Energy Efficient Solutions for Computing and Sensing

Mehdi Safarpour

Abstract

Emerging technologies, such as the Internet of Things (IoT), Deep Neural Network (DNN) based machine learning, and 6th generation wireless communications, impose substantial performance and energy efficiency demands for implementations. Thermal dissipation and energy supply constraints alone set stringent limits on the designs. In answer to the requirements, this thesis focuses on improving the efficiency of selected energy hungry system sections, from signal acquisition to computing.

The proposed novel approach to energy efficient computing is based on harnessing low-voltage operation together with the Algorithm Based Fault Tolerance (ABFT) technique to detect the reliability threshold. The scheme has been demonstrated using a Field Programmable Gate Array (FPGA) device, showing around $\approx$2x energy efficiency improvement in neural network computing. Transistor level simulations with a systolic Application Specific Integrated Circuit (ASIC) architecture promise energy dissipation reductions of up to 90%. Representing a novel compromise between energy efficiency and fault-tolerance, the approach provides new insights for designing hardware accelerators.

Two approaches are investigated for improving energy efficiency when acquiring digital signals. A Nyquist-Shannon rate sampling framework employing novel arithmetic tracking, Successive Approximation Register (SAR) Analog to Digital Converter (ADC), is proposed for signals exhibiting low activity periods. The solution reduces the power dissipation of the analog section, while requiring minor changes to the digital side. For sub-Nyquist-Shannon rate cases, a sampling scheme using a dual-mode ADC to conserve energy in signal reconstruction is proposed.

The proposed approaches have so far been studied in restricted implementation settings and simulations. However, they possess wide applicability from sensor nodes and wireless base-band designs to supercomputing.

Keywords: analog to digital converter, algorithm based fault tolerance, FPGA, low voltage operation, matrix multiplier, signal reconstruction, systolic array processor, VLSI

Tiivistelmä

Esineiden Internetin, syviin neuroverkkoihin perustuvan koneoppimisen ja kuudennen sukupolven langattoman tietoliikenteen kaltaiset orastavat teknologiat vaativat toteutuksissaan korkeaa suorituskykyä ja energiatehokkuutta. Suunnittelulle kovia vaatimuksia seuraa pelkästään jo lämmöntuoton ja energialähteiden rajoitteista. Tässä tutkimuksessa näihin vaatimuksiin vastataan parantamalla järjestelmäosien energiatehokkuutta signaalien syötöstä laskentaoperaatioihin.

Tuloksena saatu uusi energiatehokkaan laskennan ratkaisumalli yhdistää toiminnan matalalla käyttöjännitteellä virhekynnyksen havaitsemiseen algoritmisella tekniikalla (Algorithm Based Fault Tolerance, ABFT). Menettely on demonstroitu ohjelmoitavalla logiikkapiirillä (Field Programmable Gate Array, FPGA), jolle toteutetun neuraalilaskennan energiasta säästettiin 50 %. Transistoritasolla simuloidun systolisen sovellusspesifisen piiritoteutuksen (Application Specific Integrated Circuit, ASIC) tapauksessa todettiin jopa 90 % energiankulutuksen pudottaminen mahdolliseksi. Ratkaisumalli on uudenlainen kompromissi energiatehokkuuden ja virhesietoisuuden välillä, antaen aiemmasta poikkeavan näkökulman laitteistolla toteutettujen laskentakiihdyttiminen toteuttamiseen.

Digitaalisten signaalien hankinnan energiatehokkuuden parantamiseksi tutkittiin kahta menettelytapaa. Ajoittain vain vähän muuttuvien signaalien tapaukseen esitetään uutta aritmeettisella periaatteella signaalia seuraavaa peräkkäisten approksimaatioiden analogia-digitaalimuunninta (Successive Approximation Register Analog to Digital Converter, SAR ADC). Tämä Nyquist-Shannon-näytteistyskriteerin mukainen ratkaisu pienentää muuntimen analogisen osuuden tehonkulutusta ja edellyttää vähäisiä muutoksia sen digitaaliselle puolelle. Alle Nyquist-Shannon-kriteerin tapauksiin esitetään näytteistystä kaksitoimisella analogia-digitaalimuuntimella energian säästämiseksi signaalia rekonstruoitaessa.

Esitettyjä ratkaisumalleja on toistaiseksi tutkittu rajoitetuissa toteutuksissa sekä simuloiden. Niillä on kuitenkin näköpiirissä laaja sovellettavuus anturisolmuista ja langattomasta tietoliikenteestä superlaskentaan.

Asiasanat: analogia-digitaalimuunnin, algoritminen virhesietoisuus, FPGA, matala käyttöjännite, matriisikertoja, signaalin entistys, systolinen matriisiprosessori, VLSI

To my family

Acknowledgements

The doctoral studies gave me a unique opportunity to work and become friends with exceptionally talented and good-spirited individuals. I am forever indebted to my supervisor, Prof. Olli Silvén for his guidance. Not only helping me to understand and solve technical challenges, he shaped my research and scientific writing skills, both of which I lacked in the beginning of my studies. I am also deeply grateful for the pre-examiners Prof. Luigi Carro and Docent Lauri Koskinen who through insightful comments helped to improve this thesis.

I would like to thank Dr. Reza Inanlou for co-authoring many papers and agreeing to work on my ideas after I visited him at the University of Tehran. Our accidental collaboration turned into a deep friendship.

Dr. Ilkka Hautala's enthusiasm and his constant encouragements made me bold enough to be able explore new exotic areas of research. I admire his solid professional ethics and manners. Ilkka, say my greetings to Ilmari, and thank him for patiently listening to our scientific discussions, while sharing his candies and occasionally his toys, whenever I was a guest at your home.

Furthermore, I would like to thank Professors Jukka Lahti, Timo Rahkonen, Markku Juntti, and members of my follow up group, Prof. Juha Röning and Docent Konstantin Mikhaylov. They provided important technical advice and contributions to my work, and pushed to write the thesis.

I would like to thank all of my friends during my PhD studies for their support, thank you, Akabar, Hadi, Iisa, Ilmari and Eine Mäenpää, Usman, Tuomas, Pertti, Olli Niemitalo, Rebvar and Lida, Janne Mustaniemi, Miguel, Tino, Marty and Dean Stewart, Mohammad and Akram, Sadegh, Mohammadreza Kamali and my old roommate Hou Defeng. In particular, Mohammad Mahdi Mirloo and Dr. Mazaher Rezaei have been a great support.

The readability of this thesis and my scientific papers owes to the expert language revisions by Mr. Gordon Roberts. Those have been a great learning experience.

This thesis was supported under the Academy of Finland projects ICONICAL and 6G flagship. The generous support from Nokia foundation, Riitta ja Jorma J. Takanen foundation, Tauno Tönning foundation and Walter Ahlström foundation are gratefully acknowledged.

Finally, I would like to thank my parents Hashem and Shahla for their dedication and scarification for family, and my siblings, Parvin and Mohammadreza for their love and support.

List of abbreviations

5G	*5th generation*
6G	*6th generation*
ABFT	*algorithm based fault tolerance*
ADC	*analog to digital converter*
AIC	*analog to information converter*
AMP	*approximate message passing*
ASIC	*application specific integrated circuit*
ASIP	*application specific instruction-set processor*
BRAM	*block random access memory*
C	*capacitance*
CAD	*computer aided design*
CNN	*convolutional neural network*
CONV	*convolutional*
CS	*compressive sensing*
DAC	*digital to analog converter*
DCNN	*deep convolutional neural network*
DFT	*discrete Fourier transform*
DMR	*dual modular redundancy*
DNN	*deep neural network*
DSP	*digital signal processor*
DVFS	*dynamic voltage and frequency scaling*
DVS	*dynamic voltage scaling*
ECC	*error correction code*
ECG	*electrocardiogram*
EDA	*electronic design automation*
EDS	*error detection sequence*
EEG	*electroencephalogram*
FC	*fully connected*
FF	*flip flop*
FFT	*fast Fourier transform*
FLOPS	*floating point operations per second*
FPGA	*field-programmable gate array*
GOPS	*giga operations per second*
GPP	*general purpose processor*

GPU	*graphics processing unit*
HDL	*hardware description language*
HLS	*high level synthesis*
IHT	*iterative hard-thresholding*
I/O	*input and output*
IoT	*internet of things*
ITD	*inverse temperature dependence*
L	*length*
LFSR	*linear-feedback shift register*
LS	*least squares*
LSB	*least significant bit*
LTE	*long term evolution*
LUT	*look-up table*
MAC	*multiply-accumulate*
MIMO	*multiple-input and multiple-output*
MSB	*most significant bit*
N-IHT	*normalized-iterative hard-thresholding*
NN	*neural network*
NT	*near-threshold*
NUS	*non-uniform sampler*
OMP	*orthogonal matching pursuit*
PC	*personal computer*
PE	*processing element*
PL	*programmable logic*
PLL	*phase locked loop*
PM	*power management*
PMBUS	*power management bus*
PS	*processing system*
PVT	*process-voltage-temperature*
RAM	*random-access memory*
RC	*resistor-capacitor*
RZFF	*razor flip flop*
RTL	*register-transfer level*
SAR	*successive approximation register*
SNR	*signal to noise ratio*
SoC	*system on chip*
SPICE	*simulation program with integrated circuit emphasis*

SRAM	*static random-access memory*
ST	*sub-threshold*
TED	*timing-error-detection*
TMR	*triple modular redundancy*
TPU	*tensor processing unit*
TTA	*transport triggered architecture*
VHDL	*vhsiv hardware description language*
VID	*voltage identification*
W	*width*

Contents

1 Introduction

The advancing wireless technologies with their constantly increasing data bandwidths, ongoing development toward ultra-densified Internet of Things (IoT) sensor and actuator infrastructures, as well as the emerging edge computing technologies are facing substantial energy efficiency challenges. Reduced power dissipation enables smaller and cheaper battery operated and self-powered devices, while curbing the cooling needs of mains-powered computing nodes, and even super-computers. Such improvements have also a role in combating global warming, as new applications of information technologies appear to be more computing and communications hungry than their predecessors. Systems and devices impacted by potential low-power innovations are illustrated in Fig. 1.

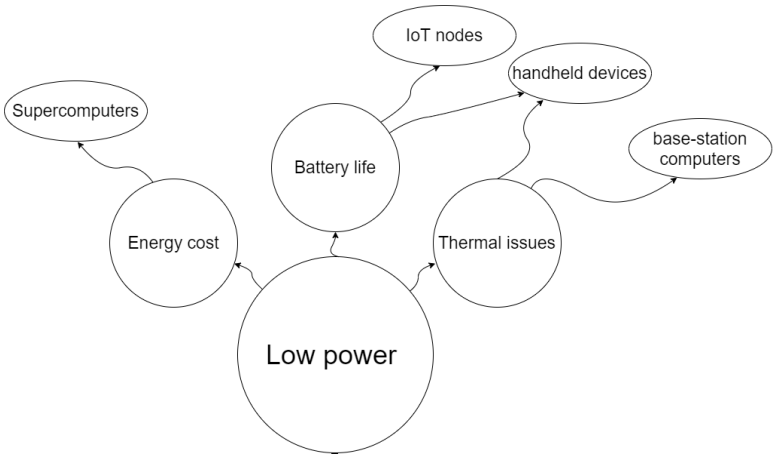

Fig. 1. Reducing power consumption addresses multiple challenges in different application areas.

For instance, the optimizations needed for operating Multi-Input and Multi Output (MIMO) and massive-MIMO wireless communications require huge computing power. While machine learning approaches might emerge as competitive replacements for the conventional algorithms, say, for channel estimation and beam forming, they are also computationally demanding [1, 2]. Interestingly, in all cases, various matrix operations dictate the computational costs of the algorithms.

Supplying the energy and computing power/energy is a design challenge for energy constrained devices such as mobile handsets and sensor nodes. In turn, the designers

and operators of base stations and edge computing nodes in the network infrastructure struggle with thermal management and energy costs. The heat generated is directly proportional to the dissipated power [3].

To illuminate the issues at a more detailed level, consider the application in Fig.2 where data from an image sensor is acquired by a sensor node that transmits a source coded result to reconstruction and analysis at the "edge" [4, 5]. Each step from the image sensor and data acquisition to radio communications and analytics dissipate energy, regardless of their varying implementation technologies.

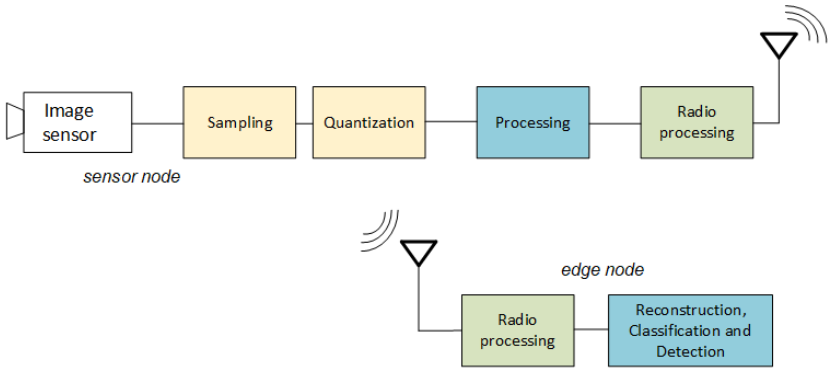

Fig. 2. A general electronics system such IoT nodes.

The interplay between the stages provides opportunities to explore options to minimize the overall power/energy consumption, as well as to maximize the energy efficiency of a particular critical step regardless of the impacts elsewhere. Sensing, transmitting, processing and reconstructing the information are the most energy demanding tasks. The focus of this thesis is on improving the efficiencies of signal acquisition and computing.

A purely economic issue is the high costs of adopting new, say 5-7 nm, circuit technologies. It is therefore attractive to use the whole energy efficiency potential of the cheaper silicon processes before having to migrate to the newer ones. The contributions of this current thesis also serve such objectives.

1.1 Contributions of the thesis

This thesis proposes novel solutions for improving the energy efficiency of computing and signal acquisition. The proposed low-voltage domain approach can be exploited even without modifications to the chips. For interfacing with analog sensors, data converters were developed that utilize signal characteristics. In this context, also a

new scheme is proposed that conserves computing in sparse signal reconstruction with slightly increased energy dissipation at the sampling step.

Conventional approaches to improve the energy efficiency of computing include employing a more advanced silicon technology, and optimizing architecture and algorithms, e.g. to maximize hardware utilization. In contrast, in this thesis, the supply voltage of a particular design is pushed to the limit beyond which errors may appear, while reliability is maintained by a simple algorithmic fault tolerance scheme. In return, the power dissipation drops substantially without adverse throughput impacts.

For time-discretization and quantization, designs that match particular kinds of signals were investigated. An adaptive arithmetic tracking successive approximation register ADC approach is proposed and demonstrated by simulations for signals with periodic higher activity. Also a dual resolution mode SAR ADC is proposed and demonstrated using physical hardware, showing benefits for signal reconstruction within the Compressive Sensing framework.

The first part of this thesis is dedicate to low-power and energy efficient techniques for digital processing platforms. After an introduction to the fundamental concepts in Chapter 2, the solutions created during the doctoral research are presented in Chapters 3 and 4. Chapters 5 and 6, that form the second part of thesis, investigate energy-efficient signal acquisition, presenting schemes for analog-to-digital conversion and signal reconstruction.

