovered since dynamic gates rely on capacitive storage and thus they do

not have static restoration. The resulting charge loss then produces reduced noise margins and potential malfunctioning [12].

This cascading issue occurs becasue the outputs of each gate (corresponding to the inputs of the next stages) are all pre-charged to V_{DD}. This can cause unwanted discharge at the beginning of the evaluation phase, as in Figure 2.3. A possibile solution to prevent this problem is setting all the inputs to 0 during the precharge phase. In tis way, all NMOS device in the PDN are in off after the precharge, thus avoiding inadvertent discharging of the storage capacitors during the evaluation. This means that correct operation is ensured as long as the inputs can only make a single 0→1 transition during the evaluation period. In this regard, two alternative dynamic design styles were proposed to implement this rule [12]:

- the Domino logic style;
- the NORA/*np*-CMOS logic style.

Domino logic style typically consists of cascading *n*-type dynamic logic blocks followed by a static inverter, as shown in Figure 2.4. This topology ensures that all inputs are set to 0 at the beginning of the evaluation phase, thus solving the aforementioned cascading issue. Indeed, during the pre-charge phase, the ouput of the *n*-type dynamic gates is charged to V_{DD}. Accordingly, the output of the static inverters are set to 0. Then, during the evaluation phase, the dynamic gates conditionally discharge to ground its output node, and hence the ouput of the inverters makes a conditional 0→1 transition. In addition to solving the cascading issue, the additional static inverter ensures higher noise immunity due to the fact the fan-out of each gate is driven by a static inverter with a low-impedance path. Such buffer

stage also decreases the capacitance at the dynamic output node by separating internal and load capacitances [12]. However, Domino logic style

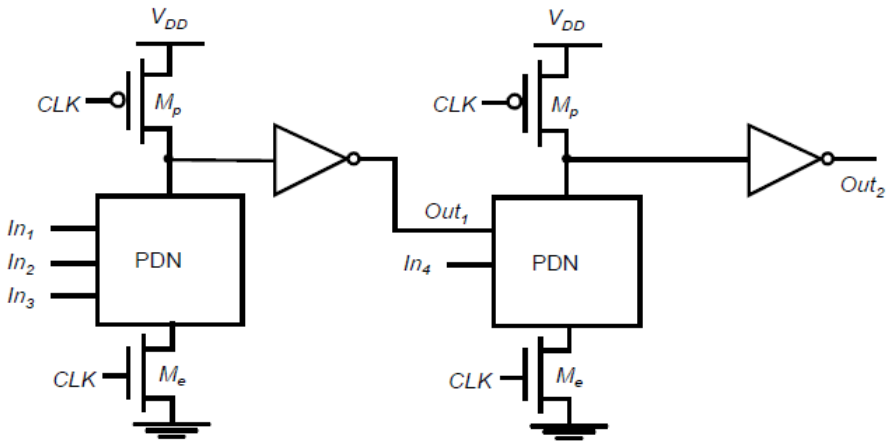

Figure 2.4. Cascading of n-type gates in Domino CMOS logic style [12].

exhibits some limiting aspects mainly concerning the intrinsic non-inverting property, the high clock load owing to the need to use the footer evaluation transistor in the cascaded *n*-type logic blocks for preventing an extended pre-charge phase, and the reduced signal propagation time due to the presence of extra static inverters in the critical path [12].

An alternative solution to efficiently cascade dynamic logic blocks is represented by *np*-CMOS circuit style. As shown in Figure 2.5, it exploits the duality between *n*-type and *p*-type logic gates by alternatly cascading them, where the *p*-type logic gates are controlled by a flipped clock signal with respect to that of the *n*-type gates. In this way, *n*-type gates can directly drive *p*-type gates, and vice-versa. Furthermore, similar to Domino logic,

the ouput of *n*-type (*p*-type) gates can be connected to another *n*-type (*p*-type) gate by inserting an extra static inverter (see Figure 2.5).

Therefore, *np*-CMOS logic guarantees high design flexibility, while also ensuring high performance and solving the cascading issue. The main

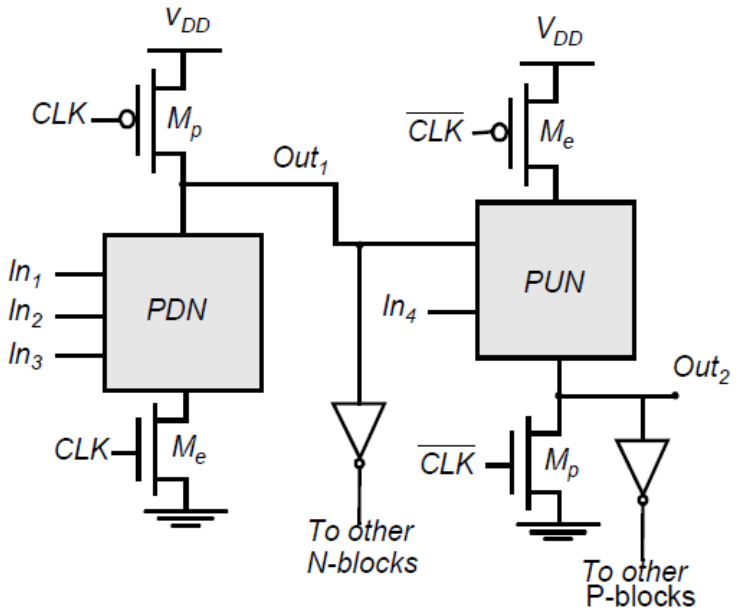

Figure 2.5. Cascading of logic gates in np-CMOS logic style [12].

disadvanteges are the higher complexity in the clocking design, and the slower speed of *p*-type blocks, due to the lower drive strength of PMOS transistors, which typically leads to use wider PMOS transistors to equalize the propagation delays and hence to a greater area occupation.

2.3 Dual Mode Logic (DML)

The DML family was recently introduced as a combination of complementary static and dynamic CMOS logic style with the aim of providing an alternative design methodology to the existing digital design techniques [17]-[19]. As a matter of fact, it allows on-the-fly controllable switching at the gate level between static and dynamic operation modes according to system requirements, input-driven control, and/or by designer considerations [18]. As shown in several previous works [17]-[24], such dual operation capability offered by the DML approach provides greater performance/energy trade-off flexibility in the design and optimization of digital circuits. As shown in Figure 2.6, the DML static mode typically assures energy saving at the expense of lower performance. Alternatively, the DML dynamic mode ensures higher performance at the cost of larger

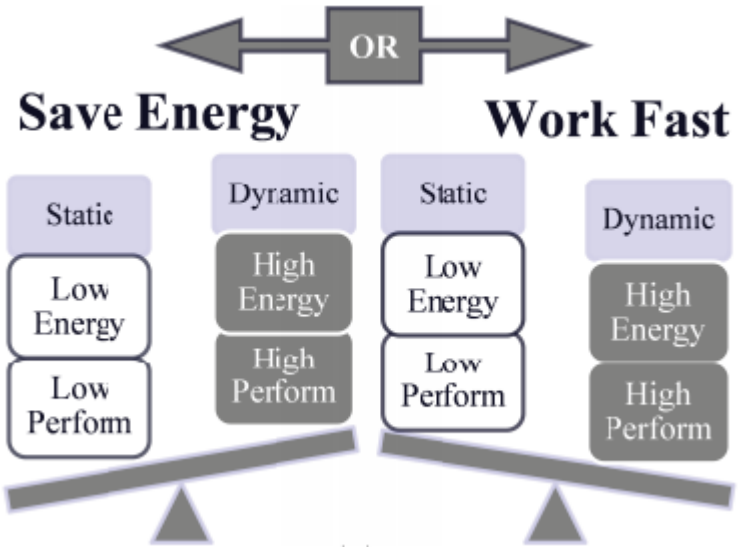

Figure 2.6. Performance/energy trade-off of DML gates in the two operation modes [18].

energy consumption [18]. This means that the two DML functional modes

exhibit different optimal curves in the energy-delay (E-D) space. Whereas the ability of on-the-fly controllable switching between the two operation modes theoretically allows the union of their optimal E-D curves, the DML design approach then provides an extended optimization space, thus making the attainment of E-D targets easier [22]. Moreover, it has been also shown that DML is fully functional, very robust and efficient for a wide range of supply voltage [18]-[21].

To implement the behavior described above, the structure of a basic DML gate relies on a conventional (complementary) static CMOS gate with an additional clocked pre-charge (or pre-discharge) transistor, which enables dynamic operation depending on a clock signal [20]. Similar to dynamic CMOS logic, DML gates can be implemented with or without a footer (or header) clocked evaluation transistor [22]. Figure 2.7 illustrates all the possible DML gate topologies: (a) footless Type-A with the pre-charge transistor and without the footer evaluation transistor, (b) headless Type-B with the pre-discharge transistor and without the header evaluation transistor, (c) footed Type-A with the pre-charge and the footer transistors, and (d) headed Type-B with the pre-discharge and the header transistors. The footer (or header) transistor is typically used to eliminate the ripple effect of the data advancing through the cascade and hence allowing faster pre-charge (pre-discharge) [18].

In all the topologies, during the static operation, the clocked pre-charge (or pre-discharge) transistor is always disabled, thus corresponding to CLK = 1 (0) in Type-A (Type-B) gates. As a result, the DML gates of both topologies

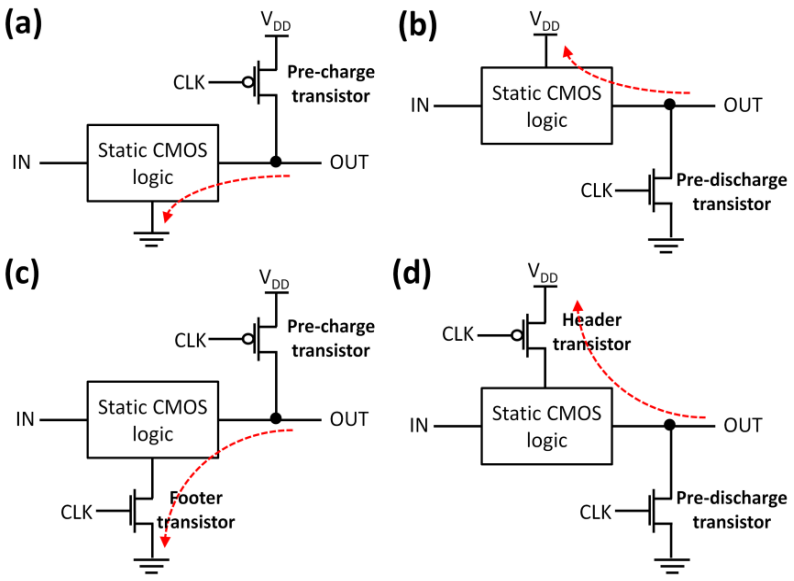

Figure 2.7. Basic DML gate topologies: (a) footless Type-A, (b) headless Type-B, (c) footed Type-A, and (d) headed Type-B. Red traces represent the evaluation paths for each topology.

retain the functionality of the static core gate, except for the extra negligible parasitic capacitance due to the additional clocked transistor [22]. On the contrary, when the dynamic operation is required, the clock signal is enabled for toggling, thus providing two separate operating phases, i.e. pre-charge (or pre-discharge) and evaluation, as in dynamic CMOS logic. During the first phase, the output is charged to V_{DD} in Type-A gates and discharged to ground in Type-B gates, whereas during the evaluation phase the output is evaluated according to the values of gate inputs [18]. Typically, the DML gates exhibit a very robust operation in both static and dynamic modes under PVT variations and at low supply voltage [18]-[20]. In particular, as compared to the conventional dynamic CMOS logic, the DML gates operating in the dynamic mode ensure higher robustness by the intrinsic active self-restorer network, i.e. the PUN in Type-A topology and the PDN

in Type-B topology, which allows sustaining glitches, charge leakage and charge sharing.