The key contributions of this thesis have earlier been published in the seven separate articles, i.e., [5], [6], [7], [8], [9], [10] and [11]. The novel contributions of this thesis are the following:

1. Proposing applicability of ABFT technique for enabling reliable reduced voltage operation in digital processors and showcasing its utility through FPGA implementation of matrix multiplication and neural networks and relevant empirical experimentation.
2. Introducing an architecture based on systolic array structure that can operate at reduced voltage by integrating ABFT and demonstrating the benefits of the scheme in achieving aggressive reduced voltage operation.
3. Developing a vision chip as a systolic processing array that operates at reduced voltage. Showing that inherent parallelism of image processing operations can be leveraged to compensate for modest clock rates imposed by low-voltage operating.
4. Proposing a SAR ADC architecture to minimize the number of conversion cycles in oversampling applications and later extending the design to be used for signals with periodic low-activity mode and finally verifying increased energy efficiency through simulations.

5. Evaluating different compressive sensing reconstruction schemes for sparse signals, and preforming empirical comparison between those on three different embedded platforms. Addressing the inefficiencies of conventional sparse signal reconstruction approaches through presenting a new architecture for sparse signal sampling and reconstruction.

This thesis is a more comprehensive presentation and reflects the author's contributions to research fields of energy efficient computing and sensing. The author has the main role in six of the publications listed above and assisted in the development of article [9]. The author crafted the hardware implementation for articles [8], [5], [10], and [11], developed system level simulation models for articles [6], [7], [5], [9], and [11], assisted in the development of transistor level simulation models for [6], [7], and [9], and conducted experiments and data collection in all manuscripts and wrote the first drafts.

2 Preliminaries of low-voltage computing

Multiple technological advancements in the areas of artificial intelligent and wireless communications push for high performance computers, while energy consumption is becoming one of the main design challenges. In particular, the gains from process scaling have proven to be modest in recent years [12, 13]. Conventional approaches to increase performance or energy efficiency, include designing optimized architectures for specific applications [5], migrating designs to more advanced process technologies [14], and utilizing algorithm innovations to cut computational demands [15, 16].

In this thesis, higher energy efficiency is pursued by pushing the digital processors to their operation voltage limits. The effectiveness of the approach is shown for matrix computation that holds a key role both in DNNs and wireless telecommunications [17]. While the ADC solutions proposed in this thesis would benefit from reduced voltage operations, in their case, a more conservative policy has been followed.

2.1 Power and speed in digital circuits

Power, speed and reliability are tied together in digital circuits. We investigate their association and the operation voltage in the following to justify our approach.

2.1.1 Power dissipation

Power supplied to digital circuits is dissipated as *a)* dynamic P_D and *b)* static power, P_S[1]. The total power consumption is their sum, as shown in Equation 1. The dynamic power dissipation is due to the level of charging and discharging activity (α) of the parasitic load capacitors (C) during transistor switching, the operating clock frequency f, and the supply voltage (V_{dd}^2). The physical design of the transistors defines C, while reducing V_{dd} results in a quadratic cut in dynamic power [18]

$$P = \underbrace{\alpha f C V_{dd}^2}_{\text{dynamic power}} + \underbrace{I_{leakage} V_{dd}}_{\text{static power}}. \tag{1}$$

Notice the power and energy relation as well, shown in Equation 2

$$P = E/t. \tag{2}$$

[1] contribution of the short-circuit power dissipation is ignored here

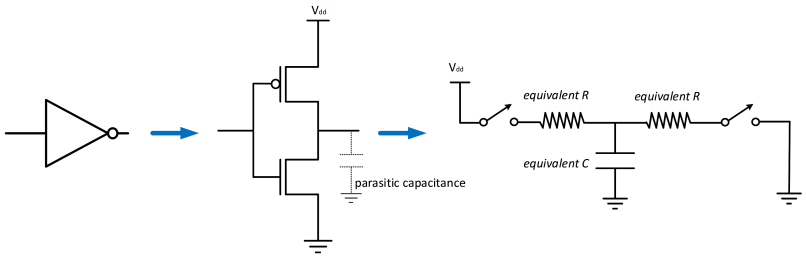

Fig. 3. Simple RC circuit model for delay and energy analysis of logic gates.

The static power depends on the the supply voltage V_{dd} and leakage current ($I_{leakage}$) that does not depend on the activity of the circuit. It is enough for the circuit to turn on, regardless of the activity level, and static power will be dissipated. With older technologies dynamic power was the dominant part in the power equation, while with newer ones the static component might be responsible for up to a third of the total power dissipation[19, 20].

2.1.2 Propagation delay and timing constraints

Considering a simplified Resistor-Capacitor (RC) circuit model, as shown in Fig. 3, the gate delay depends on the load capacitance C that is defined by transistor sizing, and the resistance R of the transistor switches. In addition to transistor sizing, R depends on the threshold and supply voltages [18, 20].

A power-performance trade off appears when the supply voltage is considered. In digital logic gates, while the power dissipation is directly proportional, the speed is "inversely" proportional to the voltage ($\propto 1/V_{dd}$) [18]. In other words, at reduced voltages, the gates consume less energy at the cost of being slower [18].

In synchronous circuits, each combinatorial logic section is surrounded by flip-flops that provide the input from the previous stage, and samples the output of the current one, as shown in Fig. 4 at the defined clock edge. In digital circuits, the propagation delay is the length of time that it takes for an input signal to travel across a combinatorial to the next set of flip-flops. Considering the process variations and the number of gates engaged, some of the delay paths are the slowest/longest ones. The maximum clock frequency of the circuit is determined by the propagation delay of the longest delay paths plus a small timing margin to avoid meta-stability of the flip-flops.

If the supply voltage is reduced, less energy is consumed, but the logic gates will be slower as well. Early sampling of output of the combinatorial logic, while the signal is still propagating within the circuit, causes a timing error. Hence, to avoid timing errors

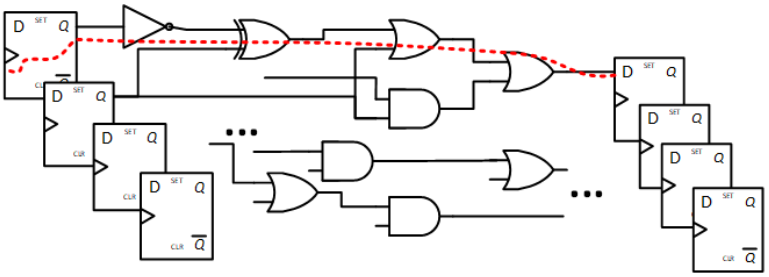

Fig. 4. An example delay path in a digital circuit.

and faulty operation at reduced voltages, it is necessary to slow the clock rate to grant enough time for the input signal to travel through the longest delay paths before the next clock edge.

2.2 Power reduction techniques

Each low-power technique targets one, or multiple contributing factors of overall power consumption in Equation 1. In the following, the widely used low-power design techniques are reviewed to provide a background for the main contributions of this thesis.

2.2.1 Clock gating

Among the simplest and most effective techniques to reduce dynamic power consumption is clock gating. When the input signals of a circuit block are constant, or when its outputs are not used, clocking to that block may be shut off [21]. Clock gating can be automatically integrated to the design using Electronic Design Automation (EDA) tools.

Considering Equation 1, the clock generation and distribution network have the highest activity factor in the digital circuit with a large driven capacitive load. Its share of the total power of a circuit can be as high as 50% [22]. Hence, reducing unnecessary clock associated activity substantially contributes to energy savings [22].

As depicted in Fig. 5 several sources contribute to the dynamic power dissipation of the clock network: the load of the wiring (Fig. 5), the intermediate buffers for driving capacitive loads, the load from flip-flops within the circuit, and unnecessary activity of the combinational circuits that might be directly connected to the clock.

While clock gating saves dynamic power wasted in the clock distribution network during idle periods, it cannot help with reducing overall static power dissipation of the

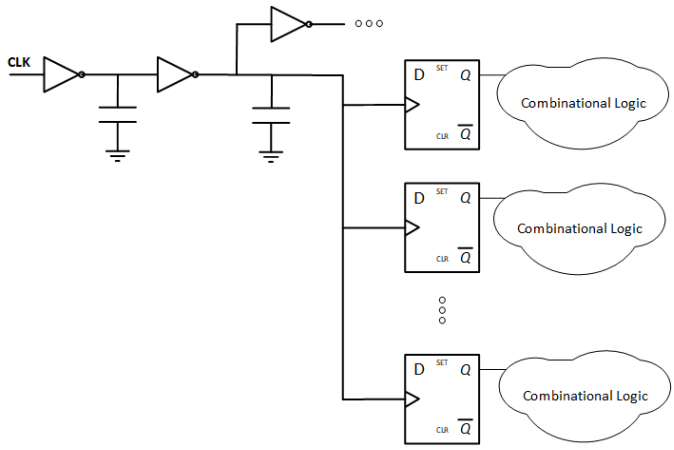

Fig. 5. The clock network delivers a clock signal to all synchronous components on the circuit.

idle circuit block. In many applications, the idle periods are predictable and long enough to justify turning off not only the clock, but even the power from entire circuit blocks.

2.2.2 Power gating

By shutting off the supply voltage, unnecessary dissipation of static power is prevented. This technique is called power gating. It is justifiable in large digital systems in which some circuit blocks, e.g. specific algorithm accelerators, may remain idle for prolonged periods of time.

A straightforward power gating implementation involves adding a transistor switch either on the supply or ground side of a circuit block, as shown in Fig. 6, and controlling power through the switch. The voltage overhead from the switch must be low to minimize wasted energy when the switch is on. This requires selecting relatively high W/L ratio switch transistors. The challenge with power gating is the long wake-up time for the powered-down circuit block that necessitates a careful task scheduling scheme.

2.2.3 Voltage and frequency scaling

Instead of operating the design at a high clock rate followed by clock and/or power gated idle periods to save power [2], voltage and frequency scaling can take place dynamically,

[2] i.e. race to sleep and race to idle schemes [23]

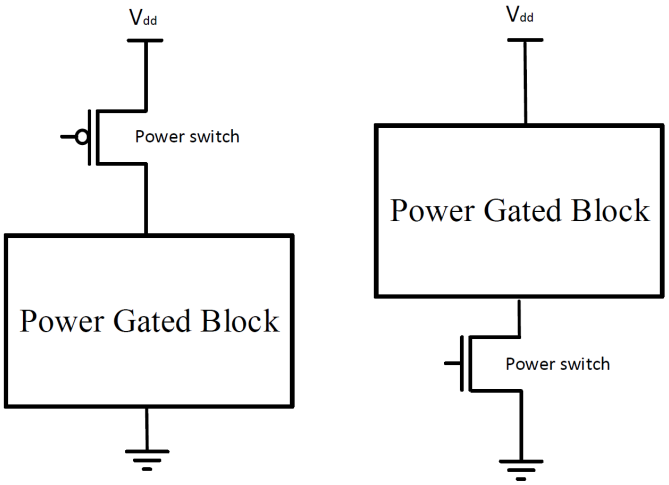

Fig. 6. Power switches shut off the leakage current to large power gated circuit blocks.

based on the workload. Equation 1 indicates that dynamic power is directly proportional to the frequency, and its reduction enables lowering the supply voltage as well.

Although most voltage scaling approaches require stepping down the clock frequency to be on par with reducing voltage, it is possible to save power without having to compromise the performance [24]. This is due to the fact chip vendors insert margins to ensure functionality in the worst case operating conditions, as described in next section.

2.2.4 Voltage guard bands

To guarantee the functioning of the delivered chips, the manufacturers take into account the process and ambient operating range, and add margins on top of the specified "best-case" operating voltage [25]. In this context, the minimum functional supply voltage is the one at which the device operates without errors. The margins ensure that even the slowest delivered devices work correctly in the worst-case ambient temperature at a specified clock frequency [26]. As shown in Fig. 7, the extra voltage margin adds a timing margin. The extra margins cover for variations that may cause timing errors [24, 25].

Recent research, e.g. [27] and [28, 29], has demonstrated that the nominal operation voltages of off-the-shelf FPGAs recommended by vendors are defined very conservatively. Investigations in commercial chips experimentally verified large margin between the nominal and critical voltages where the operations fail [24, 25]. The capability to reduce the supply voltage by 20% from the specified nominal one was demonstrated for

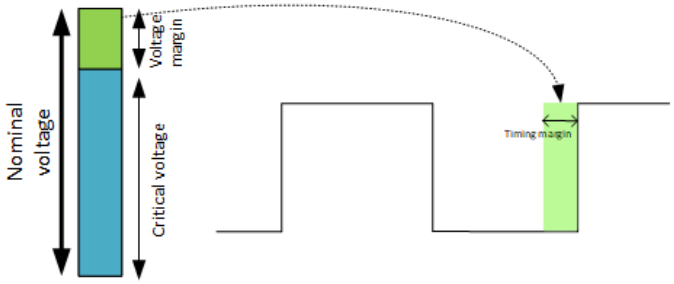

Fig. 7. Voltage margins act as guard bands against possible variations. Redrawn from [25].

commercial processors [30, 26]. Example studies of voltage margins on different platforms are listed in [31]. A safe voltage scaling scheme can save substantial energy due to quadratic energy-voltage relation, without sacrificing the performance or reliability. For example, reducing the supply voltage of an ARM processor by around 11.1% results in conserving around 18% of the energy, without performance degradation [31]. Notice, "guardbands" exist in typical chips, but for some chips with extensive testing or specific process technologies, they can be close to zero.

It is possible to scale down the voltage more aggressively beyond the voltage margins, and to reach operating voltages close to or even below the threshold voltage of the transistors [32]. In the Near-Threshold (NT) region, the operating voltage is slightly higher that the threshold voltage, while Sub-Threshold (ST) schemes reduce it below the threshold voltage [19]. In following, the opportunities and challenges of NT and ST operating regions are discussed.

2.2.5 Near-threshold and sub-threshold operating regions

Operating digital logic with near and below threshold voltage of transistors was first proposed in 1972 in [33]. In theory, these NT and ST region schemes possess potential to improve the energy efficiency by multiple folds, even 20x to 100x [19, 34]. Experimental results have demonstrated such improvements [35, 34, 36].

When the voltage is reduced beyond the design margins, down scaling of clock frequency is inevitable, due to increased circuit delay. Due to prolonged clock periods, the balance between dynamic and static power consumption changes, and an optimum operation point exists where the total energy dissipation per operation is at a minimum, as illustrated in Fig 8.

Dropping the voltage from the nominal maximum to close to the near-threshold level, the clock frequency that maintains errorless operation scales down almost linearly

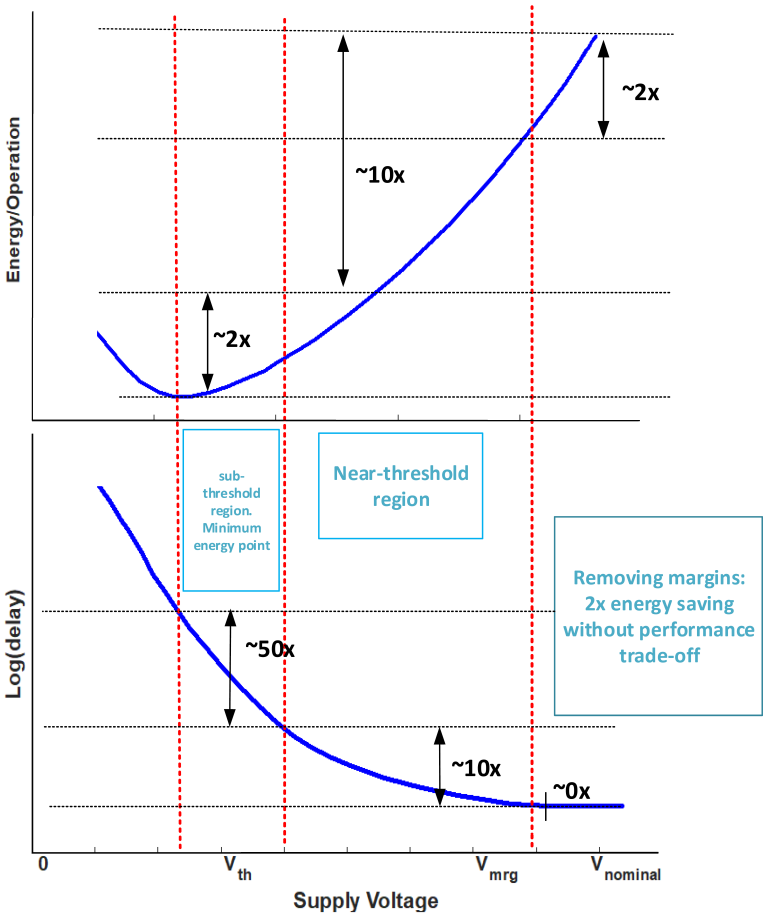

Fig. 8. Total dissipated energy per operation vs scaled supply voltage. Redrawn from [19].

by around a factor of ten. With further voltage reductions beyond the threshold voltage, the circuit delay increases exponentially. Nonetheless, the optimal energy point resides in the ST region [34].

As can be observed in Fig. 8 [19], reducing the voltage further to the near-threshold region yields a 10x improvement in energy efficiency, but at the same time 100x performance loss [19]. The performance loss in the sub-threshold region is more severe, and yet another 10x reduction in clock frequency provides for only a 2x increase in energy efficiency [19].

Consequently, sub-threshold designs are mostly aimed at energy scavenging sensor nodes where performance is less critical than energy efficiency [32]. In many other uses,

operating in the NT region loss provides for a balance between performance and energy efficiency. Moreover, in the NT region,the impact of process variations is more tractable [19].

Addressing performance loss

While operating in the near-threshold region promises a multiple fold increase in energy efficiency, it comes at the cost of substantial performance loss as shown in Fig. 8. However, depending on the application, the loss can be compensated for through massive parallelism, maintaining the energy efficiency gains.

For instance, processing arrays for many vision applications suit the near-threshold region approach. In [5], we explored such a massively parallel scheme to compensate for the performance loss from operating in the near-threshold region. According to the co-simulation results, the energy efficiency of a vision chip improved by $\approx 3x$ when the operation voltage was reduced from the nominal 1.0 V to 0.6 V. The functioning was verified by executing multiple vision algorithms in the array and showing that the performance sufficed to meet the deadlines.

Process variations

Due to imperfections in the integrated circuit manufacturing process, the physical characteristics of transistors, e.g., size (W and L), doping, oxide layer, threshold voltage V_{th}, have variations. These lead to both random and systematic differences in the voltage-current characteristics of transistors [37].

Close to the nominal voltage, the impact of variations is relatively small, but it becomes substantial as the voltage is reduced. This causes a major problem in achieving reliable energy optimum designs, since the characterization of cell libraries for static analysis becomes difficult. For example, the gate delay difference of a Fast-Fast process corner versus a Slow-Slow process corner may reach up to 100x [19, 32].

Timing errors at reduced voltages

Optimizing for energy efficiency demands matching the reduced voltage at the maximum frequency with error-free operation. In other words, as the voltage is dropped, the frequency must be kept as high as possible, while ensuring the functionality [38, 39]. To ensure correct functionality, the signal must arrive at a setup time before the next rising edge of the clock in order to avoid timing errors [38].

Reducing voltage results in longer gate delays and a longer time is needed for the signal to travel across the delay paths. So, the timing constraints become tighter, and require longer clock periods. Furthermore, operating at a reduced voltage might impact the reliability of the clocking network as well, resulting in timing violations [37].

Due to the process and ambient variations, the maximum error free clock frequency for each voltage cannot be determined statically. A set of techniques exists that enables dynamic adaptation of clock frequency to avoid timing errors. In following, we introduce some of them. The solutions proposed in this thesis fall into in this category.

2.2.6 Methods for dynamic voltage and frequency scaling

Offline characterization

Among the simplest approaches to reduced voltage operation is characterisation of the device at different operating points. Such Dynamic Voltage and Frequency Scaling (DVFS) techniques are widely adopted in commercial devices [40]. The DVFS scheme used in Intel processors [41] relies on voltage-frequency operating points stored in a look-up table, prepared during a post manufacturing calibration process. These operating points are conservative, and are subject to variations over time, e.g., due to aging. Hence, DVFS systems err on the side of caution by adopting conservative voltage-frequency scaling schemes [42]. Similarly for FPGAs, Intel has introduced SmartVID technology [43] that simply identifies and stores the minimum operating voltage for each specific device considering the performance requirements [44]. The approach supports limited voltage scaling due to timing pessimism, and hence discourages aggressive energy optimization [44].

Ahmed et al. [45], propose to move from a device specific toward a device and application specific approach [46] in FPGA based designs. In other words, in contrast to SmartVID, that only characterizes the device, they characterize the device while running a specific application. Their method finds the minimum voltage for the targeted performance at which a specific application runs error free on the FPGA [46]. After each time the FPGA is turned on, a calibration process is carried out. The result of this is information about optimal operating points in different temperatures. This data is stored as a look-up table that guides the online DVFS system.

The downside of the scheme is that calibration for multiple temperature-voltage points consumes energy, and if frequently repeated, the process can be disruptive.

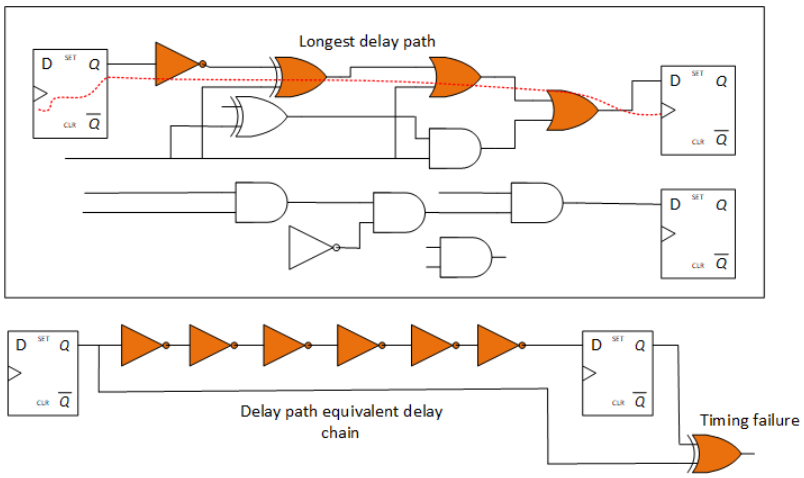

Fig. 9. A delay chain mimics timing behavior of the longest delay path.

Logic delay measurement circuits

A more advanced technique to optimize the operating point with respect to dynamic variations is to determine the logic delay and adjust the supply voltage accordingly. With this method, the longest delay paths of the design are identified, and a circuit using the same gates or an equivalent delay chain with the same timing behavior is embedded in the design, as shown in Fig. 9. The delay measurement circuit is constantly stimulated while the operating point is adjusted. Using a closed feedback loop [47], the operating point is adjusted based on the timing failures detected by the replica circuit [48]. Due to intra-die variations, a safety margin needs to be added to the replicated delay chain.

This technique has been demonstrated enabling substantial energy savings in limited applications [47]; however, it necessitates adding at least a single gate delay margin [45]. This timing pessimism prevents us from getting very close to optimum operating points, while too liberal utilization of the method is a reliability risk [46].

Timing error detection systems

Utilization of Timing Error Detection (TED) systems is a robust technique for adaptive voltage scaling that enables detecting timing errors on-the-fly in situ. The TED approach was originally devised to treat Process-Voltage-Temperature (PVT) variations in integrated circuit manufacturing, but later found a use in setting dynamic margins in low-voltage digital systems [49, 35, 34].

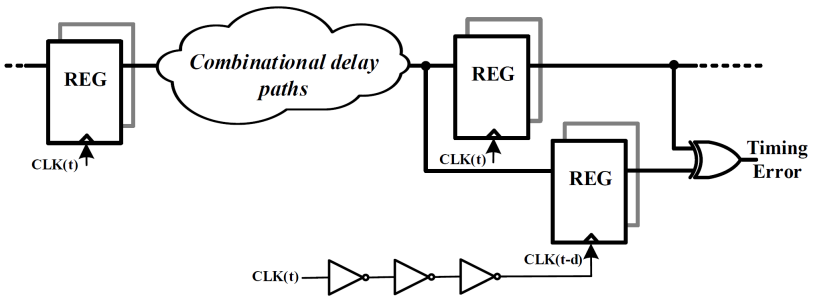

Fig. 10. The Timing-Error-Detection circuit detects the late arrival of a signal through dual sampling of the output of the combinatorial logic delay path. Reprinted, with permission, from Paper [6] © 2021 IEEE.

TED systems rely on Error Detection Sequence (EDS) circuits that are added to those critical paths where the timing errors are most likely to occur [32, 35]. A simple EDS is depicted in Fig. 10.

EDS circuits detect the late arrival of a signal by sampling the output of the combinational logic path twice, with the main and EDS flip-flops. The latter is triggered with a delayed version of the clock. In the simple case depicted in Fig. 10, the clock is delayed by a chain of inverters. The sampled versions of the outputs of both sets of flip-flops are XORed. In the case of any mismatch, a signal is generated, indicating a timing error.

TED systems require insertion into the design through circuit-level modifications before fabrication. This can be a laborious process, increasing the development time significantly [3]. Furthermore, while enabling dynamic reduction of operating voltage, and cutting overall energy consumption, TEDs themselves introduce non-trivial area and energy overheads. Even in the case of FPGAs that provide soft logic fabrics, it is difficult to add TED systems to the design, as the scheme is not supported by the commonly used design tools [51].

This thesis explores approaches for detecting timing related computational errors as effectively as with TED systems. However, the objective has been smaller overheads and design costs. These approaches are introduced in Chapters 3 and 4.

[3] Augmenting designs with error detection and prevention systems are offered as service by companies such as Minima Processor [50].

2.3 Targeted applications

The dynamic voltage scaling solutions presented in this thesis rely on algorithmic error detection mechanisms that are application specific. Hence, the applications of interest are briefly described below, before introducing the error detection methods.

2.3.1 Multiple-input multiple-output technology

Multiple-Input Multiple-Output (MIMO) and recently massive-MIMO are emerging technologies in wireless communications that enable increasing capacity of radio links by leveraging multiple antennas in the receiver and transmission to achieve spatial multiplexing [2]. MIMO technologies include the beam-forming technique in which interference of electromagnetic waves is utilized by synchronization of antenna in a manner that the transmitted or received signal beam is concentrated toward a specific direction. MIMO enables spatial multiplexing that can significantly enhance the wireless channel capacity by utilizing multi-path propagation [52].

MIMO technologies require different types of computations both on the transmitter and receiver sides. For example, in the transmitter, the channel should be evaluated first to estimate its capacity. Then, the power radiated by each antenna needs to be determined. Systems such as LTE and 5G also devote computations to pre-coding of data [53, 54]. Similarly, on the receiver side, channel estimation and decoding computing is carried out [53, 55].

The computing in current MIMO transceivers includes basic linear algebra operation such as matrix arithmetic and decompositions [53]. For example, in mobile base-stations the growing computational demands are in conflict with thermal dissipation constraints, calling for solutions to improve the energy efficiency [56].

2.3.2 Deep neural networks

Deep Neural Networks (DNN) have demonstrated significant performance in machine learning [57]. Deep Convolutional Neural Networks (DCNN), as a major sub-type of DNNs, have been extensively adopted in applications ranging from image and speech recognition to communications [1]. Both DCNN training and inference phases are computationally demanding, for instance, the model proposed in [58] requires around 40 billion floating-point operations (FLOP) and more than 500 MBytes of memory for storing the model parameters [57].

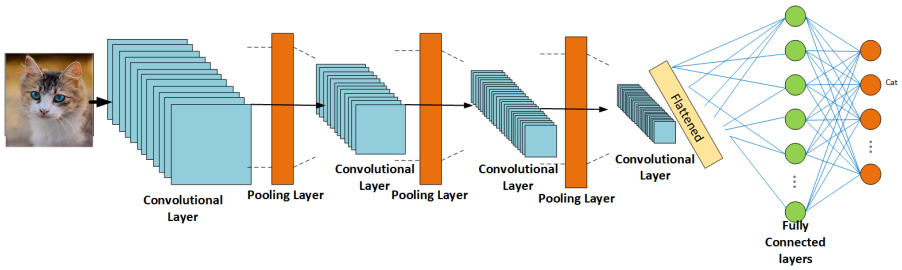

Fig. 11. An example CNN model that consists of multiple convolutional layers followed by fully connected layers for classification.

An example depicted in Fig. 11 is a DCNN that consists of convolutional (CONV), pooling, activation function and Fully-Connected (FC) layers. The largest share of the computing resources are consumed by the FC and convolutional layers [57]. The operation of the FC layers is essentially matrix-vector multiplications, while the activation layers implement non-linear functions, typically a Rectified Linear unit (ReLu) and Sigmoid [59] that are applied element-wise over the output of their preceding layers.

The convolutional layers receive multi-dimensional tensors and conduct convolution operations on the input and network weights [60]. The result is passed through the pooling layers, which simply down-scale the input matrix or tensor after applying non-linearity.

In neural accelerators such as Google TPUs[4] [61], the convolution operations are rearranged as large matrix multiplications. Intrinsic characteristics of neural computations, such as data locality, heavy parallelism, fault-tolerance, etc., have been exploited to reduce energy consumption [57].

2.3.3 Matrix multiplication

Matrix multiplication is a fundamental operation in both MIMO and DNN technologies [60, 17], while it possesses a key role in more traditional super-computing applications from weather forecasting to finite-element method based analyses in mechanical engineering. This has encouraged us to focus on designing energy-efficient matrix accelerator solutions that forms the topic of Chapters 3 and 4 in this thesis.

In linear algebra matrix multiplication it is defined as follows: consider A as an $N \times K$ matrix and B is an $K \times M$ matrix

[4]Tensor Processor Unit

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & ... & a_{1K} \\ a_{21} & a_{22} & ... & a_{2K} \\ \vdots & ... & \ddots & ... \\ a_{N1} & a_{N2} & ... & a_{NK} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & ... & b_{1M} \\ a_{21} & a_{22} & ... & b_{2M} \\ \vdots & ... & \ddots & ... \\ b_{K1} & b_{K2} & ... & b_{KM} \end{bmatrix}. \tag{3}$$

Multiplication of A in B results in an $N \times M$ matrix, as following

$$C_{N,M} = A_{N,K} B_{K,M} = \begin{bmatrix} c_{11} & c_{12} & ... & c_{1M} \\ c_{21} & c_{22} & ... & c_{2M} \\ \vdots & ... & \ddots & ... \\ c_{N1} & c_{N2} & ... & c_{NM} \end{bmatrix}. \tag{4}$$

where every element of C is the sum of element-wise multiplication of row vectors of A by column vectors of B as follows

$$c_{ij} = a_{i1} \times b_{1j} + a_{i2} \times b_{2j} + ... + a_{in} \times b_{nj}. \tag{5}$$

The computational complexity of all known matrix multiplication algorithms are more than quadratic. For multiplying two $N \times N$ matrices the textbook algorithm requires N^3 multiplications and $(N-1) \times N^2$ additions, resulting in a computational complexity of $O(N^3)$. Due to its regular control flow that favors hardware implementations, we base our work on this basic algorithm. Nevertheless, we recognize the existence of matrix multiplication algorithms with lower computational complexities, e.g. , the Strassen algorithm that achieves a complexity of $O(n^{2.8074})$ [62].