In general, when compared to its standard static CMOS counterpart, a DML circuit working in the static mode assures lower energy consumption with some performance degradation. Conversely, the dynamic mode can significantly improve the performance at the expense of higher energy consumption. The key factor to achieve the above behavior is the use of a proper transistor sizing methodology in the DML design, namely the unique sizing [18], [22]. It usually requires:

- the sizing of the evaluation networks in the static core (i.e. the PDN in Type-A gates and the PUN in Type-B gates, as shown in Figure 2.7) as in the standard static CMOS methodology, which typically consists of uniform transistor sizing to achieve a strength equivalent to that of one minimum-sized transistor;
- a downsizing of the active self-restore networks in the static core (i.e. the PUN in Type-A gates and the PDN in Type-B gates) by using minimum-sized transistors or anyway smaller devices with respect to the standard static CMOS design, which leads to a reduction of all capacitances in the DML gates, especially for the gate with large fan-in.

Accordingly, the downsizing of the self-restore networks is responsible for energy saving during the DML static operation mode at the expense of reduced performance due to their lower drive strength. At the same time, the low-resistive evaluation networks along with the capacitance reduction due to the downsized self-restore networks allows achieving fast operation in the DML dynamic mode and hence higher performance with respect to the static CMOS counterpart.

Note also that, when operating in the dynamic mode, the DML gates face the cascading issue as in the dynamic CMOS logic. Therefore, DML gates of the same topology can be connected through an extra static inverter as in the Domino logic style, otherwise the alternative solution consists of alternatly cascading Type-A and Type-B DML gates as in the *np*-CMOS logic style. Cascading DML gates of the same topology is also possibile without intermediate inverter stages when using footed (or headed) gates at each stage. Unfortunetely, this structure causes glitching. However, in this case, unlike the dynamic CMOS logic, the inherent self-restoring property of DML gates ensures the restoration of the logical value.

As stated above, the on-the-fly controllable switching between static and dynamic operation modes offered by DML gates has been efficiently exploited in the design and optimization of energy-efficient and high-performance digital circuits in several previous works [17]-[24]. For instance, the DML approach has been adopted to design carry-look-ahead [17] and carry-skip [18] adders by using a dynamic selection of critical paths according to the input vectors. The DML methodology has been also benchmarked on test chains of basic gates [19], [20]. A novel adder has been proposed in [21] by combining two independent design techniques, such as the DML and the dual-mode addition (DMADD), to achieve low energy, high performance and small area. The benefits of the DML have been also analyzed in [22], [23] on various meaningful benchmarks, such as multiplexers, decoders, comparators, arithmetic circuits. In addition, the combination of the operating characteristics of the DML along with the extended body bias capability offered by an ultra-thin box and body (UTBB) fully-depleted silicon-on- insulator (FD-SOI) technology has been recently evaluated on a NAND–NOR test chain and a 16-bit carry-skip adder [24].

2.4 Criteria for selecting the logic style

Since each logic style inherently features its own advantages and drawbacks, the choice of the most appropriate one is crucial when designing digital circuits. Obviously, such choice depends on the primary requirements in terms of speed, energy consumption and area occupation. Easiness of design, robustness and reliability also play a fundamental role in this task. It is worth pointing out that no single style allows all these specifications to be optimized at the same time. Even more, the selection criteria typically vary from application to application [12].

In general, the complementary static CMOS approach is the most widely used one thanks to the several advantages, mainly in terms of robustness to noise, and scalability with technology and supply voltage. All these features make the static design approach rather easy to be implemented and then particularly amenable to the use of design-automation tools at both logic and circuit level. Anyway, such benefits are achieved at the expense of reduced performance and high area occupation, especially in the design of complex circuits featuring large fan-in gates. The other static logic styles, such as pseudo-NMOS and pass-transistor logics, can represent a good alternative, but only for specific applications. Indeed, pseudo-NMOS allows the design of faster and simpler gates, but at the cost of reduced noise margins and higher static power dissipation. Instead, pass-transistor logic is usually attractive for the implementation of a reduced number of specific circuits, such as multiplexers and XOR-based logic blocks such as adders. On the other hand, dynamic CMOS logics allow designing fast and small digital circuits. However, this occurs at the cost of significantly higher power dissipation as compared to the static design styles and several other design

issues (such as the charge leakage, the charge sharing, the capacitive coupling, and the clock feedthrough) whose impact is typically intensified with technology and voltage scaling, and under process and temperature variations.

Depending upon the requirements of the specific application, a common approach used in the design and optimization of digital circuits is to combine different logical styles with the aim of exploiting the benefits of each single style. In this regard, thanks to its inherent dual operation capability, the DML style represents a powerful tool for digital designers. Figure 2.8 visually illustrates the DML operation space over the energy-performance plan considering that a DML circuit could operate in static or dynamic

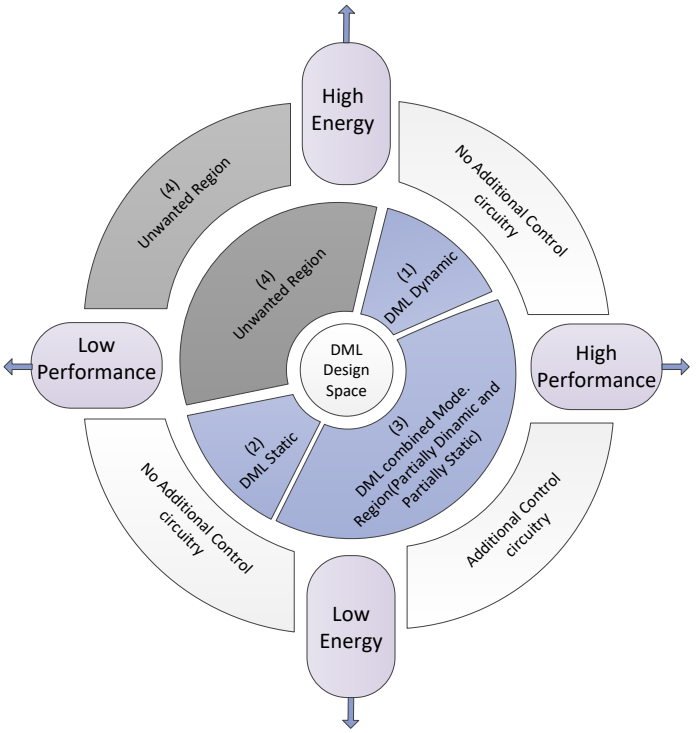

Figure 2.8. DML operation space in the energy-performance plan.

mode, or even in a combined (mixed) mode, i.e. working partly statically and partly dynamically. As stated above, the control of the DML operation mode can be implemented in different ways, e.g. a priori on the basis of specific system requirements and/or designer considerations, or by implementing a smart input-driven control [18]. Obviously, such control can change the operation point of the DML circuit from one area to another one on the energy-performance plan, as shown in Figure 2.8, where region (4) is the undesired area corresponding to low performance and high energy consumption. Region (1) and (2) typically do not require additional control circuitry. Indeed, in these regions, a DML circuit works in one of the two possible operation modes. In region (1), the whole DML network operates in the dynamic mode by toggling clocked transistors in the pre-charge (or pre-discharge) and evaluation phases. This allows high performance at the expense of high energy consumption. On the contrary, in region (2), the entire DML circuit operates in the static mode by disabling clocked pre-charge (or pre-discharge) transistors, thus achieving low energy consumption, but reduced performance. Region (3) is located between the previous regions. In this region, the use of an additional control circuitry can potentially allow one to achieve better energy-performance trade-offs by efficiently combining the DML static and dynamic operation modes. For instance, a smart controller can select which logical paths have to operate in the static mode and which ones in the dynamic mode as a function of the input data [17], [18].

The DML design approach is particularly suitable for applications with a flexible workload, such as in multi-precision arithmetic circuits. The use of on-demand variable-precision circuits is typically addressed to lossy multimedia applications (e.g., audio/video/image processing) where in many cases a reduced precision of arithmetic operations can be tolerated under an acceptable accuracy loss for energy saving. The flexibility inherently

offered by the DML is then very attractive to efficiently trade performance and energy consumption between the operations at different precisions in multi-precision digital circuits. This can be achieved by properly tuning the DML operation mode according to the on-demand precision. More specifically, the DML design must operate in a mixed (or combined) mode, i.e. by employing the static and dynamic mode for lower- and higher-precision operations, respectively. Thereby, on one hand, the use of the dynamic mode in the DML circuit for higher-precision operations (which typically limit the performance of a multi-precision circuit) ensures higher clock frequency as compared to its static CMOS counterpart, obviously at the cost of higher energy consumption. On the other hand, the use of the DML static mode for lower-precision operations assures energy saving with respect to its static CMOS counterpart, thus potentially leading to counterbalance the energy penalty occurring at higher-precision operations. This DML-based design approach of multi-precision arithmetic circuits has been benchmarked in this thesis work on a double-precision (8×8-bit or 16×16-bit) multiplier, whose DML implementation is described in detail in the next chapter along with the comparative results with respect to its static CMOS counterpart.

Chapter 3

3 DML Evaluation and Its Application for a Double-Precision Multiplier

This chapter evaluates the benefits of the DML design approach with respect to the standard static CMOS style. Such analysis is first performed on a flexible circuit benchmark consisting of 10 levels of 11-stage NAND/NOR chains. Then, as case study, the DML style is exploited for the design of a double-precision (8×8-bit or 16×16-bit) carry-save adder (CSA)-based array multiplier to efficiently trade performance and energy consumption between the operations at the two different precisions. In particular, this occurs when the proposed DML multiplier works in a mixed operation mode, i.e. by employing the DML static and dynamic mode for lower and higher precision operations, respectively.

For our particular case, all the circuits discussed in this chapter have been implemented by using low voltage threshold (LVT) transistors of a commercial 1.2V 65-nm low-power CMOS technology and characterized by means of circuit simulations in a commercial computer-aided circuit design tool, such as Cadence Virtuoso environment. However, we must also point out that in other works[24][43] *it can observe that the benefits of Technolgy scales are consistent with the use of the DML logic, that is, the advantage in terms of Energy and performance is maintained with respect to the CMOS logic.*

3.1 DML evaluation in a flexible circuit benchmark

As discussed in the previous chapter, the DML style allows on-the-fly switching at the gate level between static and dynamic operation modes [17]-[24]. In particular, the DML operation modes can be tuned at low-level of design granularity according to application-specific system requirements and/or designer considerations, or automatically by an input-driven control [18]. This means that DML static and dynamic operations can be enabled at the same time in different portions of a circuit, thus potentially allowing one to fully exploit the benefits of the two DML operation modes for better energy-performance trade-offs. Such powerful ability of the DML can be suitably highlighted on a flexible circuit benchmark consisting of 10 levels of 11-stage chains built by alternately cascading 2-bit DML Type-B headless NAND and Type-A footless NOR gates. This circuit can be considered as representative of a generic logic circuit. The schematics of the DML gates and the simulated test circuit are illustrated in Figures 3.1(a)-(c). Figure 3.1(c) also highlights the critical path of the circuit, which crosses all 10 levels as well as the whole bottom NAND/NOR chain. For the sake of

comparison, the same circuit has been implemented in standard static CMOS style by using the typical uniform transistor sizing approach [12], whereas the DML design exploits the unique sizing methodology already discussed in section 2.3. Accordingly, the transistors of the evaluation networks in the DML gates (i.e. the PDN and the PUN in Type-A footless NOR gates and Type-B headless NAND gates, respectively) have been sized as in the standard CMOS circuit, whereas those of the self-restore networks (i.e., the PUN and the PDN in Type-A footless NOR gates and Type-B headless NAND gates, respectively) have been downsized to save energy when using the DML static operation mode.

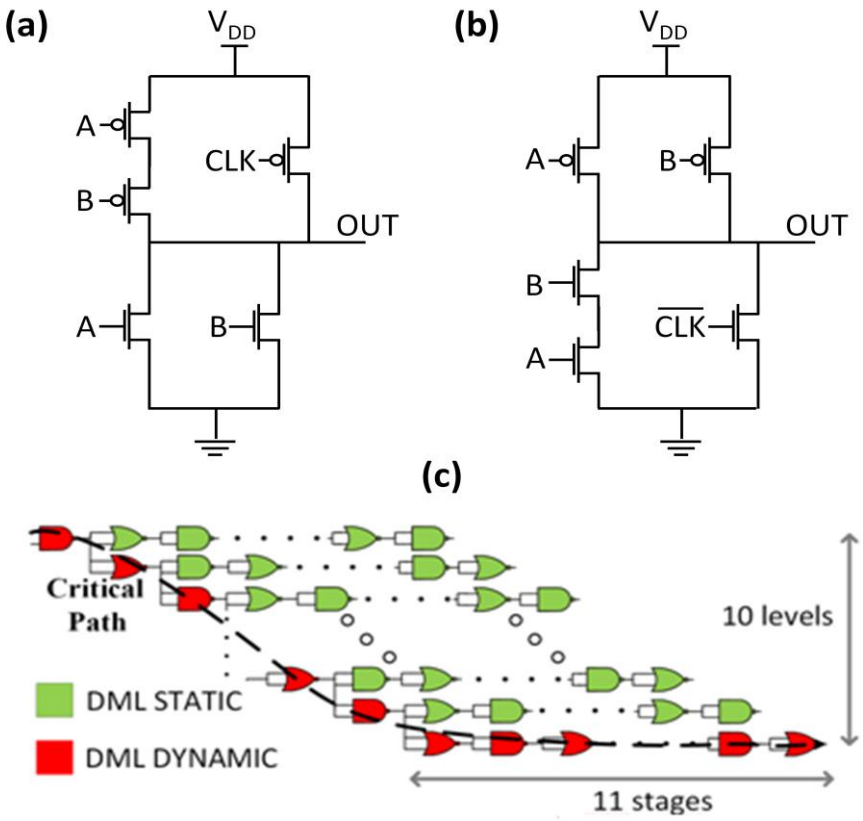

Figure 3.1. Schematic of (a) the 2-bit DML Type-A footless NOR gate, (b) the 2-bit DML Type-A footless NAND gate, and (c) the simulated NAND/NOR test bench

Figures 3.2(a)-(b) report simulation results in terms of delay and energy consumption obtained under supply voltage scaling (with V_{DD} ranging from the nominal voltage of 1.2 V down to 0.6 V) at the nominal process-temperature (PT) corner, i.e. (TT, 27°C), corresponding to typical N/PMOS transistors and operating temperature of 27°C. Reported data refer to both the standard CMOS and the DML implementations of the test chain. For the DML design, static (STAT), dynamic (DYN), and mixed (MIX) operation modes have been analyzed. The latter corresponds to the case when logic

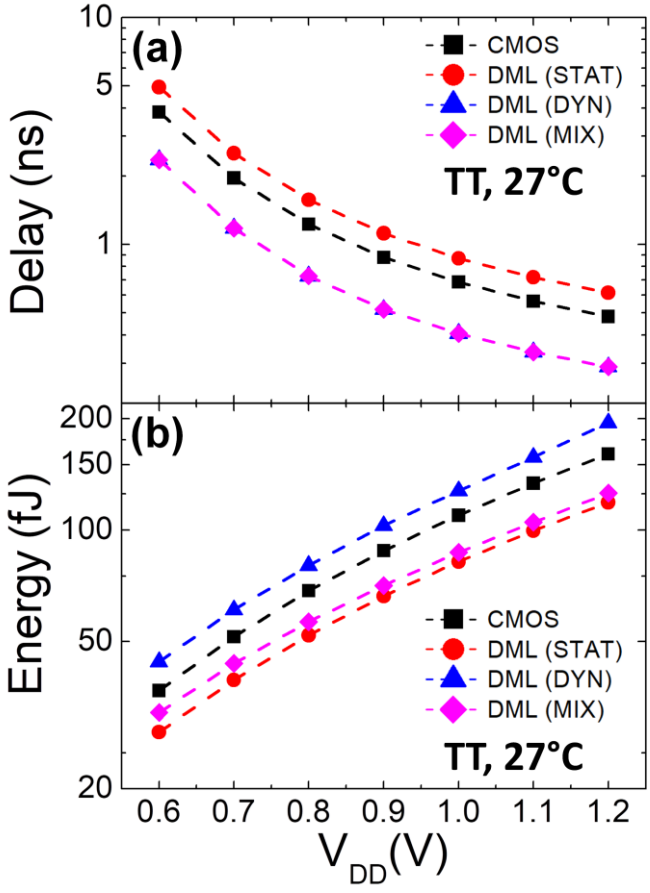

Figure 3.2. (a) Delay and (b) energy results as a function of supply voltage (V_{DD}) at the nominal process-temperature (PT) corner (TT, 27°C) for the simulated DML NAND/NOR test chain working in static (STAT), dynamic (DYN) and mixed (MIX) modes, and its static CMOS counterpart.

gates belonging to the critical path operate in the high-performance dynamic mode, while the remaining gates operate in the low-energy static mode. As expected, the best performance are achieved when using dynamic operations along the critical path, i.e. for the DML circuit operating in DYN and MIX modes, while the DML design operating in STAT mode exhibits the lowest energy consumption, thanks to the downsizing of the self-restore networks in the DML gates, but at the cost of the worst performance. As shown in Figure 3.2, the MIX operation mode allows saving a significant amount of energy as compared to the case when the whole DML circuit operates in the dynamic mode, while keeping the same performance. This because, as stated above, in the MIX mode most of the DML circuit employs static operations to save energy, except for the DML gates along the critical path, which operate in the dynamic mode to ensure high performance. This can be also appreciated in Figure 3.3, where the simulation results are reported in the E-D plan along with the Minimum-Delay-Points (MDPs) and the Minimum-

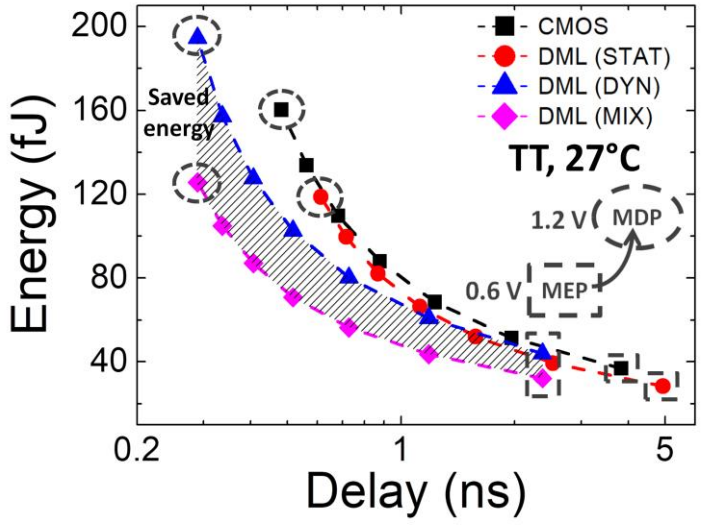

Figure 3.3. Energy-delay trade-off under V_{DD} scaling at the nominal process-temperature (PT) corner (TT, 27°C) for the simulated DML NAND/NOR test chain working in static (STAT), dynamic (DYN) and mixed (MIX) modes, and its static CMOS counterpart.

Energy-Points (MEPs), obviously achieved at the highest V_{DD} (1.2 V) and the lowest V_{DD} (0.6 V), respectively, in the considered V_{DD} range. According to previous results, the DML circuit working in the DYN and MIX modes exhibits the lowest MDP, whereas the lowest MEP is achieved when using the DML STAT mode. Figure 3.3 demonstrates that, when the two DML operation modes are properly mixed within the same architecture, the benefits offered by static and dynamic operations (i.e., energy saving and improved performance, respectively) can coexist, thus resulting into an extended optimization space in the E-D plan and hence better energy-performance trade-offs.

3.2 Case study: a double-precision DML carry-save multiplier

Multipliers play an important role in today's computing systems, especially for DSP and multimedia applications. Indeed, the design and implementation strategies of multipliers substantially contribute to the area, speed, and power consumption of computational intensive digital systems [33]-[38]. In general, a wide variety of computing arithmetic techniques and schemes can be used to implement a multiplication operation [12]. Among these, array multiplier is one of the most popular architectures for implementing parallel multiplication due to its regular layout and simple interconnects [36]. Array multipliers are typically implemented by directly mapping the manual multiplication into hardware, thus involving two steps: the generation of partial products (PPs) and their accumulation. The former consists of parallel logical AND operations between each bit of the multiplier and the multiplicand words. The latter is a multi-operand addition, which is implemented in an array of adder circuits to properly combine the generated PPs. Carry-save adders (CSAs) are widely used in array multipliers to accumulate the partial products [35]-[38]. Then, a final sum operation is needed for the generation of the multiplication result [12].

In this work, a CSA-based 2-stage pipelined multiplier for binary numbers has been designed by using the DML style. The circuit ensures on-demand double-precision (8×8-bit or 16×16-bit) operations at a constant clock frequency, which is typically limited by higher-precision operations in multi-precision digital circuits. As stated in the previous chapter, to efficiently exploit the dual operation capability inherently offered by the DML, the proposed DML multiplier works in a mixed (or combined) mode by tuning its operation mode according to the two different precisions, i.e.

by employing the static and dynamic mode for lower- and higher-precision operations, respectively.

3.2.1 Top-level architecture

Figure 3.4 illustrates the top-level architecture of the proposed 2-stage pipelined multiplier. The first stage consists of a CSA-based array for the generation and the accumulation of the PPs. The second stage produces the final multiplication result with a final sum operation performed by a carry-skip adder, whose structure is similar to that proposed in [24]. The circuit also includes five registers, i.e., two 16-bit registers for the multiplicand (A)

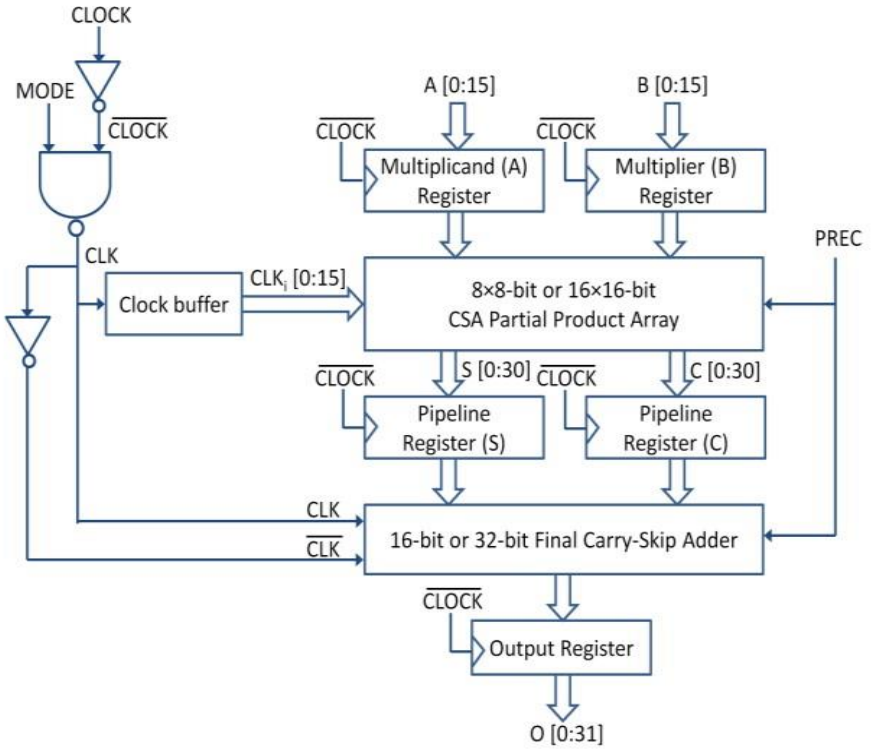

Figure 3.4. Top-level architecture of the proposed DML double-precision multiplier.

and the multiplier (B) inputs, respectively, one 32-bit register for the final

output (O), and two 31-bit intermediate (i.e., between the PP array and the final adder stage) pipeline registers.