2.4 Algorithm based fault tolerance

The proposed voltage scaling solutions in Chapters 3 and 4 rely on Algorithm Based Fault Tolerance (ABFT) as the error detection mechanism. The idea of ABFT, proposed by Huang and Abraham [63], initially described a low-overhead technique to handle computational errors in matrix multiplication operations carried out by supercomputers. Subsequently, the idea was generalized to other linear algebra operations, including QR decomposition, convolution and Fast Fourier Transform [64, 65].

The fundamental idea of ABFT for matrix operations is to augment input matrices with a *checksum property* which reveals computational errors in the final results. Assuming A is an $N \times N$ matrix, a *row checksum* matrix A^r is defined as a $N \times (N+1)$ matrix, as below [66]

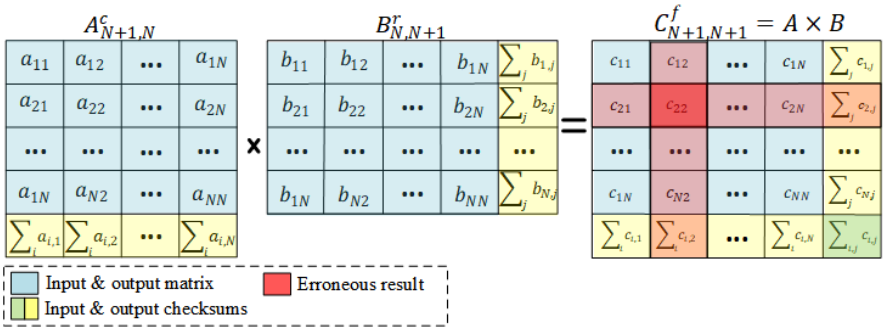

Fig. 12. A checksum row and column helps to detect the errors through identifying mismatches. Reprinted, with permission, from Paper [6] © 2021 IEEE.

$$A^r = \begin{bmatrix} A & Ae^T \end{bmatrix}. \tag{6}$$

where $e_N = [1, 1, ..., 1]$. The nth element of the vector A^r is equivalent to the sum of the corresponding row in matrix A, i.e. $a^r_{n,(N+1)} = \sum_i^N a_{n,i}$. Similarly *column checksum*, A^c, and *full checksum*, A^f matrices are defined as Equation 7 and 8, respectively

$$A^c = \begin{bmatrix} A \\ eA \end{bmatrix}, \tag{7}$$

$$A^f = \begin{bmatrix} A & Ae^T \\ eA & eAe^T \end{bmatrix}. \tag{8}$$

The multiplication result of matrices $A_{N\times N}$ and $B_{N\times N}$ is $C = A \times B$. Multiplying a row checksum matrix A^r and a column checksum matrix B^c results in a full checksum matrix, as shown in Equation 9. As depicted in Fig. 12, the exact location of an error can be detected and even corrected by intersectioning the corresponding row and column checksums. Other basic linear algebra operations preserve the checksum property in a similar manner [66]

$$\begin{bmatrix} A \\ eA \end{bmatrix} \begin{bmatrix} B & Be^T \end{bmatrix} = \begin{bmatrix} AB & ABe^T \\ eAB & eABe^T \end{bmatrix} = C^f. \tag{9}$$

The basic form of ABFT supports the correction of only one erroneous element per row and column. However, in our solution only the error detection capability of ABFT is utilized.

In the past, fault-tolerant techniques such as Triple-Module-Redundancy (TMR) or Double-Module-Redundancy (DMR) have been utilized for designing robust processors

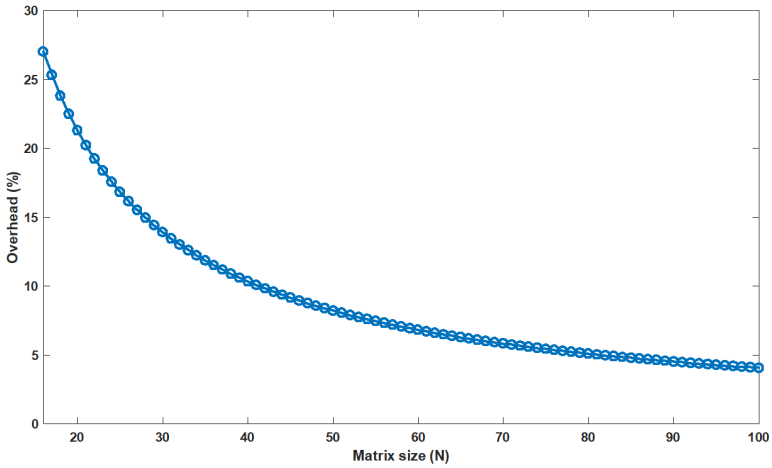

Fig. 13. Overhead of ABFT computations for *error detection* in matrix-matrix multiplication with size of N. The overhead ratio declines with larger matrices.

[67]. Those approaches have substantial overheads that severely penalize their usage for low-power applications [67, 68].

2.4.1 Overhead analysis of ABFT

Employing ABFT for detection of errors in $N \times N$ matrix-matrix multiplication increases the number of arithmetic operations from $(2N^3 - N^2)$ to $(2N^3 + 5N^2 + 4N)$. Furthermore, checking for errors by inspecting the row and column checksums requires additional $2N^2$ summations and $2N$ comparisons. This totals in $(2N^3 + 7N^2 + 6N)$ operations.

The overheads from ABFT for large matrices, e.g., 128×128 are only 2%. In the case of 32×32 matrices, the overheads are 8% which could be justified by energy savings if even a minor reduction of operation voltage is enabled. The ABFT overhead ratio is plotted for different matrix sizes in Fig. 13. For storage, the overhead is simply an extra row or column.

It has been shown that integrating ABFT into the convolutional layers of DNNs introduces around 7% to 8% overheads [64]. Unfortunately, current ABFT technique is not applicable to the pooling and activation layers, but their associated delay paths are short, and their share of computational resources is negligible [64, 69]. Activation and pooling layers are essentially non-linear while current ABFT technique of Huang and Abraham [63] is applicable only for linear operations.

2.4.2 Similar approaches

Established error detection or correction techniques such as Reed-Solomon [70] or Hamming codes [71] provide strong data protection against noise and distortion added in the transmission channel or storage device. However, they are not applicable for protection against computational errors.

A similar technique to ABFT that protects against computational errors is the Result-Checking method [72]. Consider the following simple matrix multiplication

$$C = A \times B. \tag{10}$$

After completion of each matrix multiplication, vectors R_1 and R_2 are calculated from Equation 11 and Equation 12, respectively, where r is a random vector. Any difference between R_1 and R_2 indicates an error in the computations

$$R_1 = C \times r, \tag{11}$$

$$R_2 = A \times (B \times r). \tag{12}$$

Assuming N as the matrix size, the result-checking method has the computational overhead rate of $O(N)$, i.e., its overhead grows linearly with the matrix size [72]. In contrast, the overhead rate of ABFT is the sub-linear $O(1/N)$ [72]. Moreover, result-checking requires computation to complete detecting the errors, which means extra memory round-trips that make it slow and energy consuming. However, ABFT can be adapted to operate on-the-fly in the hardware structure that is presented later in Chapter 3.

3 Energy efficiency through voltage scaling in commercial FPGAs

An application such as an audio processing algorithm can be implemented on different computing platforms ranging from General Purpose Processors (GPP), to Digital Signal Processors (DSP), Application Specific Instruction-set Processors (ASIP) and Graphic Processing Units (GPU) to Field Programmable Logic Gates (FPGA) and Application Specific Integration Circuits (ASIC). Each of these platforms offers different trade-offs in terms of performance, flexibility and cost of development.

As shown in Fig. 14, GPPs offer the highest flexibility where the applications simply are written as high level language software, and through highly developed compilers are translated into micro-operations supported by the processor [73]. This level of flexibility incurs overheads that substantially sacrifice the energy efficiency [74].

At the other end of the spectrum, ASICs provide the highest performance and energy efficiency by implementing the application as integrated circuit logic. However, the cost and duration of development for the manufacturing of an ASIC is high [5], e.g.1000x times higher than the cost of implementing using an off-the-shelf GPP [74].

FPGAs are a specific type of integrated circuit that can be configured for any custom digital design. They are next to ASICs in performance, while they offer more flexibility by providing reconfigurable logic gates and storage blocks, configurable routing and various hard-blocks which enable implementing designs at far lower development costs than with ASICs. In particular, thanks to the introduction of High Level Synthesis (HLS) tools, FPGAs have gained a traction as flexible programmable compute devices. Furthermore, adoption of FPGAs into cloud data-centers [75] as accelerator units fuels their popularity.

Unfortunately, the energy efficiency gap between FPGA and ASIC designs remains significant, e.g., up 10x. Designs employing large FPGAs, such as Startix 10 or UltraScale+ series can consume tens of watts [76, 27, 75].

Considering design margins imposed by manufacturers, dynamic voltage scaling without performance loss is an attractive solution to improve the competitiveness of FPGAs. In this chapter, a novel voltage scaling scheme that leverages the ABFT technique in an FPGA based design is proposed and investigated.

[5] In mass production these trade-offs are subject to change.

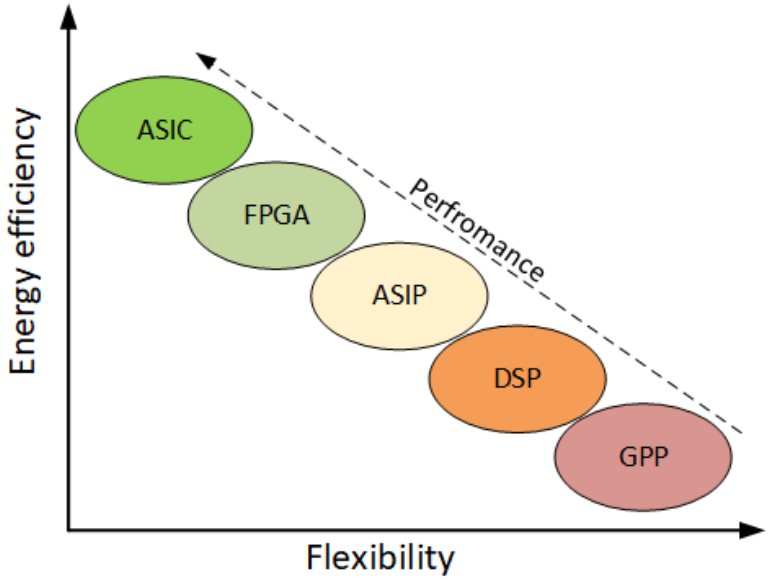

Fig. 14. Trade-offs with respect to energy efficiency and flexibility exist for various platforms.

3.1 Design margins in commercial FPGAs

The margins on top of the “best-case” operating supply voltage guarantee correct functionality for all manufactured chips. The voltage margins of commercial devices have been investigated in, e.g., [31, 28, 45, 77].

A study on FPGA operating voltage reduction [78] demonstrated large guard bands, but did not provide a solution for online detection of errors and voltage adjustment. This is necessary, as the voltage reduction needs to be halted when the errors start to appear [79]. In a similar work [28], the impacts of voltage reduction in memory blocks of commercial FPGAs and the utility of ECCs as error detection mechanism were studied. However, the power consumption of memory blocks is outweighed by other sections in FPGAs.

Ahmed, et. al [45] proposed a more holistic strategy for FPGA voltage reduction: the voltage-frequency dependency of the design is characterized at every power-up. The calibration process is carried out in an "offline" manner before the actual application is started [45]. The calibration bit-stream is generated in accordance with the design and the results are stored on the FPGA as a frequency-voltage-temperature lookup table. Energy savings of 40% and 25% performance improvement were reported. Though their approach addresses process variations, aging and slow temperature variations, it is

inaccurate for fast ambient variations, so errors between the calibrations can escape detection.

The ease of integrating voltage scaling schemes to designs is another important issue to consider [80]. To the best of our knowledge, there is no straightforward method supported by CAD tools to introduce aggressive DVS to FPGA based designs. The majority of techniques proposed in the literature are rather complex "hacks" of the CAD tools [44].

However, recently, FPGA vendors have started shifting toward supporting dynamic voltage scaling. [45]. The approaches introduced by Xilinx and Intel (Altera) both rely on testing during manufacturing. For example, the “voltage identification” (VID) bit solution introduced by Xilinx is a hardwired bit on the device that indicates whether the device is capable of running at lower than nominal and still deliver the same performance. *SmartVID* is another solution introduced by Intel, briefly discussed in Chapter 2. The temperature and voltage range coverage of these techniques is limited [44].

In contrast, in this thesis, a software-based approach is proposed. It is directly applicable using off-the-shelf HLS tools and enables aggressive reduction of FPGA operating voltage, without trading off performance or reliability. The efficacy of the solution is demonstrated for matrix-matrix multiplication operation.

3.2 FPGA architecture and development tools

A simplified scheme of an FPGA is depicted in Fig. 15. The fundamental elements are Look-Up Tables (LUT) shown in Fig. 15(a) that can implement any logic function by setting the SRAMs that determine a truth-table. The LUT units support sequential logic functions by configuring the associated SRAMs multiplex the signal path to registers. By wiring together LUT units, more complex logic functions are realized [81].

Although complex arithmetic functions, such as floating-point Multiply-Accumulation (MAC), can be realized using the LUTs, they are commonly implemented as non-configurable hard-blocks to minimize area and power overheads. These Digital Signal Processing (DSP) units are shown in Fig. 15(b). The energy efficiency and performance of FPGA designs dramatically improve with their proper utilization [81].

In FPGAs, basic RAM blocks consisting of banks of SRAM cells are used to build memory sub-systems. Those RAMs are separate from the SRAMs used for configuring the LUTs, and may even be powered from a different supply voltage rail.

The Input-Output (I/O) circuitry connects the design to the outside world. The I/O and associated auxiliary circuits provide for interfaces such as bit-serial connectivity, clocking management and mixed-signal functionalities [81]. All the elements

are connected through a configurable mesh of wires that runs on top of everything, Fig. 15.(c).

The power to different components might be supplied through different voltage rails, e.g., VCCAUX for I/O and clock, VCCINT for internal logic circuits and VCCBRAM for BRAMs as shown in Fig. 15.

3.2.1 Zynq system on chips

Zynq and Zynq Ultrascale+ Programmable System on Chips (SoC) are a class of programmable devices. They pair a multi-core ARM microprocessor and an FPGA on a single integrated circuit chip [82, 83].

Each section of the Zynq SoC is optimized for specific types of tasks. The ARM microprocessor or "Processing System" (PS) provides for versatility that fits general purpose computing and control tasks. The "Programmable Logic" (PL) is utilized to implement accelerators for the PS. Both sections work in tandem to meet the needs in embedded systems by simultaneously providing flexibility and performance [80].

The provider of Zynq SoCs, Xilinx, has introduced HLS toolsets for programming both the PS and PL parts. The solution proposed in this chapter was developed using those tools.

3.2.2 High level synthesis for Zynq

The traditional implementation approach with FPGAs was to write the application using Hardware Description Language (HDL) such as VHDL or Verilog [84]. While providing for high flexibility, the HDL languages are challenging to learn and use compared to high-level procedural programming languages such as C/C++. Aiming at reducing the development workload in digital hardware design, HLS tools have been created and advocated as an alternative by chip vendors. They increase productivity and ease the development process by adding more abstraction in hardware development. The HLS tools transform high level code written in, e.g., C, C++, or SystemC into a Register Transfer Level (RTL) implementation that can be synthesized to an FPGA in the manner shown in Fig. 16.

Using HLS tools the functional correctness validation is much faster than with HDL based development. Furthermore, it is easier to explore the design space by creating different implementations from the the same source code, each targeted at different performance, area and power specifications. Figure 16 compares the traditional FPGA tool and the HLS based FPGA tool flows [84].

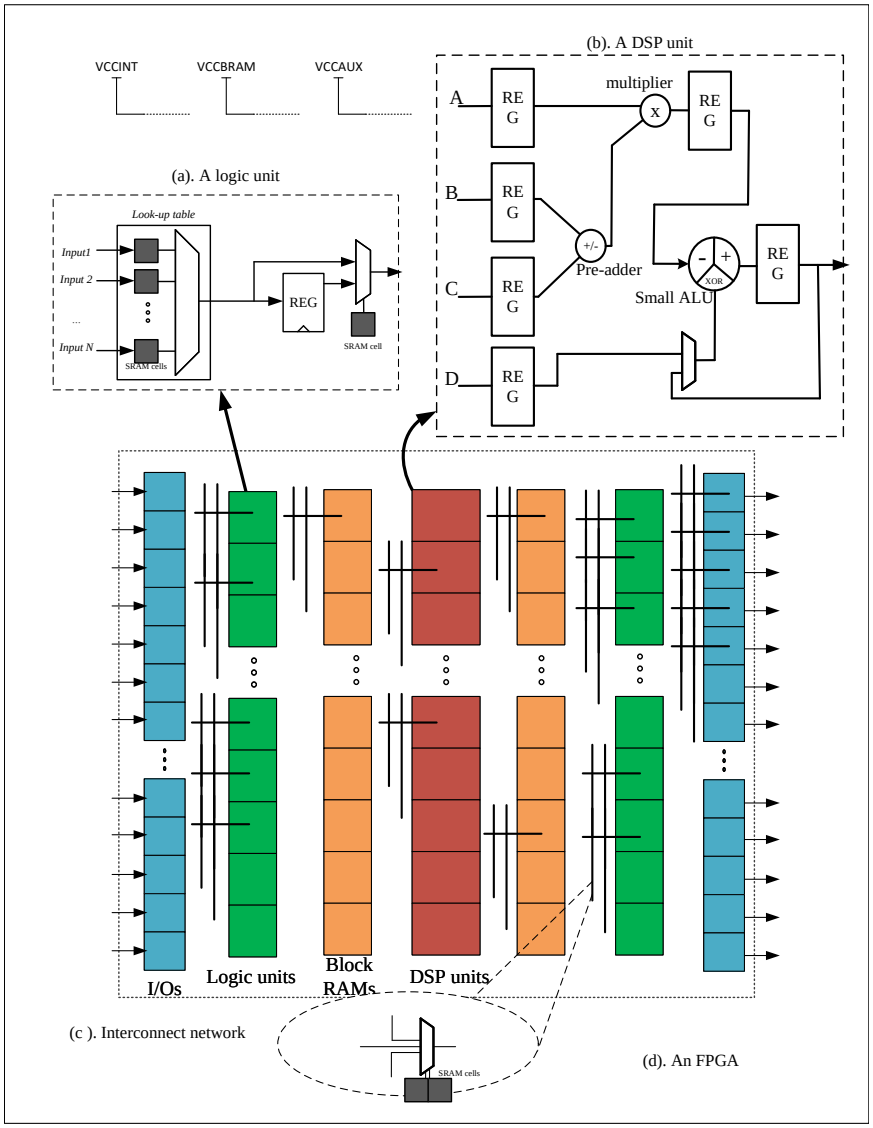

Fig. 15. Internal components of a typical FPGA.

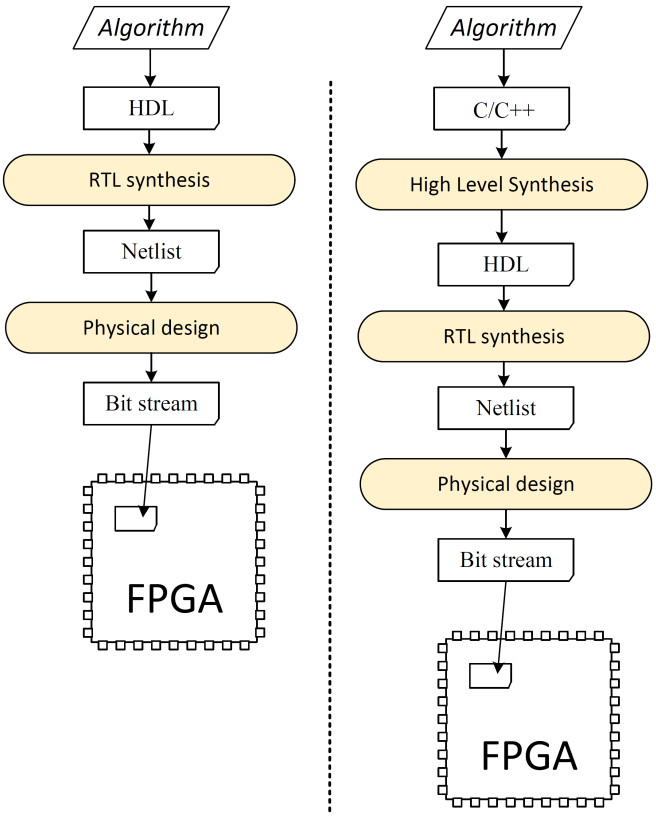

Fig. 16. Traditional hardware design tool flow and the HLS based design tool flow.

In the case of Zynq SoC, a compute intensive parallel sub-routine of an application can be developed as a C procedure that is compiled into HDL code. The hardware is implemented on the PL side as an accelerator engine accessed by the program running on the PS side. The interfacing code can be written in HLS as well [8].

Xilinx provides tools such as Vivado HLS and SDSoC that can automatically generate RTL from C/C++ codes [85, 86]. Furthermore, by using tools such as SDSoC [86], the whole application can be written in C/C++ language. Depending on the target platform, the tool compiles the different sections of the program for the PS and PL, and takes care of interfacing, memory and connectivity within the SoC [87, 86]. Similar HLS tools exist for Intel FPGAs as well.

3.2.3 PMBUS and modern voltage regulators

Due to the need for power management purposes, FPGA vendors support using programmable voltage regulators with adjustable voltage outputs [88]. In our Xilinx Zynq board for example, board-level voltage regulators supply the chip through different voltage-rails. The Power Management BUS (PMBUS) standard [89] is supported by an alliance of multiple vendors to monitor and control the power supplies. With PMBUS, powering can simply be controlled via a set of standard software commands [88, 8].

A PMBUS supported power supply accepts specific Power Management (PM) commands via I2C serial communication protocol. In our experimentation, the evaluation boards used were equipped with Texas Instruments' voltage regulators that support PMBUS. The connection to voltage regulators was via the Zynq device and a PC [82].

3.3 Operating FPGAs at reduced voltages: a novel approach

Operating at a reduced voltage increases the probability of timing errors in digital logic. Safe minimum voltage is device specific and subject to ambient variations. To detect computational errors, and to trigger adjustment of the voltage accordingly, we propose utilizing an ABFT technique. This relaxes the need for costly hardware based solutions such as TED systems, or extensive testing.

The concept of the proposed solution is depicted in Fig. 17. The accelerator is delegated with a compute intensive power-hungry task, while a smaller low-cost processor controls the process. The accelerator operates at a reduced voltage that is adjusted according to the error detections. The errors are detected using an auxiliary, very low overhead and cost inspector programmable processor, which also governs the clock rate of the digital accelerator.

Unlike the state-of-the-art methods, this approach requires minimal circuit modifications. In theory, it can be applied on any digital processor already manufactured. The solution is software-based, requires only minor changes in firmware, and is blind to the underlying architecture.

The ABFT based scheme was employed to detect timing error related inconsistencies in computations. The proposed method was implemented on a commercial FPGA without fabrication changes, and 2.5x faster clocking in addition to a 20% voltage reduction was achieved.

The drawback of utilizing ABFT for an error detection mechanism is that it is limited to specific algorithms. However, the targeted matrix arithmetic operations are fundamental in neural network computations and wireless communications.

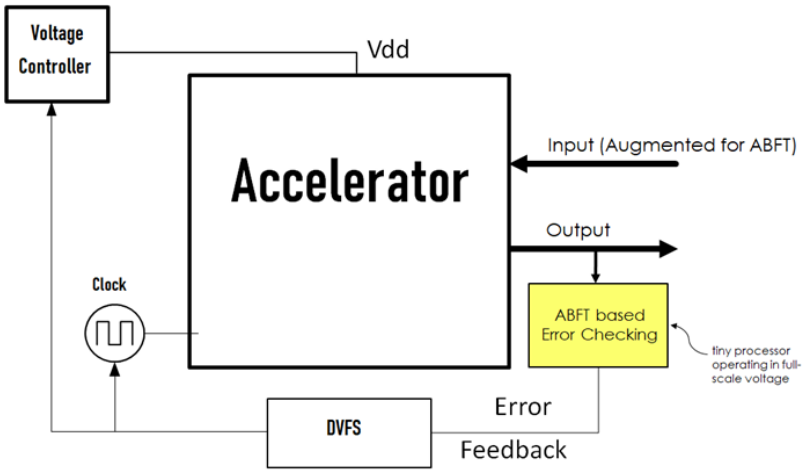

Fig. 17. System level error detection enabling aggressive voltage down-scaling.

3.4 Methodology and experimentation

Voltage margins of a Xilinx Zynq-7000 SoC ZC702 platform were studied through experimentation using an XC7Z020 evaluation board [90]. The design on the PL side of the SoC (FPGA) acted as "Accelerator" , as shown in Fig. 17, while the PS executed "ABFT Error Checking" and "DVS control". An 32×32 matrix-matrix multiplication was implemented on the FPGA using HLS tools, following documentation [91, 92] from Xilinx.

The evaluation board is equipped with digital voltage regulators (Texas Instruments UCD7242) that support PMBUS commands. The regulators support twelve voltage rails that supply various components in the Zynq SoC. Since, in this study, we targeted voltage scaling of the FPGA (PL side), three voltage rails, VCCINT, VCCBRAM and VCCAUX, were subjected to adjustments. The VCCINT supplies all the internal circuitry of the FPGA (LUTs, DSPs, interconnect network, etc.) while the VCCBRAM powers the Block-RAMs. The auxiliary circuits are powered by the VCCAUX. The voltage rails are controlled by transmitting PMBUS commands through serial communication from the host PC or the PS to the voltage regulators [89]. The setup is shown in Fig. 18.

The pseudo-code of the operation is given as Algorithm 1, where A^c and B^r are the ABFT augmented input matrices and C^f is the full checksum matrix. Inspection of the checksum property in the result after each multiplication exposes computational errors.

The Vivado timing analysis tool recommended a 120 MHz clock rate for the ABFT augmented matrix-matrix multiplication hardware. After validation of the

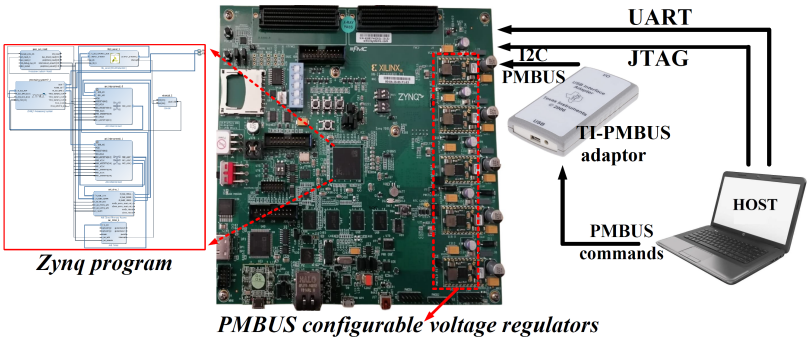

Fig. 18. Experimental setup using a PMBUS adaptor to reduced voltage of the FPGA. Reprinted from Paper [8]. Reprinted, with permission, from Paper [8] ©2021 IEEE.

Algorithm 1 The pseudo-code for error detection

```
Augment matrices to A^c and B^r
Compute C^f = A^c × B^r on the FPGA
Inspect checksum properly of C^f
if (checksum property in C^f is violated) then
    error_detected()
else
    no_error_detected()
end if
```

implementation at nominal voltage and frequency, the clock of the PL was increased to 250 MHz to force the FPGA to generate errors to be studied in the experimentation.

The main voltage rails of the FPGA in the Zynq SoC were adjusted separately, and after each change, the error rate and power dissipation were measured. Finally, the voltage rails were adjusted at the same time.

3.4.1 Scaling voltage of internal logic circuitry

The internal logic circuitry in the PL consumes the largest share of power. In our experiments, the VCCINT voltage could be reduced by around 23% percent from the nominal to ≈ 770 mV without any errors. This translated into about a 50% cut in power dissipation, as depicted in Fig. 19. Notice, since the clock is not changed while the voltage is being reduced, energy and power reduce on the same pace.

Temperature change as an ambient variation factor was introduced in the experiments using a thermal chamber built for the evaluation board. A heat gun was used for warming

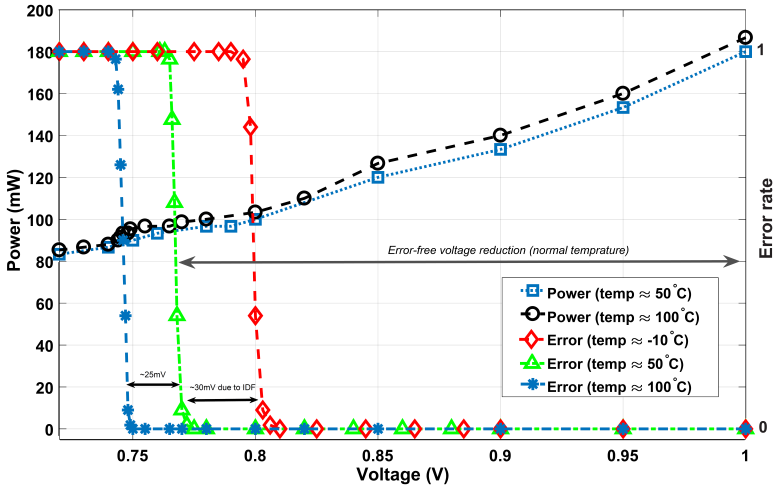

Fig. 19. Power consumption and error rate of the PL part of an FPGA running an matrix operation. Reprinted, with permission, from Paper [8] ©2021 IEEE.

up the SoC to $\approx 110\ ^{\circ}$C. To study the low-temperature behavior down to $\approx -10\ ^{\circ}$C cool-spray was employed. The temperature of the chip was measured by reading the internal temperature sensor of the SoC. The measured results in hot and cold ambient temperatures are shown in Fig. 19 as well.

Interestingly, with lower temperature, errors start to appear at higher voltage. This observation complies with the inverse temperature dependence (ITD) phenomenon [93] reported in [94].

3.4.2 Scaling voltage of BRAMs

Block-RAMs (BRAMs) of FPGA are substantial energy sinks [95, 96]. In the experimentation, the VCCBRAM voltage was adjusted, while the other voltage rails were held fixed. Although the effectiveness of Error Correction Codes (ECCs) is well established for data protection in memory systems [95, 77], we investigated the utility of ABFT when the matrices to be multiplied are stored in BRAMs. Row checksums were added to matrices of different sizes before committing them into the BRAMs.

In this context, the advantage of the ABFT over ECCs is the ability to detect both computing and memory errors. Admittedly, pairing anABFT scheme with a light ECC might provide even more robustness, but at the cost of higher power dissipation.