To ensure the on-the-fly switching at the gate level between DML static and dynamic operation modes, a low-complexity extra clock control circuitry has been added in the proposed circuit. As shown in the left-upper part of Figure 3.4, this block receives as inputs the external clock signal (CLOCK) and a selection mode signal (MODE) to generate the internal clock signal (CLK) and its flipped version for DML gates. In this way, when MODE is low, it generates a high (low) CLK for Type-A (Type-B) DML gates to disable pre-charge (pre-discharge) clocked transistors, thus enabling the static operation mode. Conversely, when MODE is high, the dynamic operation mode is enabled by generating a CLK signal (and its flipped version) for properly toggling clocked transistors of DML gates in the pre-charge (or pre-discharge) and evaluation phases.

As illustrated in Figure 3.4, a 16-stage clock buffer tree has been also introduced to generate 16 delayed clock signals (CLK_i) (i.e., one for each PP array row), with the aim of arranging an appropriate timing during dynamic operations. In particular, a proper design of the clock buffer tree is mandatory to ensure glitch-free and safe operations of the PP array at the two different precisions when the dynamic mode is enabled, especially at higher precision that involves longer signal propagation paths. A further control signal, namely PREC, has been then used as input for both the PP array and the final adder to set the on-demand operation precision. More specifically, an active low PREC signal translates into a higher-precision (i.e., 16×16-bit) multiplication, which requires a 32-bit final sum operation. Alternatively, a high PREC signal enables two contemporary 8×8-bit

multiplication operations, which require two independent 16-bit final sum operations.

3.2.2 The CSA-based partial product array

Figure 3.5 illustrates the general scheme of the 16×16-bit CSA array used for the generation and the accumulation of the PPs. It consists of three elementary blocks: AND gate, modified half adder (MHA), and modified full adder (MFA). Considering *i* as the row index and *j* as the diagonal column index, both ranging from 0 to 15, each of these blocks receives the *j*-th bit of the multiplicand (A) and the *i*-th bit of the multiplier (B) to produce the corresponding (*i, j*) partial product by a logical AND operation. In addition to such operation, MHA and MFA blocks also perform an addition operation to accumulate the generated PPs. To properly manage double-precision operations, all the elementary blocks are implemented in two different versions: (*i*) one that receives the additional PREC control signal to disable some PP generations when operating at lower precision, and (*ii*) the other one without such control. As shown in Figure 3.5(a), the former is employed for the blocks belonging to the left-upper and right-lower quadrants of the PP array, which correspond to PPs related to most significant bits (MSBs) of A and least significant bits (LSBs) of B, and to LSBs of A and MSBs of B, respectively. On the contrary, the second version without PREC control is used for the blocks belonging to the right-upper and left-lower quadrants of the PP array, which instead correspond to PPs related to LSBs of both A and B, and to MSBs of A and MSBs of B, respectively. As a consequence, when a lower precision is required (i.e.,

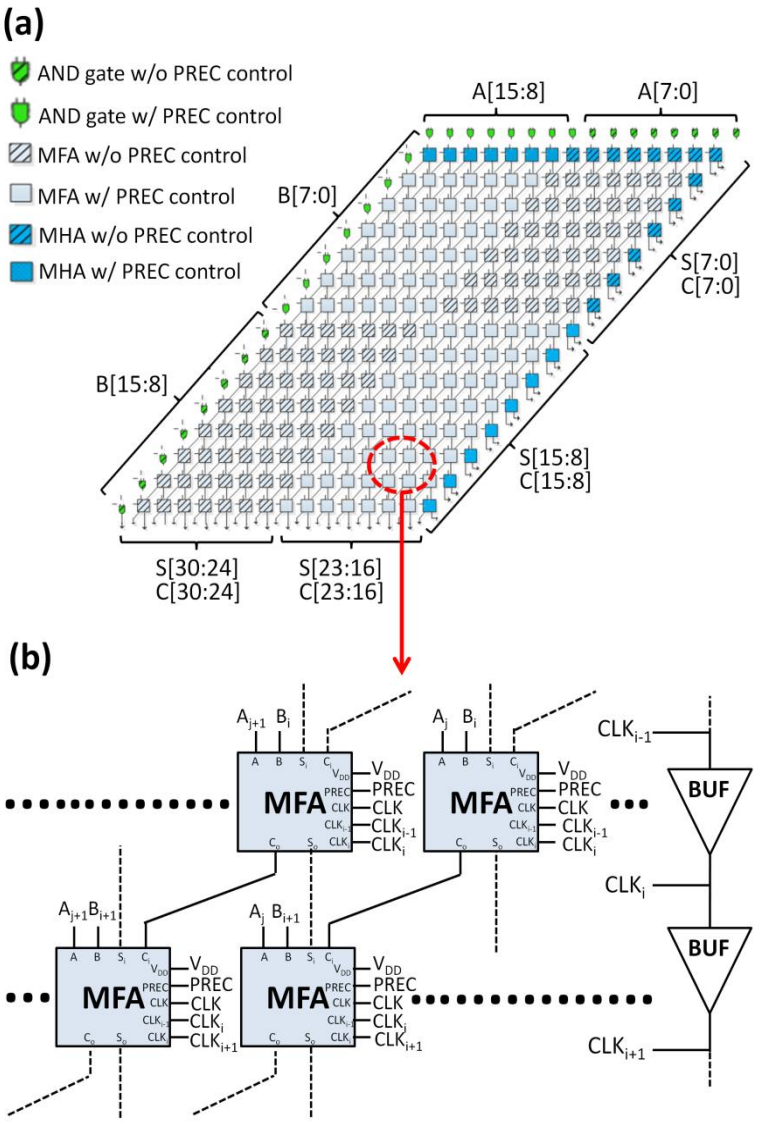

Figure 3.5. 16×16-bit CSA-based array for the generation and the accumulation of the partial products: (a) block diagram and (b) detailed sketch of the array with the clock buffer tree.

PREC is high), the generation of PPs is disabled in the blocks of the left-upper and right-lower quadrants of the array, thus enabling two contemporary 8×8-bit multiplications. Conversely, when PREC is low, one

higher-precision (i.e., 16×16-bit) multiplication is performed. The interconnections among MFA blocks belonging to two subsequent rows and two subsequent diagonal columns of the array are highlighted in Figure 3.5(b) along with the detail of the clock buffer tree. Note that, to ensure the correct timing during dynamic mode operations, for each *i*-th row the adder blocks receive three different versions of the clock signal: the internal clock signal (CLK), the delayed clock signal corresponding to the previous row (CLK_{i-1}), and the delayed clock signal of the corresponding *i*-th row (CLK_i).

Figure 3.6 illustrates in detail the adopted DML implementation for the three elementary blocks of the CSA PP array. The reported schematics refer to the blocks that receive the additional PREC signal to control the operation precision of the multiplier. As shown in Figure 3.6(a), the PP generation is implemented through a 2-bit DML Type-A footed NAND gate followed by a 2-bit standard CMOS NOR gate, which is driven by the output of the NAND gate and the PREC signal. Accordingly, when PREC is high (i.e., a lower precision operation is required), the PP generation is disabled (i.e., $Y_o = 0$). Obviously, the 2-bit standard CMOS NOR gate is replaced by a standard CMOS inverter in the AND gates without PREC control belonging to the right-upper and left-lower quadrants of the array. Figure 3.6(b) shows the transistor-level design of the MHA block. It consists of one DML AND gate (with or without the PREC control depending on the placement in the array of Figure 3.5(a)) and a DML Type-A footed HA circuit with two output standard CMOS inverters to produce the output sum (S_o) and carry (C_o) bits. Therefore, this HA circuit adds the PP (I_i) bit coming from the internal AND gate with the input (Y_i) bit coming from the previous row. Finally, the schematic of the MFA block is shown in Figure 3.6(c). Similar

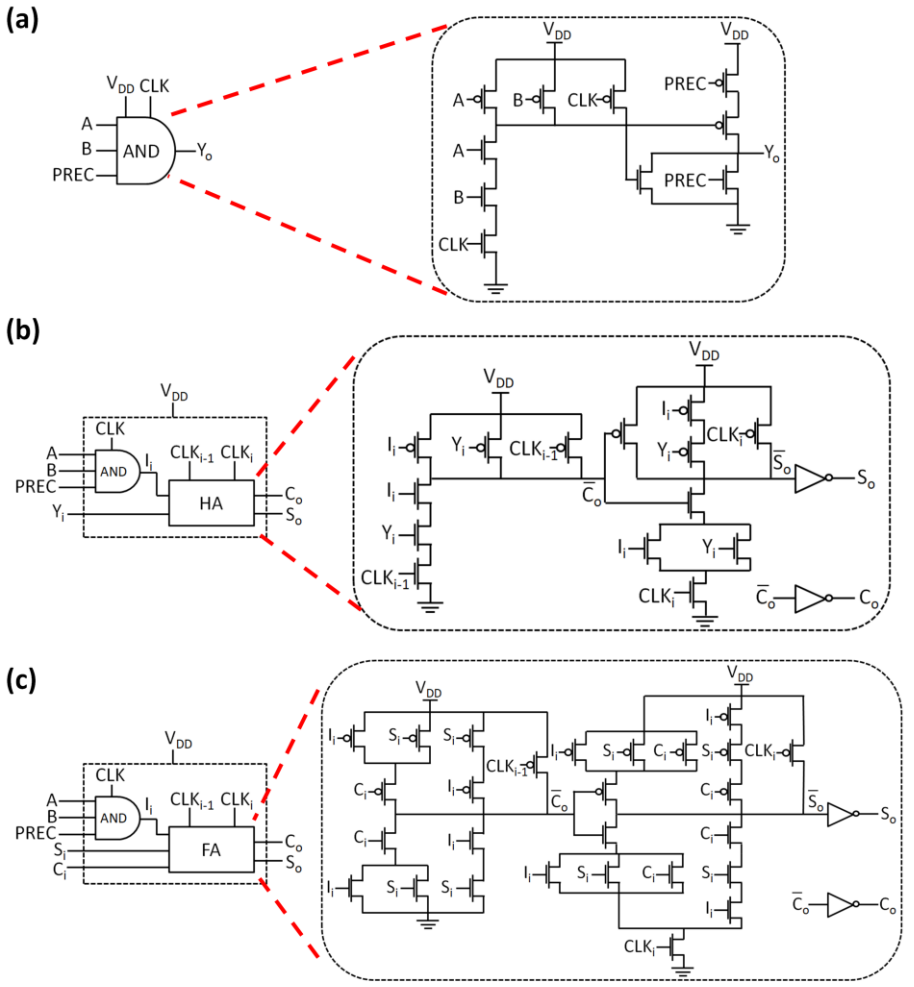

Figure 3.6. DML implementation of the three elementary blocks with PREC control in the 16×16-bit PP array: (a) AND gate, (b) modified half adder (MHA) with the schematic of the HA subcircuit, and (c) modified full adder (MFA) with the schematic of the FA subcircuit.

to the MHA, it includes one DML AND gate to generate the internal PP (I_i) bit, and a DML Type-A FA circuit that adds I_i with the input sum (S_i) and carry (C_i) bits coming from the previous row, according to Figure 3.5(b). To avoid glitches during DML dynamic operations, note that the carry and sum generation portions of the FA circuit are implemented in Type-A footless

and footed topologies, respectively. Again, two output standard CMOS inverters are used to generate the output S_o and C_o bits.