For experimentation, a design that utilized 91% of available BRAMs was programmed on the FPGA. Firmware was written on the PS, which is in separate voltage

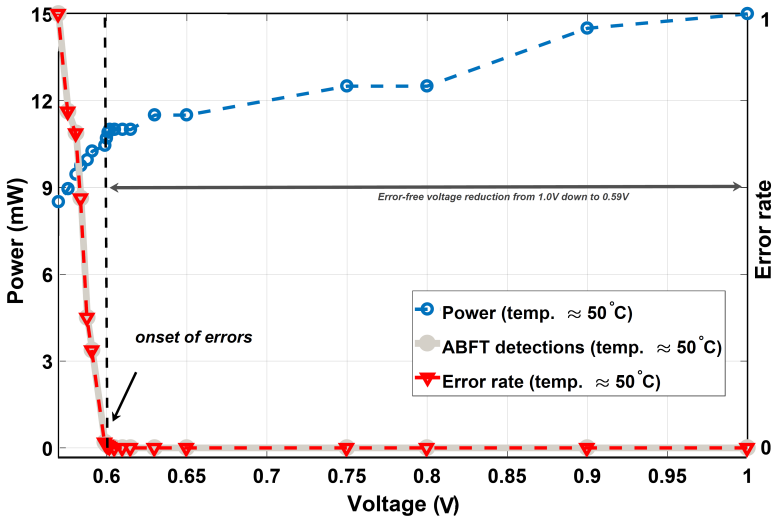

Fig. 20. BRAM error rate and power consumption versus voltage. Reprinted, with permission, from Paper [8] ©2021 IEEE.

domain, to access the BRAM content constantly and inspect checksums, while the VCCBRAM voltage was scaled down to 0.5 V, just before the connection from the host PC to the FPGA was lost. The encoding, i.e. checksum augmentation, and later error checking was carried out by the PS side in a procedure similar to Algorithm 8.

The BRAM power dissipation was halved when the voltage was reduced from 1.0 V to around 0.6 V, while no errors were detected. The results are shown in Fig. 20. The voltage at which errors emerged is lower than with the internal logic, indicating more robustness against voltage scaling. Our experiments confirmed the results reported in [29].

3.4.3 Scaling voltage of auxiliary circuits

A concern with reduced voltage schemes is that the energy gains might be rendered insignificant when the energy consumption in other components, such as clock generation and distribution network, the Phase Lock Loops (PLLs) and peripherals, are taken into the account [97, 32]. This is a particular challenge for designs operating in the near threshold voltage region [32].

The peripherals, interfacing, clock management, I/O, etc., are categorized as Auxiliary Circuits of the PL and are supplied by the VCCAUX voltage rail in our Zynq

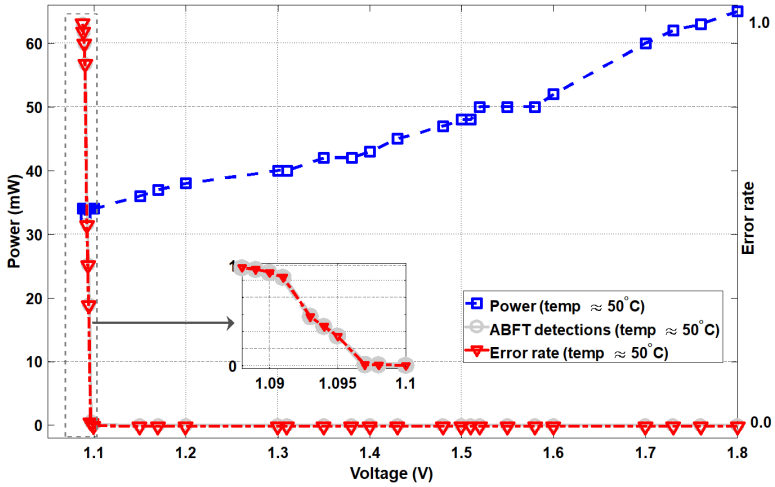

Fig. 21. VCCAUX related error rate and power consumption versus voltage. Reprinted, with permission, from Paper [8] ©2021 IEEE.

platform [82]. In experimentation, the impacts of scaling down VCCAUX voltage on the error and detection rate, and the power savings were explored.

VCCAUX power dissipation against the error rate is shown in Fig. 21. The onset-of-errors voltage is be very close to the crash voltage, and the error rate increases from 0% to 100% over a small voltage interval.

Aggressive powering of auxiliary circuitry may result in an unrecoverable crash due to the short interval between the onset of errors and the crash point. This indicates that the voltages of the SoC should be reduced in a manner such that the errors appear first in computations and BRAM.

3.4.4 Impact of down-scaling of all voltage rails

To experiment on achieving the minimum power/energy dissipation, the voltages of all three domains were reduced by 1% steps from the nominal level. The first errors appeared from reducing the VCCINT, then the VCCBRAM, and finally the VCCAUX. The results are shown in Fig. 22. Scaling down all three voltage rails reduced the power consumption from around $\approx$ 380 mW down to $\approx$ 240 mW. The main share of power is dissipated in the internal logic circuity that is supplied by the VCCINT. Scaling the VCCINT beyond 80% of the nominal causes errors, while PS and BRAMs appeared to tolerate larger voltage drops.

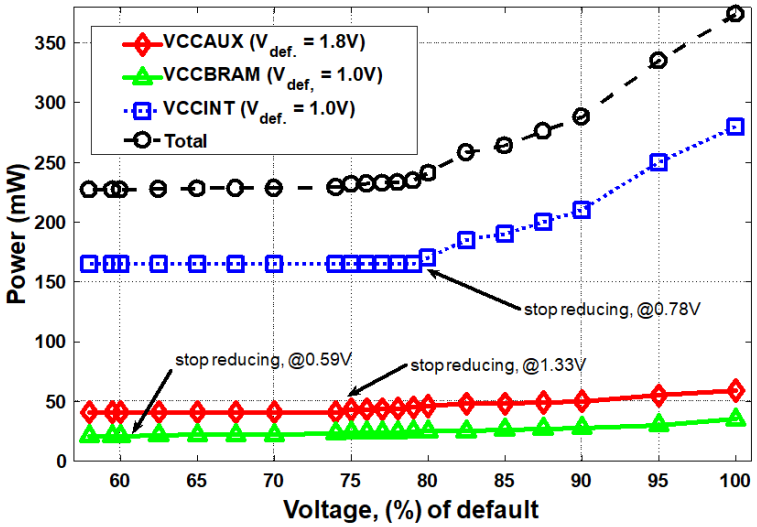

Fig. 22. Combined impacts of voltage reduction in all rails. Reprinted, with permission, from Paper [8] ©2021 IEEE.

Inverse temperature dependence (ITD)

Due to ITD, transistors can operate with higher frequencies when the temperature is increased [93]. Alternatively, as shown in Fig. 19, the voltage at which errors appear depends on the chip temperature: when the supply voltage is much higher than the threshold voltage of the transistors, increasing temperature results in higher circuit delay. However, when the voltage is close to the threshold voltage, increased temperature results in higher conductivity of the transistors [98].

To the best knowledge of the author, there have been few proposals to exploit the ITD either to enhance reliability or energy efficiency [98, 93]. Solutions such as the ABFT approach proposed here may enable exploiting opportunities provided by the phenomenon.

3.5 Application example: neural nets

To examine the opportunities for using ABFT in an application, an ABFT augmented five layer fully-connected neural network was designed and implemented on the SoC. The layers with a large number of nodes incorporated a simple version of ABFT. Then, the number of times that the output of the NN on the FPGA differed from golden output was determined at each step of reducing voltage [99]. A single variable back from the

Table 1. Resource utilization of the implemented NN with simple ABFT detections.

Resource	Estimation	Available	Utilization
LUT	21k	53k	39%
FF	43k	106k	41%
DSP	37	220	17%

PL indicated whether an error was detected by the ABFT. Table 1 presents a utilization report of the implemented design on the Zynq SoC.

The results shown in Fig. 23 indicate a strong correlation between ABFT error detections and wrong output rate of the NN. This demonstrates that ABFT can be used to determine the point of first failure in the NNs, as the onset of errors detected by ABFT occurs slightly earlier.

Neural networks themselves exhibit some error tolerance, i.e., a small number of computational errors may not impact on the final classification outcome [100]. This is why ABFT based error detections could be a useful early warning sign. We anticipate that the results from an FC-NN can be extrapolated to larger DNN models.

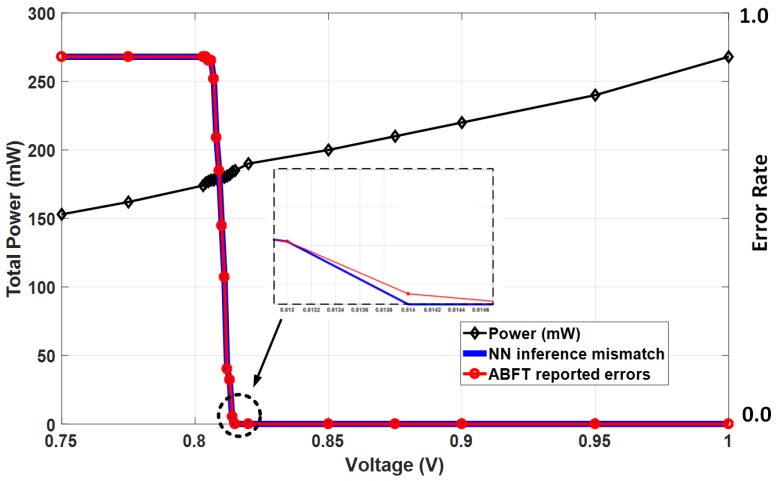

Fig. 23. Combined impacts of voltage reduction in all rails.Reprinted, with permission, from Paper [8] ©2021 IEEE.

Table 2. Resource utilization of the implemented matrix multiplier.

Implementation	Resources	LUTs	FFs	DSPs	BRAMs
Reference [17]	Utilized	842390	1000062	3280	1524
	Available	1033608	2174048	6834	1906
	Percent. (%)	81	46	48	80
This Work	Utilized	15177	14501	160	38
	Available	53200	106400	220	140
	Percent. (%)	27	13	73	27

3.6 Comparison and discussion

The experiments were repeated on another similar evaluation board with the Zynq SoC from a different production batch. Only negligible differences in the error onset voltages were observed.

In the experiments, the throughput of the implementation was 50 000 32×32 full-precision floating point matrix-matrix multiplications per second. Since a row and a column are dedicated to the ABFT checksum, the actual matrix in an $N \times N$ implementation would be $(N-1) \times (N-1)$. Hence, the effective size of our implementation is 31×31, which gives a power efficiency of 6.2 GFLOPS/W. The run-of-the-mill design in this experiment [91], with voltage scaling ability delivered energy efficiency close to highly optimized solutions such as [17] where 11 GOPS/W (18 bit fixed point, Xilinx Virtex-7) was reported, respectively. This indicates a cost effective opportunity to increase the efficiency of designs [101]. Table 2 reports utilization of resources for our design and provides comparison with a similiar design.

Prior proposals exist to shift FPGA designs from fixed to adjustable supply voltages. A comparison of such investigations is given Table 3. Ease of implementation as a design factor was assessed subjectively, and presented as a potential point of comparison.

ABFT possesses promising utility for the detection of errors caused by reduced operating voltage. In the current study, it was used to eliminate voltage margins without performance loss (i.e. clock adjustment), however, in future works the solution could be utilized to pursue Near-Threshold voltage operation schemes that promise manifold improvement in energy efficiency [102, 19]. Other potential uses may be in approximate computing applications to keep the error rate tractable. The application of the proposed approach in signal reconstruction and/or MIMO communications in low-power sensor nodes can be investigated as well.

Finally, approaches such as [104] propose overclocking to enhance energy efficiency through increasing the share of dynamic versus static power consumption. However, overclocking leads to heating of the chip, contributing to more static leakage, while the

Table 3. Comparison with similar solutions.

Solution	[103]	[45]	[27]	Ours
Technique	TED circuits	Offline calibration	ECC	ABFT
Logic errors	$\surd$	x	x	$\surd$
Memory errors	x	x	$\surd$	$\surd$
Implementation	Difficult	Moderate	Easy	Easy

energy per operation does not improve much [104]. Voltage reductions without clocking changes, as in our study, appears to be a more viable scheme to improve the energy efficiency [8].

3.7 Summary

Fairly large voltage margins are specified by the manufacturers on top of the "best-case" operating voltages of digital integrated circuits. Elimination of those margins increases energy efficiency, but requires a means to identify the lowest voltage that guarantees errorless operation. Algorithm level error detection tackles this challenge through online inspection of the results.

The approach was demonstrated and implemented for matrix-matrix multiplications on a Zynq SoC platform, and a $\approx$50% reduction of energy dissipation was achieved. The approach has attractive uses in applications in which occasional errors can be are tolerated, such as deep neural networks and wireless communications.

4 Energy Efficiency through low-voltage systolic structure

Operating near and below the threshold voltage of transistors enables a substantial improvement in energy efficiency [105]. In the previous chapter, we proposed to utilize ABFT as an error detection mechanism to enable safe voltage reductions.

However, due to the limitations of the physical device, we did not explore near-threshold/sub-threshold voltage settings. In the following, a more specific structure that integrates ABFT is proposed and operating points down to sub-threshold voltages are explored through simulations. Moreover, the reliability of ABFT as an error detection mechanism is studied.

4.1 Systolic arrays

In [74] and [106], Horowitz et al., analysed and identified sources of inefficiencies in general-purpose processors. When the energy consumption per each specific operation in a typical processor is profiled, as presented in Table 4 [107, 106], memory accesses appear to be the most expensive ones. A computing architecture that dedicates more resources to "useful" operations would inevitably increase the performance and energy efficiency. These observations call for architectures that minimize overheads in achieving computational tasks [107].

Introduced by Kung,et. al in 80s [109], Systolic Arrays are among the most energy efficient and highest performing architectures. As a domain specific solution [110], they are an excellent choice especially for various matrix operations.

Table 4. Energy consumption of individual operations in a typical processor [108].

Operation	Energy [pJ]	Relative Cost
32 bit int ADD	0.1	1
32 bit float ADD	0.9	9
32 bit Register File	1	10
32 bit int MULT	3.1	31
32 bit float MULT	3.7	37
32 bit SRAM Cache	5	50
32 bit DRAM Memory	640	6.4k

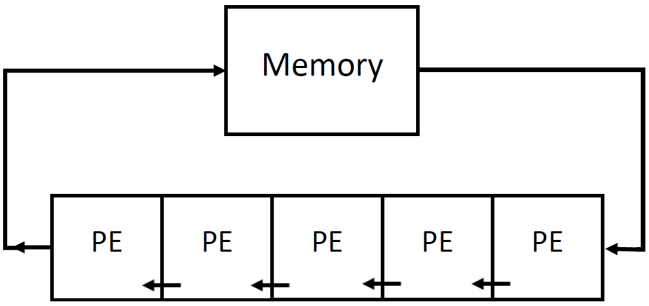

Fig. 24. Internal components of an example PE used in a Systolic Array.

A systolic array consists of several connected Processing Elements (PE) that carry out basic operations. In Fig. 24 and Fig. 25, simple 1D and 2D systolic array structures are shown. Each PE is connected to one or multiple neighbouring PEs. Passing partial results from one PE to another PE within the array removes many unnecessary memory round-trips, which, as shown by Table 4, are among the most energy demanding micro-operations. The massive number of PEs operating in parallel and in pipeline provides high performance as well. Based on the configuration of the array, different algorithms can be defined to be carried out by the systolic processor [111, 112].

Despite substantial performance and energy efficiency gains [53], systolic array based processors were not widely adopted until recently [112]. With the rise of deep neural network applications and the associated need for high performance accelerators, systolic structures have gained popularity. Pioneered by Google, systolic arrays are being used as the core of Tensor Processor Units (TPU) [61] for neural acceleration. Furthermore, recent studies suggest systolic arrays are promising solutions to address the performance demands of future generations of telecommunication standards, e.g., for implementing the computing needed by massive MIMO technologies [113, 114, 53].