3.2.3 The final carry-skip adder

Adders are fundamental modules in the design of computing systems, such as DSP architectures, microcontrollers and microprocessors [39]-[41]. Depending upon the application-specific requirements and/or the chosen logic style, different schemes can be selected to implement an addition operation [12]. Among the several topologies, the carry-skip design is typically regarded as a good alternative in terms of area occupation, performance and easiness of design [24]. Accordingly, the second stage of the proposed DML pipelined multiplier, aimed at performing the final sum operation and then generating the multiplication result, has been implemented by a double-precision (16/32-bit) carry-skip adder [24] consisting of an 8-stage chain. Figure 3.7(a) illustrates its top-level architecture with the detailed sketch at the middle of the chain. In particular, each stage is composed by a 4-bit ripple-carry adder (RCA), a 4-bit skip logic (SL) block, and a 2-bit multiplexer (MUX). The 4-bit RCA blocks receive the corresponding output sum (S) and carry (C) bits from the PP array, and the input carry (C_{in}) bit from the previous stage to generate the output carry (C_{out}) bit for the subsequent stage, the final output (O) bits, and the propagate (P) bits. The latter are inputted to the SL block to produce the select signal (SEL) for the MUX. Note that the control of double-precision operations is implemented by inserting an additional 2-bit MUX at the middle of the chain, as shown in Figure 3.7(a). Such MUX receives as inputs a low signal and the C_{out} bit from the fourth stage of the chain (i.e., C_{out3} in

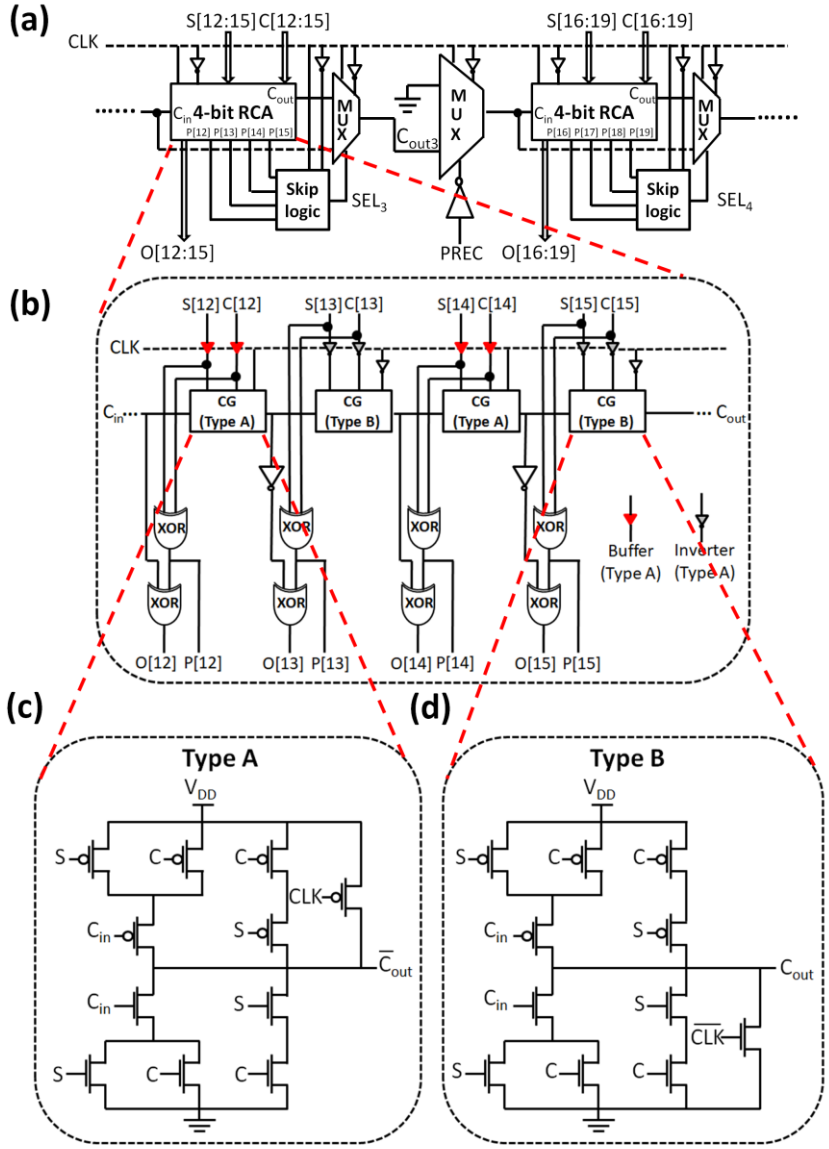

Figure 3.7. Design of the final 16/32-bit carry-skip adder: (a) detail of the top-level architecture at the middle of the chain, (b) block diagram of the 4-bit ripple-carry adder (RCA), and DML implementation of the (c) Type-A and (d) Type-B carry-generator (CG) blocks.

Figure 3.7(a)), while its select signal is the flipped PREC signal. As a consequence, when PREC is low (i.e., for higher-precision operations), the output of this additional MUX follows C_{out3}. This allows the carry

propagation to the subsequent stage, thus translating into a 32-bit final sum operation. On the other hand, a high PREC signal disables the carry propagation at the middle of the chain to perform two independent 16-bit final sum operations. The detailed scheme of the 4-bit RCA block is illustrated in Figure 3.7(b). It consists of a 4-stage chain, where each stage is composed by a carry-generator (CG) block and some basic gates, such as buffers, inverters, and XORs. It is worth pointing out that the stage design depends on its position along the chain. In particular, the first and the third stages are implemented with two input buffers (consisting of a DML Type-A footed inverter followed by a standard CMOS inverter) and a DML Type-A footless CG block, whose schematic is shown in Figure 3.7(c). Conversely, the second and the fourth stages employ two input DML Type-A footed inverters and a DML Type-B headless CG block, whose schematic is shown in Figure 3.7(d). In both cases, output inverters and XOR gates are designed in standard static CMOS logic. Note that the proposed design for the 4-bit RCA block allows avoiding cascading issues during dynamic operations. This is achieved thanks to the use of the standard CMOS inverter in the input buffers of the first and the third stages, and by alternatly cascading Type-A and Type-B DML CG blocks along the RCA chain.

3.2.4 Simulation results and discussion

This section reports and discusses simulation results in terms of performance and energy consumption of the proposed double-precision (8×8-bit or 16×16-bit operations) DML multiplier operating in static (STAT), dynamic (DYN) and mixed (MIX) modes in comparison with its standard static CMOS counterpart. More specifically, STAT and DYN modes refer to the cases when using static and dynamic operations,

respectively, for both lower and higher precision in the DML design. On the contrary, the MIX mode consists of combining the DML operation modes at the different precisions, i.e. when using dynamic operations at the higher precision and static operations at the lower precision. Regarding the transistor sizing, the typical uniform sizing approach has been used for the standard static CMOS design [12], whereas the DML multiplier again exploits the unique sizing methodology. Accordingly, the evaluation networks of the DML gates have been sized following the same approach used for the static CMOS circuit, whereas the self-restore networks (i.e., the PUN in Type-A DML gates and the PDN in Type-B DML gates) have been properly downsized to save energy in the static operation mode.

The speed/energy results of the DML and CMOS multipliers at the two different precisions have been firstly evaluated under V_{DD} scaling (from 1.2 V down to 0.6 V) at the nominal process-temperature (PT) corner, i.e. (TT, 27°C). A set of 500 randomly distributed input vectors has been used to estimate the average energy per operation (E_{op}). Such analysis has been also performed for different PT corners [42]. The sensitivity to random mismatch variations at the nominal PT corner has been also evaluated for worst-case delay operations [42] at the two different precisions and two V_{DD}s (1.2 V and 0.8 V) through 1000-run Monte-Carlo simulations.

Figure 3.8 shows comparative results in terms of operation frequency and average E_{op} versus V_{DD} (at the nominal PT corner) for the operations at the two different precisions. Plotted data refer to the DML multiplier working in STAT or DYN mode for both lower- and higher-precision operations, and its static CMOS counterpart. As expected, the use of the DYN mode in the DML multiplier assures the highest operation frequency in all cases. In

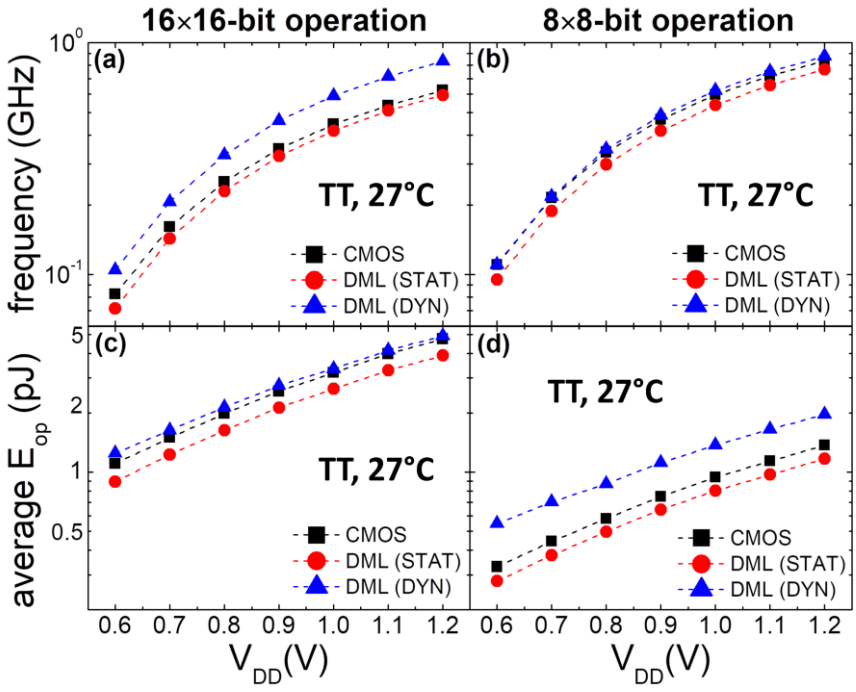

Figure 3.8. Operation frequency and average energy per operation (E_{op}) as a function of supply voltage (V_{DD}) at the nominal process-temperature (PT) corner (TT, 27°C) for the operations at the two different precisions n the DML multiplier working in static (STAT) and dynamic (DYN) modes, and its static CMOS counterpart.