As an example, an $N\ timesN$ square matrix multiplication requires approximately $O(N^3)$ multiplication-summation operations. Assuming one operation is completed per cycle, a systolic array of size $N \times N$ needs only $O(N)$ cycles when memory bottlenecks can be neglected.

4.2 Systolic array with integrated ABFT

While operating in the vicinity of the threshold voltage of transistors promises significant energy efficiency improvements, the cost is large performance loss due to a reduced

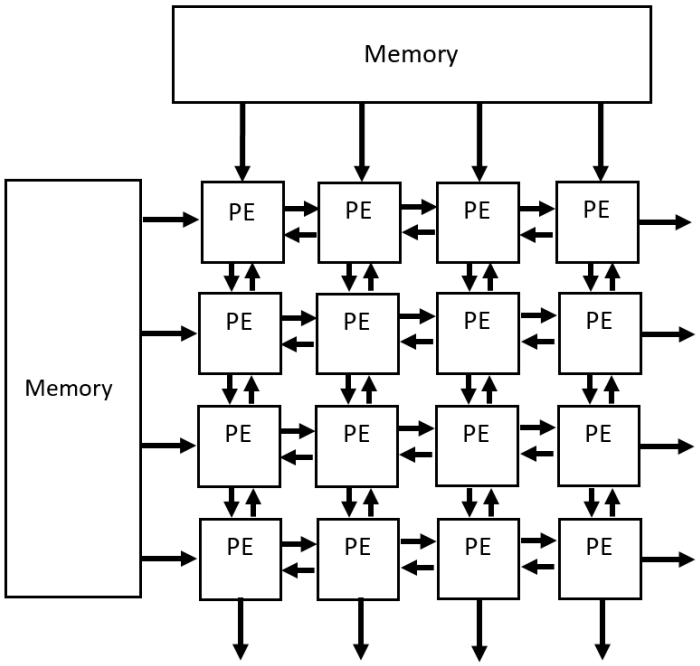

Fig. 25. 2D systolic array for matrix operations.

clock rate, which might be scaled down to 1%. Furthermore, designing for extremely low voltages is challenging due to the exacerbated impact of process and ambient variations. We propose tackling the reduced voltage performance loss and reliability issues by *i)* adopting a highly parallel systolic array architecture and *ii)* integrating ABFT as the error detection mechanism, as depicted in Fig. 26 for matrix-matrix multiplication.

The idea is to augment the input matrices with ABFT checksums and detect computational errors by inspection of the checksums at the output of the array. Checksum augmentation and inspection do not affect the systolic flow, and are carried out on-the-fly without incurring any extra memory operations.

The proposed arrangement is shown in Fig. 27. Operating as a conventional systolic array [115], the processing elements perform multiply-accumulate (MAC) operations on input data received from the neighbouring PEs. The checksum augmentation of the input matrix is carried out on-the-fly as it enters the array. The input augmentation block is highlighted as the top row in the structure. Notice in the scheme only one of the matrices needs to be augmented with checksums. A column of N PEs is included in the systolic array to perform checksum multiplications. Another column of N integrators

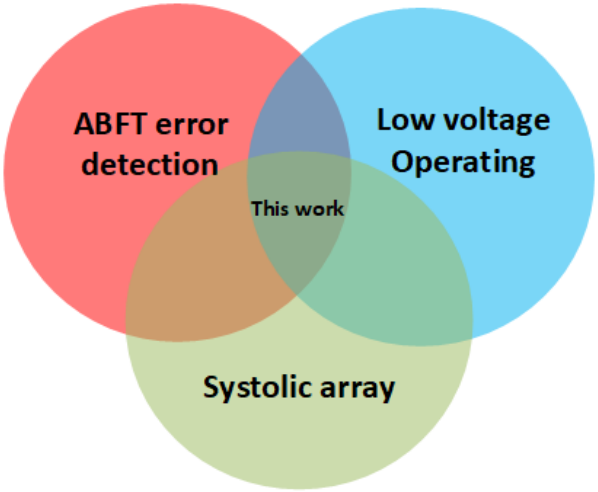

Fig. 26. Technologies that act as complementarities to achieve superior power.

and comparators follows the last one of the systolic array, and contains the ABFT error detection circuity.

The errors are detected within the systolic array in a row-wise manner and on-the-fly. Adding column checksum logic would contribute to reduced error escape rates. However, while that would help in pin-pointing the errors to specific PEs, the overheads would double.

The operation flow of the proposed scheme is illustrated in Fig. 28 for a 4-by-4 array. In each clock cycle, one rearranged row and column of each input matrix, A and B, enter the array. The arrangement of matrix entries has been modified slightly with zero-padding following the original systolic data flow [112]. The checksums of the input matrix are calculated on the fly without impacting the systolic data-flow.

The 4×4 example array has an extra column for checksums, similar to Fig.27. From this figure, the ABFT error detection logic has been omitted. The highlighted column is the checksum column $B_f = [B \;\; Be^T]$ of matrix B_f.

Structure of processing elements

The PEs of the systolic array consist of a few registers for handling data and a MAC unit, as shown in Fig. 29.The multiplier is based on the Wallace tree structure [116], while the adder-subtractor is a ripple-carry design [117].

The MAC units receive two inputs from adjacent (east and north) PEs as *in1* and *in2* at each clock cycle. The inputs are matrix entries that at the next clock are passed to

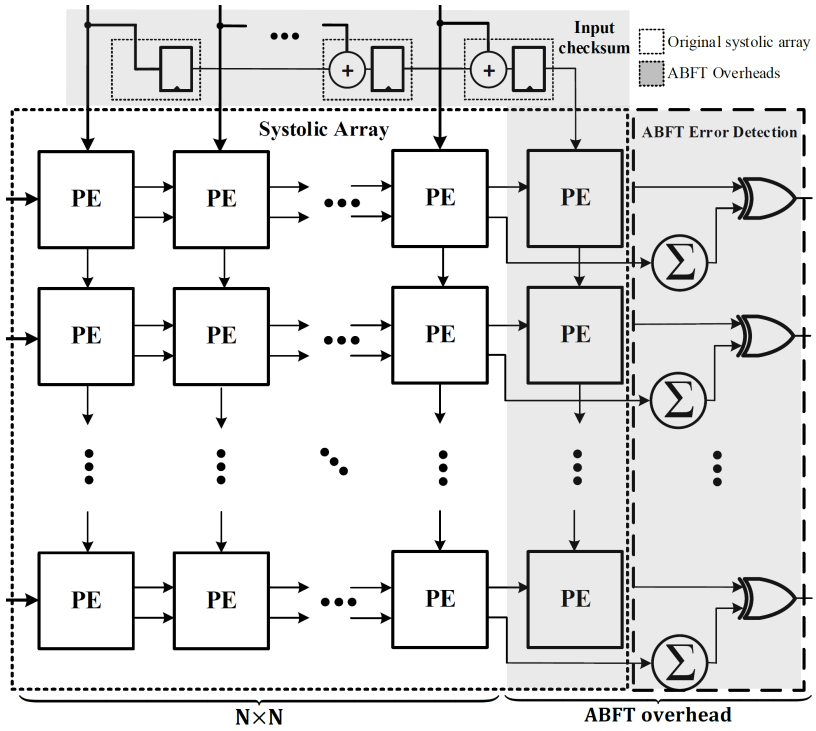

Fig. 27. The proposed systolic structure fully integrates ABFT in its data-flow. Reprinted, with permission, from Paper [6] © 2021 IEEE.

out1 and *out2* to the next PEs. When the calculations are done, the FO bit is set *HIGH* for all PEs, and the output of the accumulator register is multiplexed to the next PE to be stepped out of the array.

4.3 System modeling

The standard approach for simulating synchronous digital circuits employs register-transfer level (RTL) models with abstracted transistor gates [118]. In the context of the current research, the availability of low voltage cell libraries required devising a SPICE[6] model for the PEs [19]. Compared to RTL, SPICE based simulations employ far more detailed models of transistors [119]. The downside is heavy computational demands that limit the number of transistors in the simulated designs. In contrast, RTL simulation can handle millions of transistors in a reasonable time.

[6]Simulation Program with Integrated Circuit Emphasis

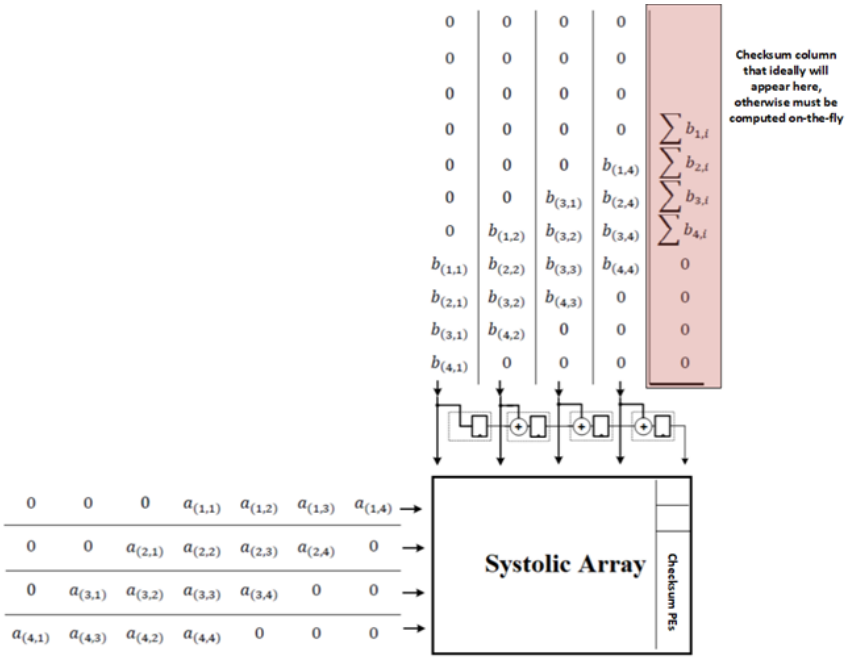

Fig. 28. Illustration of systolic flow of data and checksum augmentation.

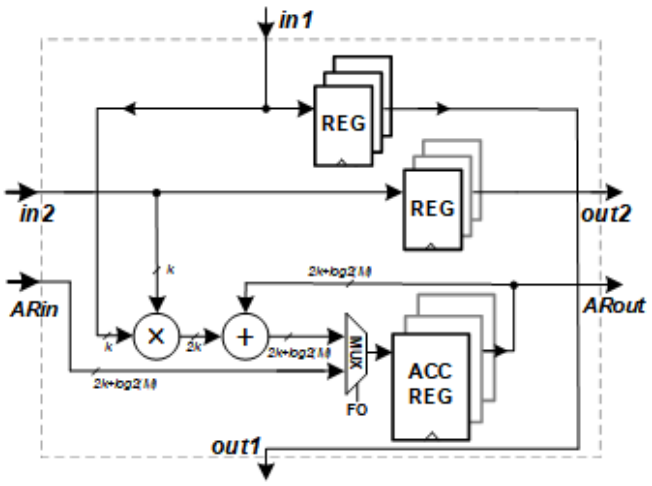

Fig. 29. Internal components of an example PE used in a Systolic Array. Reprinted, with permission, from Paper [6] © 2021 IEEE.

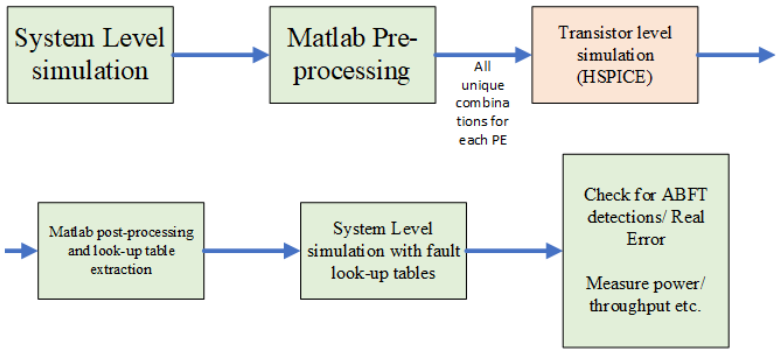

Fig. 30. Simulations incorporate system level and circuit level approaches.

In the current research an HSPICE simulator [120] and MATLAB software were used to implement a co-simulation environment. Pairing the SPICE models of PEs with the behavioral model of the system resulted in fast enough simulations for studying the proposed solution.

The flow of simulations is shown below in Fig. 30 [7]. The system level simulation of a matrix multiplication operation was carried out in MATLAB with random data. The PE SPICE model was developed and simulated using 90nm CMOS technology in HSPICE. All transient data and micro-operations were stored and duplicated computations, i.e., the same operands and operations, were removed. Then the data was converted into an HSPICE vector file, and using HSPICE *Transient* mode simulations, the same operations were carried out in the PEs at different simulated voltages. The output data from HSPICE was stored and returned to MATLAB.

Most of the simulation time was consumed by SPICE; however, MATLAB post-processing of signals generated by HSPICE took considerable time as well. The impacts of clock variations were modeled in MATLAB by sampling the HSPICE generated signals using a variable clock timing by using the original clock samples generated in HSPICE. The clock grid obtained from HSPICE is read in MATLAB and the clock edges were displaced slightly to represent variations of timing. The matrix multiplications were then done again in MATLAB; however, this time the result of each operation was substituted by the one from HSPICE. Then, ABFT inspection was carried out to inspect whether the errors were detected by the solution. Since the circuit complexity of comparators is trivial compared to the MAC unit, they were only simulated in MATLAB.

[7]The computer code or SPICE script of each section can be found in the Github repository at https://github.com/NeuroFan/SystolicArray

4.4 Power dissipation

Using the simulation models, we studied two aspects of the proposed solution: first, the power saving opportunities enabled by operating at reduced voltage, and second, the reliability of ABFT in detecting errors at reduced voltage.

In the real-world, the stabilization time required by the voltage regulator after changes is often a few hundred clock cycles, while the clocking frequency can be changed in a few cycles. Therefore the latter was the first response to the errors detected in the simulations. Admittedly, this experimental setting is not a fully faithful mirror image of a practical environment, but it does, however, suffice for studying the implications of using the proposed low-voltage solution.

In the simulations, we reduced the voltage gradually until errors appeared. Then, frequency steps down were made until a correct operation result was obtained, enabling us to reduce the voltage again. In realistic applications, the clock frequency adjustment scheme might be different, but now to study the power savings opportunities the supply voltage was swept from 0.9 V down to 0.4 V with a small step size.

The power dissipation of the ABFT logic augmented systolic array is presented, with the points, in the upper part of Fig. 31. The supply voltage was gradually reduced by 0.01 V steps, the frequency following based on error detections. The power dissipation of a simulated PE at the nominal voltage $V_{dd} = 0.9\,V$ and maximum frequency $f_{clk} = 400$ MHz was around 74 μW. For simplicity same amount of timing variation was applied for all voltages.

The objective of the experiment was to understand the energy savings versus voltage by sweeping across all of the voltage range. The peaks in Fig. 31 are re-computation penalties after error detections.

The "goodput" depicted in the lower part of Fig. 31 is throughput from which erroneous results are omitted. Small timing variations $(\sigma_T/\mu_T = 0.5\%)$ proportional to the clock period were added to account for PVT variations, following the approach in [121].