particular, when compared to the static CMOS design, the frequency boost of the DML circuit working in the DYN mode is up to 25% at 1.2 V for higher-precision operations. As shown in Figure 3.8(a), such speed advantage slightly drops with the V_{DD} scaling down to 21% at 0.6 V. This because, according to what discussed in the previous chapter, dynamic circuits typically most suffer from voltage scaling with respect to static ones. The speed gain of the DML circuit working in the DYN mode significantly decreases at lower-precision operations, as shown in Figure 3.8(b). This behavior can be explained as follows. As stated above, when the DML multiplier operates in the DYN mode, the clock buffer tree plays a fundamental role to ensure glitch-free, safe and fast dynamic operations by

managing an appropriate timing in the PP array. Moreover, the proposed DML multiplier has to ensure on-demand double-precision operations at the same clock frequency, which is obviously limited by higher-precision operations involving longer signal propagation paths. Therefore, the design of the clock buffer tree has been strictly constrained by higher-precision dynamic operations. As a consequence, the clock buffer tree imposes a similar timing during dynamic operations at the two different precision, thus translating into a very small increase of the operation frequency for lower-precision dynamic operations with respect to those at the higher precision, as it can be observed from the comparison of Figures 3.8(a) and (b). On the contrary, the operation frequency of both CMOS circuit and DML circuit working in STAT mode notably increases at the lower precision, thus leading to the decrease of the speed advantage of the DYN mode for lower precision-operations, as shown in Figure 3.8(b). Obviously, the higher operation frequency of the DML circuit working in the DYN mode is achieved at the expense of higher energy consumption (which can be mainly ascribed to the extra clock power), as shown in Figures 3.8(c) and (d). As compared to the static CMOS circuit, the energy penalty of the DYN mode ranges from 4% at 1.2 V up to 12% at 0.6 V for higher-precision operations (7% on average). Such energy penalty strongly increases for lower-precision operations, ranging from 30% at 1.2 V up to 40% at 0.6 V (34% on average), as the clock energy consumption during lower-precision dynamic operations remains practically unchanged with respect to higher-precision operations. Conversely, the use of the STAT mode in the DML multiplier ensures the lowest energy consumption over the whole considered V_{DD} range for both operations at the two different precisions. When compared to the CMOS circuit, the energy saving achieved by the DML circuit working in the STAT mode is quite constant with the V_{DD}, i.e. 18% (15%) on average at the higher

(lower) precision. This occurs at the cost of a performance reduction of 8% (11%) on average at the higher (lower) precision for, as shown in of Figures 3.8(a) and (b).

The previous analysis has been carried out for different PT corners to cover a wide range of possible operating conditions. In addition to the nominal PT corner, the other two considered PT corners involve fast N/PMOS transistors at T = -25°C (FF, -25°C) and slow N/PMOS transistors at T = 125°C (SS, 125°C), respectively. Accordingly, Figures 3.9(a)-(f) shows operation frequency and average E_{op} results versus V_{DD} assuming 50% of operations at higher precision and 50% of operations at lower precision for the DML circuit operating in STAT, DYN and MIX modes, and its CMOS counterpart at the three considered PT corners. As shown in Figures 3.9(a)-(c), the use of the DYN mode in the DML design for both operations at the

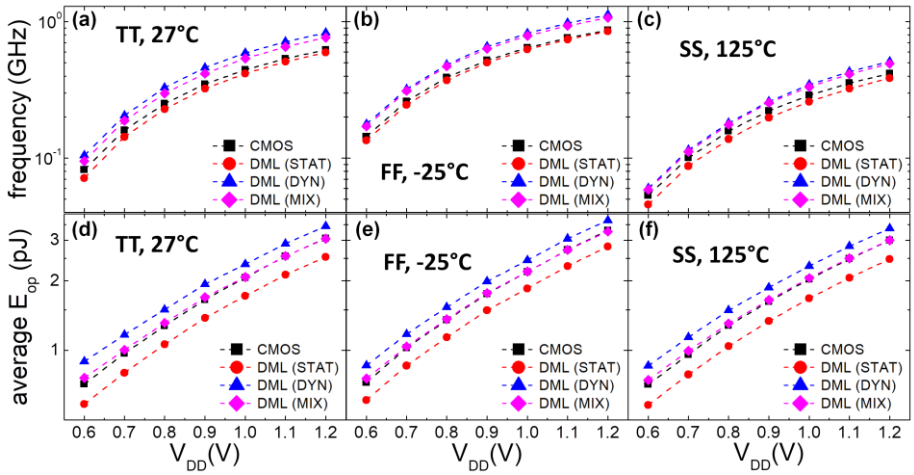

Figure 3.9. Operation frequency and average energy per operation (E_{op}) as a function of supply voltage (V_{DD}) at the three considered process-temperature (PT) corners for the DML multiplier working in static (STAT), dynamic (DYN) and mixed (MIX) operation modes, and its CMOS counterpart when assuming 50% of operations at higher precision and 50% of operations at lower precision.

two different precisions provides the highest operation frequency in all the

simulated conditions. When compared to the CMOS circuit, the DML multiplier working in the DYN mode allows a frequency boost of 24% on average at the (TT, 27°C) corner. According to the results of Figure 3.8, for a given PT corner, the speed improvement of the DYN mode drops when decreasing V_{DD} due to the higher sensitivity of dynamic operations to the voltage scaling. In addition, the frequency boost of dynamic operations also decreases down to 21% and 15% on average at the (FF, -25°C) corner and the (SS, 125°C) corner, respectively, thus demonstrating the higher sensitivity of dynamic circuits even to PT variations. Again, such frequency boost occurs at the expense of the highest energy consumption, as shown in Figures 3.9(d)-(f). According to results of Figure 3.8, the energy penalty of the DYN mode with respect to the static CMOS design increases with the V_{DD}, ranging from 11% at 1.2 V up to 20% at 0.6 V (15% on average) for the (TT, 27°C) corner, from 10% at 1.2 V up to 16% at 0.6 V (12% on average) for the (FF, -25°C) corner, and from 11% at 1.2 V up to 17% at 0.6 V (14% on average) for the (SS, 125°C) corner. On the contrary, the use of the STAT mode in the DML circuit ensures the lowest energy consumption at the cost of the worst performance in all the simulated conditions. As compared to the CMOS circuit, the energy saving (performance reduction) of the DML multiplier working in the STAT mode is on average of 17% (8%) at the (TT, 27°C) corner, 15% (4%) at the (FF, -25°C) corner, and 18% (11%) at the (SS, 125°C) corner. From Figures 3.9(a)-(f) we can observe that the adoption of the MIX mode in the proposed double-precision DML multiplier allows efficiently exploiting the benefits offered by the two different DML operation modes. In fact, on one hand, the use of dynamic operations only at higher precision assures in the MIX mode a higher operation frequency with respect to both the static CMOS counterpart and the DML circuit working in the STAT mode for both double-precision

operations in all the simulated operating conditions, as shown in Figures 3.9(a)-(c). This because, unlike the other cases, the operation frequency of the DML multiplier working in the MIX mode is limited by lower-precision operations, for which the STAT mode is employed. As a consequence, as compared to the static CMOS circuit, this leads to a frequency boost that ranges from 19% at 1.2 V down to 14% at 0.6 V (16% on average) for the (TT, 27°C) corner, from 19% at 1.2 V down to 16% at 0.6 V (18% on average) for the (FF, -25°C) corner, and from 15% at 1.2 V down to 9% at 0.6 V (12% on average) for the (SS, 125°C) corner. On the other hand, the energy penalty owing to the use of dynamic higher-precision operations is counterbalanced in the MIX mode by the energy saving achieved at lower precision thanks to the adoption of static operations. As shown in Figures 3.9(d)-(f), this translates into a similar average E_{op} of the DML multiplier working in the MIX mode as compared to the static CMOS circuit, and into a significant energy saving with respect to the DML circuit operating in the DYN mode. The benefits achieved by the DML multiplier working in the MIX mode can be also appreciated in Figures 3.10(a)-(c), where the simulation results of the performed PT corner analysis under V_{DD} scaling (from 1.2 V down to 0.6 V) are plotted on the energy-frequency plan. In these graphs, the MDPs corresponding to the highest V_{DD} (1.2 V), and the MEPs corresponding to the lowest V_{DD} (0.6 V) are also highlighted at the three different PT corners for the the DML circuit operating in STAT, DYN, and MIX modes, and its static CMOS counterpart.

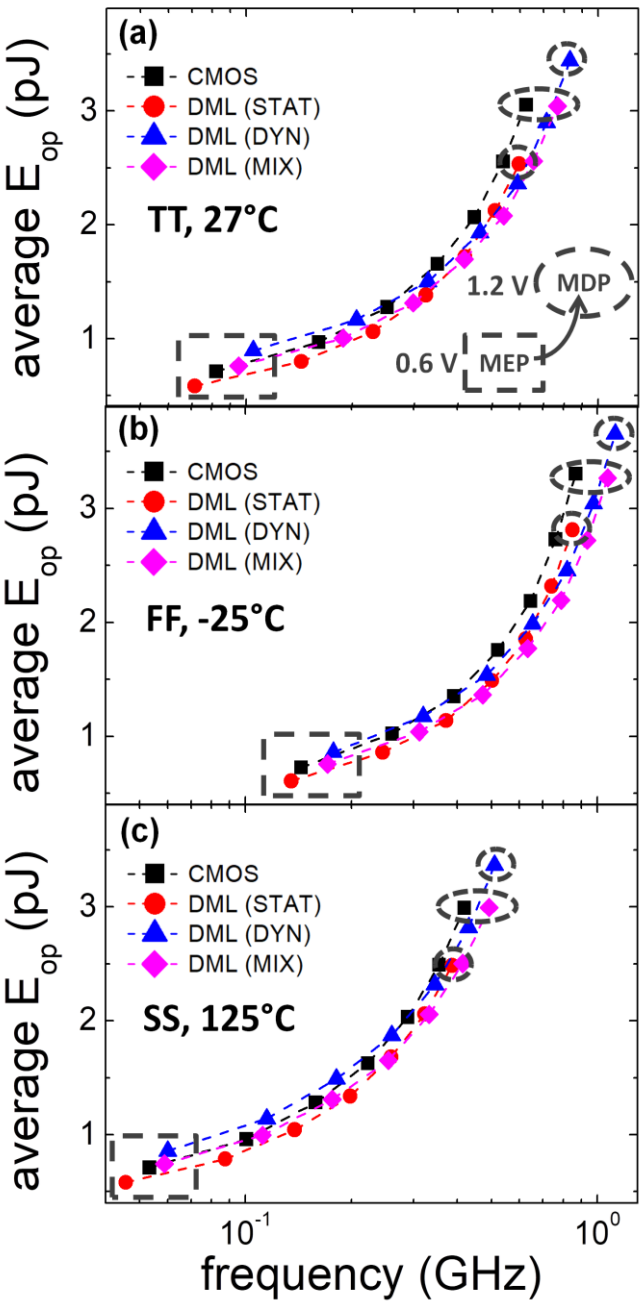

Figure 3.10. Simulation results of the PT corner analysis under V_{DD} scaling on the energy-frequency plan for the DML multiplier working in static (STAT), dynamic (DYN) and mixed (MIX) operation modes, and its CMOS counterpart, when assuming 50% of operations at higher precision and 50% of operations at lower precision.

According to previous results, for a given PT corner, the DML STAT mode allows for the lowest MEP, whereas the DML DYN mode leads to the lowest MDP. When the two DML operation modes are properly combined for the operations at the two different precision, both DML static and dynamic benefits (i.e. improved speed and energy saving, respectively) can be efficiently exploited, thus achieving a more extended optimization space. As a matter of fact, overall, the DML multiplier working in the MIX mode results into (*i*) higher performance (i.e. lower MDP) with a similar energy consumption (i.e. similar MEP) with respect to the static CMOS counterpart, (*ii*) higher energy consumption (i.e. higher MEP) but significantly higher performance (i.e. much lower MDP) with respect to the DML circuit working in the STAT mode, and (*iii*) slightly lower performance (i.e. slightly higher MDP) but significantly lower energy consumption (i.e. much lower MEP) with respect to the DML circuit working in the DYN mode. These results translate into a better energy/performance trade-off when using the DML MIX mode, as shown in Figures 3.11(a)-(c) where the EDP versus V_{DD} is plotted at the three considered PT corners. In particular, when compared to the static CMOS circuit, the DML multiplier working in the MIX mode achieves an EDP reduction at the (TT, 27°C) corner ranging from 19% at 1.2 V down to 8% at 0.6 V (15% on average), as shown in Figure 3.11(a). The EDP achieved by using the MIX mode is also better than that obtained when using the DYN mode or the STAT mode in the DML circuit (i.e. 11% and 10% on average, respectively, at the nominal PT corner). As shown in Figures 3.11(b) and (c), such benefit is confirmed at (FF, -25°C) and (SS, 125°C) corners where, as compared to the static CMOS circuit, the DML MIX operation mode provides an average EDP reduction of 17% and 10%, respectively.

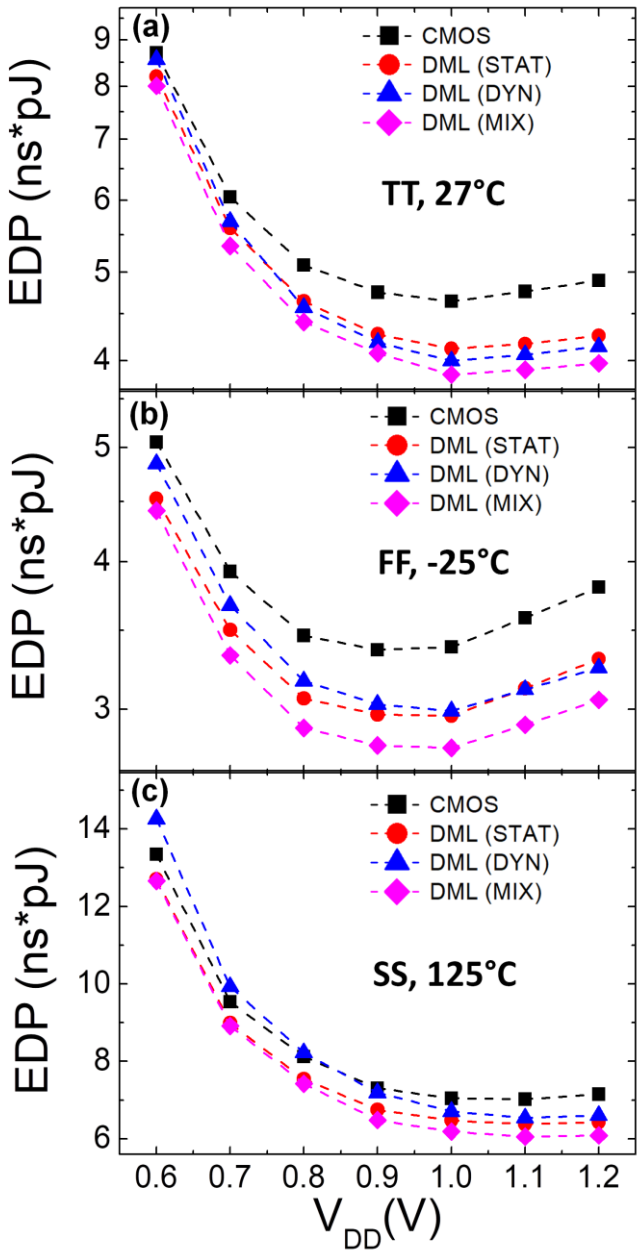

Figure 3.11. Energy-delay product (EDP) as a function of supply voltage (V_{DD}) at the three considered process-temperature (PT) corners for the DML multiplier working in static (STAT), dynamic (DYN) and mixed (MIX) operation modes, and its CMOS counterpart, when assuming 50% of operations at higher precision and 50% of operations at lower precision.

According to previous results, note also in Figure 3.11 that the EDP of the DML circuit working in the DYN mode significantly degrades at low V_{DD} with respect to the other cases.

Table 3.1 summarizes the comparative results in terms of worst-case delay, average E_{op}, and EDP obtained at the considered PT corners for two different V_{DD}s (1.2 V and 0.8 V). It is worth noting that, for a given V_{DD}, the DML circuit working in the DYN mode shows the highest spread in percentage in terms of delay and hence EDP along the considered PT corners, thus confirming the higher sensitivity of dynamic circuits to the process variability. For instance, at V_{DD} = 1.2 V, the delay (EDP) related to the use of the DYN mode spreads from 0.89 ns (3.25 ns*pJ) at the (FF, -25°C) corner up to 1.96 ns (6.60 ns*pJ) at the (SS, 125°C) corner, thus corresponding to a spread in percentage of 79% (72%) with respect to the average value obtained along the considered PT corners. Conversely, the

Table 3.1. Summary results at (TT, 27 °C), (FF, -25°C), and (SS, 125°C) corners.

	V_{DD} = 1.2 V								
	Delay (ns)			**E_{op} (pJ)**			**EDP (ns*pJ)**		
	TT	**FF**	**SS**	**TT**	**FF**	**SS**	**TT**	**FF**	**SS**
CMOS	1.60	1.15	2.39	3.06	3.30	2.99	4.90	3.81	7.15
DML (STAT)	1.68	1.18	2.59	2.54	2.81	2.48	4.26	3.31	6.43
DML (DYN)	1.20	0.89	1.96	3.44	3.65	3.37	4.14	3.25	6.60
DML (MIX)	1.31	0.93	2.03	3.04	3.27	2.99	3.97	3.06	6.09
	V_{DD} = 0.8 V								
	Delay (ns)			**E_{op} (pJ)**			**EDP (ns*pJ)**		
	TT	**FF**	**SS**	**TT**	**FF**	**SS**	**TT**	**FF**	**SS**
CMOS	3.98	2.56	6.33	1.29	1.35	1.28	5.13	3.46	8.11
DML (STAT)	4.37	2.69	7.24	1.06	1.14	1.04	4.64	3.06	7.54
DML (DYN)	3.05	2.06	5.52	1.50	1.55	1.48	4.58	3.19	8.18
DML (MIX)	3.35	2.12	5.67	1.32	1.37	1.31	4.42	2.90	7.43

CMOS circuit is the most robust against PT variations, owing to the use of static operations and larger devices with respect to the DML design. As a matter of fact, at V_{DD} = 1.2 V, the delay (EDP) of the CMOS circuit spreads from 1.15 ns (3.81 ns*pJ) at the (FF, -25°C) corner up to 2.39 ns (7.15 ns*pJ) at the (SS, 125°C) corner, thus corresponding to a spread in percentage of 72% (63%) with respect the average value obtained along the considered PT corners.

Finally, the tolerance to random intra-die (mismatch) variations has been evaluated at the nominal PT corner for the two different precision operations in the proposed DML multiplier working in STAT and DYN modes, and its CMOS counterpart. Accordingly, Figure 3.12 illustrates the energy for the worst-case delay operation ($E_{w.c.d.}$) versus delay spreads obtained at two different V_{DD}s (1.2 V and 0.8 V). Mean (μ) and standard deviation (σ) energy and delay values are also reported in Figure 3.12. As expected, the use of the DYN mode in the DML implementation leads to the best mean delay values in all the simulated conditions at the cost of the highest mean $E_{w.c.d.}$ values. Conversely, the DML multiplier working in the STAT mode shows the lowest mean $E_{w.c.d.}$ values and the highest mean delay values. In addition, the DML multiplier exhibits a higher delay variability $(\sigma/\mu)_d$ with respect to the CMOS circuit for both operations at the two different precisions due to the use of smaller transistors, whereas the highest energy variability $(\sigma/\mu)_e$ is observed in the DML circuit working in the STAT mode. Again, we can observe that, as compared to its CMOS counterpart, the performance improvement (energy penalty) of the DML circuit working in the DYN mode decreases (increases) when decreasing the V_{DD}.

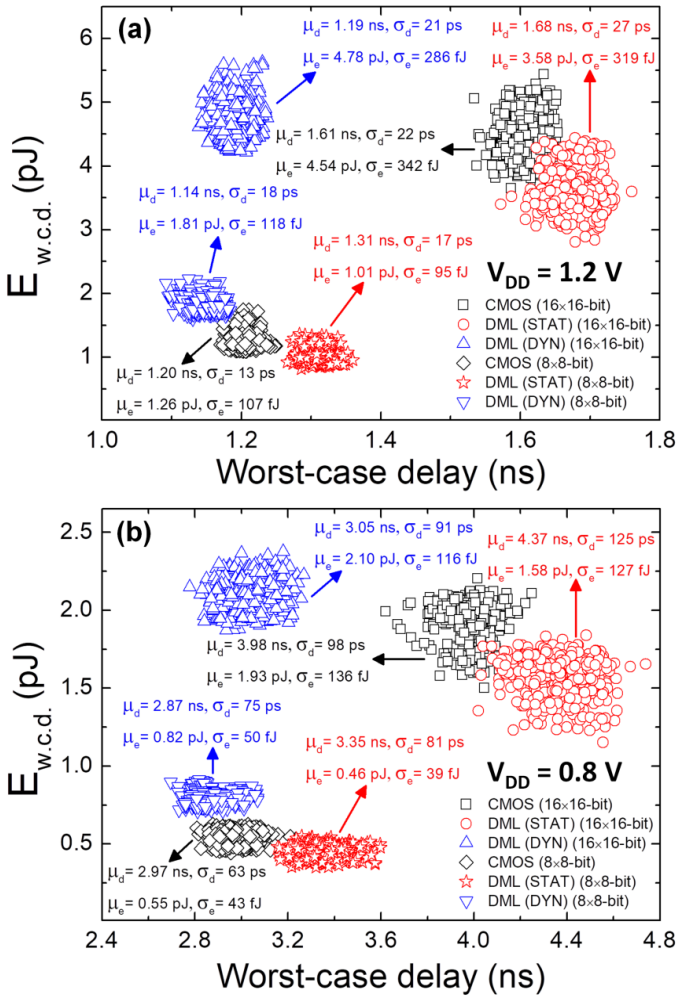

Figure 3.12. Monte Carlo results in terms of energy consumption of the worst-case delay operation ($E_{w.c.d.}$) versus worst-case delay at the nominal process-temperature (PT) corner (TT, 27 °C) for the two different precision operations in the DML multiplier working in static (STAT) and dynamic (DYN) operation modes, and its CMOS counterpart: (a) V_{DD} = 1.2 V and (b) V_{DD} = 0.8 V.

Chapter 4

4 Conclusions

This PhD thesis has been mainly focused on the evaluation of some solutions to be implemented in the design and optimization of digital circuits, in particular when facing lossy multimedia applications (e.g., audio/video/image processing) where reducing the precision of arithmetic operations can be tolerated under the acceptable accuracy loss. In this case, the design of multi-precision arithmetic circuits along with the use of a new logic design technique, namely the Dual Mode Logic (DML), can be potentially very attractive to achieve more energy-efficient computing platforms while keeping high performance, also thanks to the unique ability of the DML to switch on-the-fly at the gate level between static and dynamic operation modes.

In this work, the benefits coming from the flexibility inherently offered by the DML has been firstly evaluated on a flexible circuit benchmark consisting of 10 levels of 11-stage NAND/NOR chains. In this circuit, the DML design takes advantage of its dual operation capability that allows working in a combined (mixed) operation mode, i.e. operating at the same time partly statically and partly dynamically, thus leading to fully exploit the benefits of the two DML operation modes for better energy-performance trade-offs. Then, as main case study, the DML approach has been used to design a double-precision (8×8-bit or 16×16-bit) carry-save adder (CSA)-based array multiplier in a commercial 65-nm low-power CMOS technology. Here, the DML dual operation ability is exploited to efficiently trade performance and energy consumption between the operations at the two different precisions. In particular, this occurs by properly tuning the DML operation mode according to the two different precisions. This means that the proposed DML multiplier works in a mixed (combined) operation mode, i.e. using the DML static and dynamic modes for lower- and higher-precision operations, respectively. Simulation results under supply voltage scaling at the nominal process-temperature (PT) condition have shown that the use of such mixed operation mode in the DML multiplier leads to 16% and 23% gains on average in speed when compared to the standard static CMOS counterpart and the DML circuit working in the static mode, respectively. At the same time, as compared to the standard CMOS circuit, the DML multiplier working in the mixed mode exhibits a similar energy consumption, which corresponds to 13% reduction on average with respect to the DML circuit operating in the dynamic mode. Therefore, when compared to its standard CMOS counterpart, the DML multiplier working in the mixed mode achieves an average improvement in the energy-delay product (EDP) of 15%, which is also better than that obtained when using

the DML static or the dynamic mode for both operations at the two different precisions. It has been also shown that such benefits are maintained over a wide range of process-voltage-temperature (PVT) variations.

Bibliography

[1] J. de Boeck, "IoT: the impact of things," in *Symposium on VLSI Technology Dig. Tech. Papers*, Kyoto, Japan, 2015, pp. T82–T83.

[2] M. Alioto (Ed.), "Enabling the Internet of Things," Springer, Cham, Switzerland, 2017.

[3] S. Jain, L. Lin, M. Alioto, "Dynamically adaptable pipeline for energy-efficient microarchitectures under wide voltage scaling," *IEEE J. Solid State Circ.*, vol. 53, no. 2, pp. 632–641, 2018.

[4] G. E. Moore, "Cramming more components onto integrated circuits," *IEEE Solid-State Circuits Soc. Newsl.*, vol. 11, no. 3, 33–35, 2006 (Reprinted from *Electronics*, vol. 38, no. 8, 1965).

[5] "www.intel.com" [Online]. Available: www.intel.com.

[6] S. Narendra, V. De, S. Borkar, D. A. Antoniadis, and A. P. Chandrakasan, "Full-chip subthreshold leakage power prediction and reduction techniques for sub-0.18-µm CMOS," *IEEE J. Solid-State Circuits*, vol. 39, no. 3, pp. 501–510, 2004.

[7] B. H. Calhoun and a. P. Chandrakasan, "Standby power reduction using dynamic voltage scaling and canary flip-flop structures," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1504–1511, 2004.

[8] A. P. Chandrakasan and R. W. Brodersen, "A Portable Multimedia Terminal," *Low Power Digital CMOS Design*, Springer US, pp. 309–366, 1995.

[9] J. M. Rabaey, "Low power design essentials," Springer, 2009.

[10] A. Shehabi *et al.*, "United States Data Center Energy Usage Report," June 2016.

[11] D. Markovic, V. Stojanovic, B. Nikolic, M. A. Horowitz, and R. W. Brodersen, "Methods for true energy-performance optimization," *IEEE J. Solid-State Circuits*, vol. 39, no. 8, pp. 1282–1293, 2004.

[12] J.M. Rabaey, A.P. Chandrakasan, B. Nikolic, "Digital Integrated Circuits: a Design Perspective," Prentice-Hall, 2003.

[13] Sengupta and R. Saleh, "Generalized Power-Delay Metrics in Deep Submicron CMOS Designs," *IEEE Trans. CAD Integr. Circ. Syst.*, vol. 26, no. 1, pp. 183–189, 2007.

[14] S. Perri, P. Corsonello, M.A. Iachino, M. Lanuzza, G. Cocorullo, “Variable precision arithmetic circuits for FPGA-based multimedia processors,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol . 12, no. 9, pp. 995–999, 2004.

[15] H. Kaul *et al.*, “A 1.45GHz 52-to-162GFLOPS/W variable-precision floating-point fused multiply-add unit with certainty tracking in 32nm CMOS,” in *IEEE International Solid-state Circuits Conference* (*ISSCC*), 2012, pp. 182–184.

[16] S.-R. Kuang, K.-Y. Wu, K.-K. Yu, “Energy-efficient multiple-precision floating-point multiplier for embedded applications,” *J. Signal Process. Syst.*, vol. 72, no. 1, pp. 43–55, 2013.

[17] I. Levi, O. Bass, A. Kaizerman, A. Belenky, A. Fish, “High speed dual mode logic carry look ahead adder,” in *IEEE International Symposium on Circuits and Systems* (*ISCAS*), 2012, pp. 3037–3040.

[18] I. Levi, A. Fish, “Dual mode logic–Design for energy efficiency and high performance,” *IEEE Access*, vol. 1, pp. 258–265 2013.

[19] A. Kaizerman, S. Fisher, A. Fish, “Subthreshold dual mode logic,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 21, no. 5, pp. 979–983, 2013.

[20] I. Levi, A. Belenky, A. Fish, “Logical effort for CMOS-based dual mode logic gates,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 22, no. 5, pp. 1042–1053, 2014.

[21] I. Levi, A. Albeck, A. Fish, S. Wimer, “A low energy and high performance DM2 adder,” *IEEE Trans. Circ. Syst. I: Regular Pap.*, vol. 61, no. 11, pp. 3175–3183, 2014.

[22] V. Yuzhaninov, I. Levi, A. Fish, “Design flow and characterization methodology for dual mode logic,” *IEEE Access*, vol. 3, pp. 3089–3101, 2015.

[23] L. Moyal, I. Levi, A. Teman, A. Fish, “Synthesis of dual mode logic,” *Integrat. VLSI J.*, vol. 55, pp. 246–253, 2016.

[24] R. Taco, I. Levi, M. Lanuzza, A. Fish, “Evaluation of dual mode logic in 28nm FD-SOI technology,” in *IEEE International Symposium on Circuits and Systems* (*ISCAS*), 2017, pp. 1–4.

[25] R. De Rose, P. Romero, M. Lanuzza, “Double-precision Dual Mode Logic carry-save multiplier,” *Integrat. VLSI J.*, vol. 64, pp. 71–77, 2019.

[26] L. Heller, W. Griffin, J. Davis, and N. Thoma, “Cascode voltage switch logic: A differential CMOS logic family,” in *IEEE International Solid-State Circuits Conference*, 1984, pp. 16–17.

[27] K. M. Chu and D. L. Pulfrey, “A Comparison of CMOS Circuit Techniques:

Differential Cascode Voltage Switch Logic Versus Conventional Logic," *IEEE J. Solid-State Circuits*, vol. 22, no. 4, pp. 528–532, 1987.

[28] C. R. Tretz, R. K. Montoye, and W. Reohr, "Ratioed CMOS: a low power high speed design choice in SOI technologies," in *IEEE International SOI Conference*. Proceedings, 2000, pp. 28–29.

[29] D. Radhakrishnan, S. R. Whitaker, and G. K. Maki, "Formal design procedures for pass transistor switching circuits," *IEEE J. Solid-State Circuits*, vol. 20, no. 2, pp. 531–536, 1985.

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

[31] K. Yano, Y. Sasaki, K. Rikino, and K. Seki, "Top-down pass-transistor logic design," *IEEE J. Solid-State Circuits*, vol. 31, no. 6, pp. 792–803, 1996.

[32] R. Zimmermann and W. Fichtner, "Low-power logic styles: CMOS versus pass-transistor logic," *IEEE J. Solid-State Circuits*, vol. 32, no. 7, pp. 1079–1090, 1997.

[33] M. R. Meher, C. C. Jong, and C.-H. Chang, "A High Bit Rate Serial-Serial Multiplier With On-the-Fly Accumulation by Asynchronous Counters," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 19, no. 10, pp. 1733-1745, 2011.

[34] S. Abed, B. J. Mohd, Z. Al-bayati, and S. Alouneh, "Low power Wallace multiplier design based on wide counters," *Inter. Jour. Circ. Th. and App.*, vol. 40, no. 11, pp. 1175-1185, 2012.

[35] A. Cilardo *et al.*, "High Speed Speculative Multipliers Based on Speculative Carry-Save Tree," *IEEE Trans. Circ. Syst. I: Regular Pap.*, vol. 61, no. 12, pp. 3426-3435, 2014.

[36] S. Jia, S. Lyu, X. Li, L. Liu, and Y. He, "Simplified carry save adder-based array multiplier scheme and circuits design," *Inter. Jour. Circ. Th. and App.*, vol. 43, no. 9, pp. 1226-1234, 2015.

[37] A. A. Del Barrio, R. Hermida, and S. O. Memik, "A Partial Carry-Save On-the-Fly Correction Multispeculative Multiplier," *IEEE Trans. on Computers*, vol. 65, no. 11, pp. 3251-3264, 2016.

[38] D. Esposito, D. De Caro, E. Napoli, N. Petra, and A. G. M. Strollo, "On the use of approximate adders in carry-save multiplier-accumulators," in IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, May 2017, pp. 1-4.

[39] M. Aguirre-Hernandez and M. Linares-Aranda, "CMOS Full-Adders for

Energy-Efficient Arithmetic Applications," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 19, no. 4, pp. 718–721, 2011.

[40] P. Bhattacharyya, B. Kundu, S. Ghosh, V. Kumar, and A. Dandapat, "Performance Analysis of a Low-Power High-Speed Hybrid 1-bit Full Adder Circuit," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 23, no. 10, pp. 2001–2008, 2015.

[41] M. Linares Aranda, R. Baez, and O. Gonzalez Diaz, "Hybrid adders for high-speed arithmetic circuits: A comparison," in *7th International Conference on Electrical Engineering Computing Science and Automatic Control*, 2010, pp. 546–549.

[42] R. Taco, I. Levi, M. Lanuzza, and A. Fish, "Low voltage logic circuits exploiting gate level dynamic body biasing in 28 nm UTBB FD-SOI," *Solid-State Electronics*, vol. 117, pp. 185-192, 2016.

[43] N. Shavit, R. Taco, and A. Fish, "Efficiency of Dual Mode Logic in Nanoscale Technology Nodes," in *2018 IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE)*, 2018, pp. 1–4.

Acknowledgments

At the end of this PhD activity, I would like to express my gratitude to all the people and institutions that have contributed in some way helping and encouraging me.

First of all, I wish to express my utmost thanks to my advisor, Prof. Marco Lanuzza, for his support and patience. Without his help, this work would not have been possible.

My sincere thanks to Dr. Raffaele De Rose. His invaluable professional contribution and the time dedicated have been fundamental in this research activity.

To my colleague and friend Ramiro, for his support in the first stage of this activity, and for his friendship demonstrated during the time shared at the university and as a roommate.

I would like to thank my friends from the laboratory and the University of Calabria (UNICAL) for their advice, words of encouragement, spirit of companionship and good moments shared.

To the SENESCYT of Ecuador for its assistance for the completion of my studies, to my institution the Polytechnic School of Chimborazo- ESPOCH and my always remembered friend and polytechnic reference Dr. Romero Rodriguez.

My gratitude to the University of Calabria (UNICAL) and especially to one of its academic representatives, Prof. Felice Crupi for his collaboration during the PhD program.

I wish to sincerely thank my parents Manuel and Gloria, my always remembered second mother Rosita María, my sisters and the whole family in general for their concern, their love, support at all times and circumstances.

And of course, to my beloved wife Jenny Isabel and my two little treasures Paul Antonio and Sebastian Alejandro, for their sacrifice, understanding, love and for being the source that motivated and inspired me to complete this stage in my professional life.

List of Publications

- R. De Rose, **P. Romero**, M. Lanuzza, “Double-precision Dual Mode Logic carry-save multiplier,” *Integrat. VLSI J.*, vol. 64, pp. 71–77, 2019.