Reducing the voltage from 0.9 V to 0.7 V saves almost half of the energy of matrix multiplication without performance compromise. In other words, there is a significant timing margin in our design, similar to those defined by the chip vendors. Lowering the voltage beyond 0.7 V requires reducing the clock frequency. In the near-threshold region, when the voltage approaches around 0.5 V, the energy use is reduced by a further 70%, at the cost of 50% reduced computing performance. As can be anticipated for near-threshold operating points, in the vicinity of 0.4 V, the goodput is only 10% of that

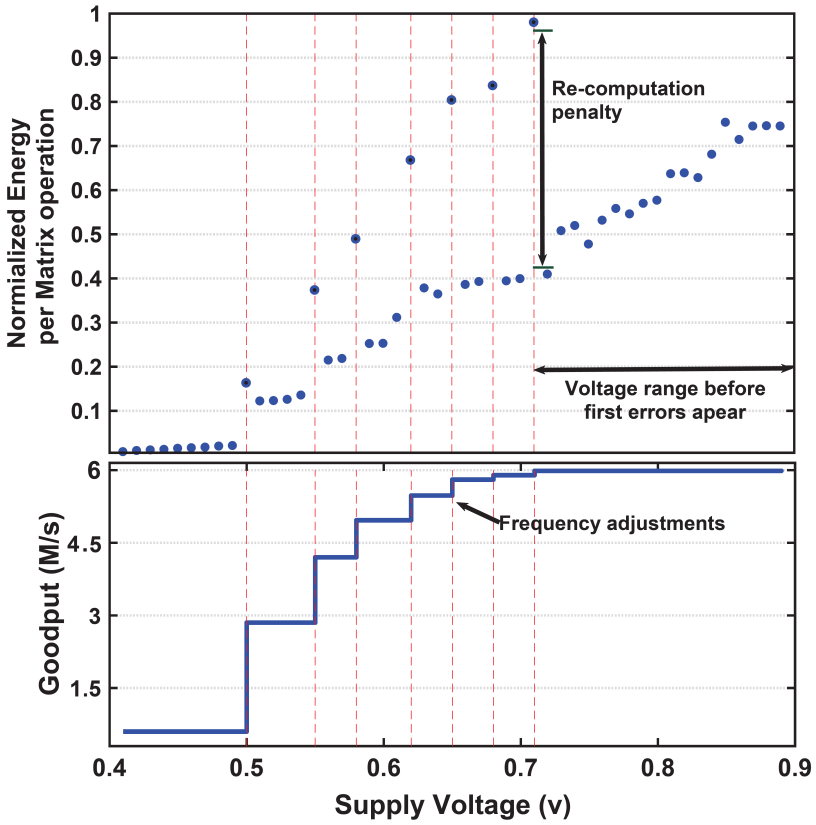

Fig. 31. Top: Energy consumption per matrix multiplication. Bottom: The "goodput" that correlates to operating frequency. Reprinted, with permission, from Paper [6] © 2021 IEEE.

at 400 MHz and 0.7 V, but with around 10% of energy dissipated per operation, see Fig. 8.

4.5 Reliability and overheads

The usefulness of the proposed voltage reduction solution depends on the reliability of error detection. The errors in the results that remain undetected by the ABFT checksum inspection are called "silent errors". Those occur when two or more results on a row are incorrect, but cancel out each other. The opposite can take place, too, due to circuit non-idealities, and a false error is reported[8]. However, the energy and performance cost of a false detection is not severe, requiring only an extra re-computation and an unnecessary clock/voltage adjustment.

Silent errors, however, might not be tolerated by applications as the incorrect results could result in catastrophes. Therefore, the silent error rate needs to be analyzed more closely.

4.5.1 Error coverage

In the proposed approach, the matrix row checksums are tested for errors, so the more rows that have errors, the lower the silent error rate becomes. Assuming silent error probability P_i for the ith row, the silent error probability is $SE_{matrix} = \Pi_{i=1}^{N} P_i$. Notice that SE is the probability of a silent error in the matrix, meaning that is the probability when existing errors remain undetected in the first row AND the second row AND so forth. It is enough that an error is detected in any of the rows [63].

If B_i bits are affected by errors in each row, the silent error probability becomes $SE_{row} = 2^{-B_i}$. As the PEs and rows of the array are replicated, we can assume at least n rows are hit by errors. In that case, the probability of an undetected error in the whole matrix is $SE_{matrix} = 2^{-\Sigma_i^n B_i}$. We may conclude that the probability of silent errors is extremely small.

In telecommunications, a silent error might result in a re-transmission due to a packet error. Moreover, in many applications multiple matrix operations are carried out consecutively, e.g., in DNN inference. Assuming a silent error rate of s, the final silent error rate in the case of q matrix multiplications would be s^{-q}.

[8]Notice the numbering format is integer here, in the case of a floating-point, mantissa truncation might cause false detections as well

For example, AlexNet inference requires 22 000 of 32×32 matrix multiplications. If the silent error rate is very high, say 50%, the probability of a silent error in an inference episode is less that 2^{-22000}.

4.5.2 Simulation based error coverage estimates

The developed co-simulation model was employed to provide estimates on the possible silent error rates with respect to Process-Voltage-Temperature (PVT) variations. To characterize silent error in the model, the bit error rate of a PE was determined in HSPICE using a fixed voltage ($V_{dd} = 0.7$ V) and clock period ($f_{clk} = 280$ MHz) with clock jitter representing all random variations. As can be expected, the MSBs showed higher sensitivity to variations as they are associated with longer delay paths in the Wallace tree and ripple-carry structures. The resulting error probability density was then used to flip the output bits of the PEs.

The results are depicted in Fig. 32. In the top plot, the error rates for output bits of the PE are shown. The blue curve labeled "Total error rate in matrix" denotes the errors that were detected by comparing all the elements of the result matrix with the pre-computed result, so it exposes all the errors, including the silent ones. Notice, that ABFT detects the errors by inspecting only the checksums, i.e. from the last column of Fig. 27.

The red curve labeled "Silent error rate in a row" shows the error rate of undetected errors that occur in a single row. Notice that as more rows become erroneous, the rate of undetected errors in a matrix goes down. The green plot, i.e. "Silent error rate in matrix" is the undetected error rate for the whole matrix (not just a row) and remained zero in our simulations with millions of trials.

4.5.3 Overheads

ABFT adds to circuit logic and power overheads to the systolic array. Figures 33 and 34 show the gate count and power consumption of the extra circuity in the proposed solution for different matrix sizes. For a 32×32 matrix, the ABFT scheme demands 8.5% more logic, and HSPICE simulations indicated 9% increased power dissipation. The overheads estimated for a 128×128 array are 2.2% and 2.5%, respectively.

The energy efficiency gains from the capability to operate close to the threshold voltage significantly outweigh the overheads from ABFT. Furthermore, the overheads scale sub-linearly with the systolic array.

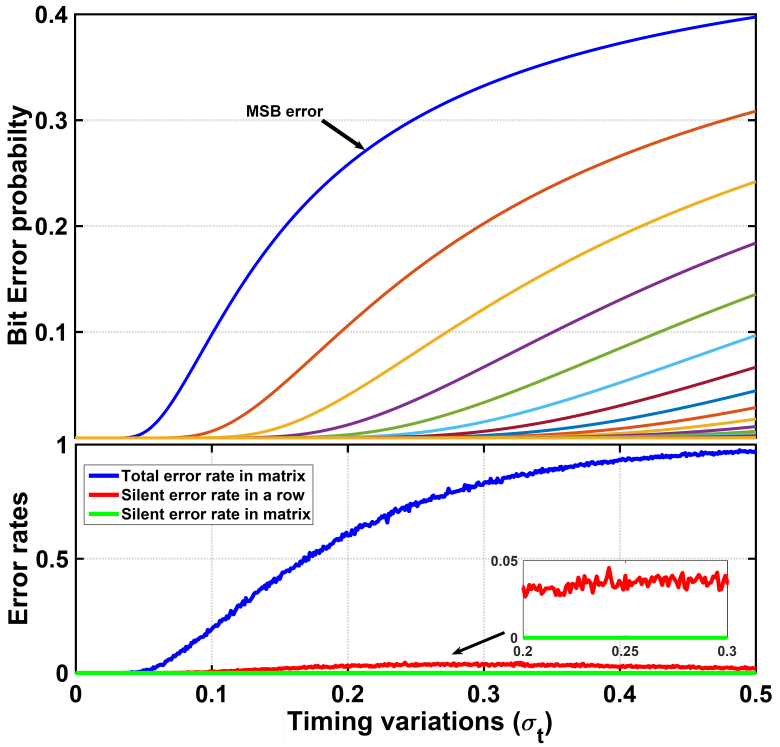

Fig. 32. Top: Error rates of PE output bits. Bottom: Total error and silent error rates for 32 point row-column and 32-by-32 matrix multiplication. The horizontal axis represents clock period variations (σ_T, nanoseconds). Reprinted, with permission, from Paper [6] © 2021 IEEE.

The costs of storing the checksums also needs to be taken into account as the external memory access overheads are much higher than for on-chip-memory [74]. However, the checksums need only a column in an $N \times N$ matrix, translating into $\approx 100/N\%$ memory overheads and are negligible for large arrays. The checksums can be disposed of after their inspection has been carried out.

4.5.4 Overheads comparison with similar works

Implementations of a common TED method, i.e. "Razor" [49] on an ARM processor have shown significant overheads [122, 32]. For systolic processors, Whatmough [123] proposed adding a TED system to the MAC units based on Razor Flip-Flops (RZFF).

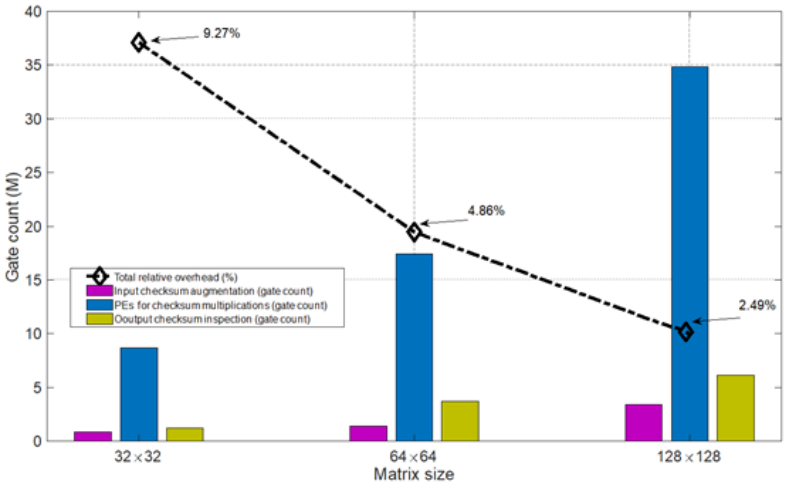

Fig. 33. Comparison of estimate area overheads with regard to design size (based on the SPICE model).

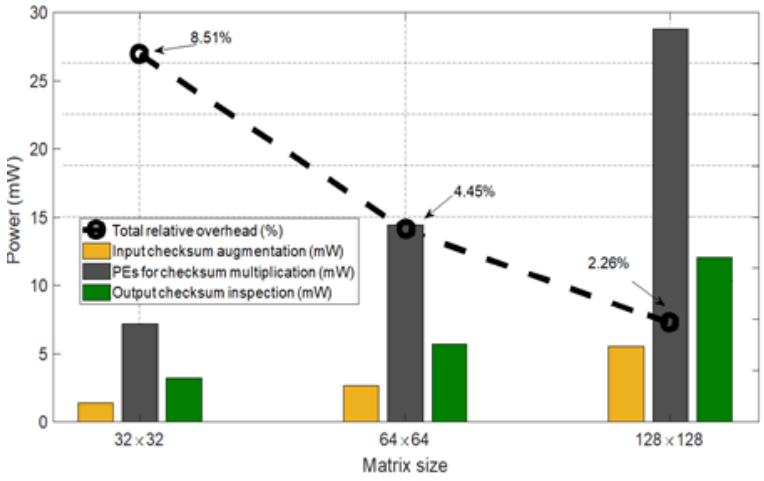

Fig. 34. Comparison of estimate power overheads with regard to design size (based on the SPICE simulations).

Zhang adapted Whatmough's work for a systolic array [124], and the datapaths of MAC units in Zhang's work are similar to those used in the systolic array proposed in this thesis. Both have 8-bit integer inputs with 32-bit integer accumulators. Adapting their work for the MAC units in the systolic array increases the transistor counts of PEs. With full RZFFs, a MAC unit may suffer from up to a 30% increase in power overheads, as RZFFs are clocked circuit components [122]. Ideally, majority of the flip-flops at the timing end-point must be augmented with a counterpart RZFF.

However, Zhang et al. allowed some headroom for errors. Based on their analysis, a fraction of the flip-flops (FF) are responsible for the majority of timing errors, hence, only the most critical timing paths were augmented with RZFF. They state that 14 FF out of 40 FFs were augmented, and their experiment showed 35% energy savings from the approximate computing scheme.

Based on Zhang's experiment with their approximate computing approach to the systolic array, the cost is 10.3% of power overheads in MAC units, while suffering a 1% drop in accuracy with neural models. However, the actual error rate was not reported.

A similar PE, modified for systolic designs by Gundi et al. [125], utilizing Razor based circuitry to detect errors, in co-simulations achieved a 2.5x improvement in energy efficiency by operating in the near-threshold region. However, this was gained at a cost of 20.8% of the logic overheads in the systolic array [125], similar to the original studies on adding RZFF to processors [97, 49]. This approach also suffers from a 2% average accuracy drop in neural inference, since RZFFs are added partially to MAC units, covering only some delay paths.

Since the RZFF are added to all PEs of the array, the overheads of Razor based methods grow linearly with array size, while the current proposed solution enjoys sub-linear scalability with systolic array size growth. For instance, in the case of a systolic array with the size of Google's TPU, i.e., 256×256, the overheads of the proposed solutions are negligible ($\approx 1.2\%$).

Furthermore, for reconfigurable logic devices such as FPGAs, RZFF are not utilizable due to hard blocks such as DSPs, while for the other logic components, RZFFs need to be inserted through "hacking" the vendor's tool [46]. The current proposed approach does not require any circuit level modifications. In fact, for FPGA implementation, off-the-shelf HLS tools can be used, with no need to modify the RTL code.

4.6 Summary and future work

An ABFT equipped systolic array detects computational errors on-the-fly with low overheads, supporting dynamic scaling of the operating voltage. The potential has been

explored through co-simulations with 90 nm CMOS technology [6]. Employing a highly parallel architecture as the systolic array can compensate for the performance loss from an aggressively reduced voltage operation.

Operating in the near-threshold region can be a very interesting application enabler, for instance, when embedded low-power neural network acceleration is needed. This is of interest for energy constrained "intelligent" IoT nodes and mobile computing applications.

There are applications for which even approximate results suffice. For example, when "bounded approximation" is required [126] the voltage could be scaled down to a degree at which the results are confined within certain error probability bounds.

5 Efficient signal acquisition in the Nyquist-Shannon sampling paradigm

5.1 Introduction

The share of signal acquisition from the total power dissipation can rise by up to 20% in energy constrained electronic systems, such as battery or energy scavenging powered IoT sensor nodes [127, 128]. As the interface to the analog world, Analog to Digital Converters (ADC) transform continuous-time physical signals into discrete-time numerical sequences through sampling and quantization, as shown in Fig. 35.

The power dissipated in ADCs is dependant on the ADC resolution and sampling rate [129], so curbing the resolution and sampling to the essential is a key means of energy conservation. The resolution is usually determined by the application, while the sampling rate is defined following the Nyquist criterion [130]. According to the Nyquist-Shannon sampling theorem, the input signal must be sampled at at least twice the rate of its highest frequency to enable perfect reconstruction (from non-quantized samples). Otherwise, the *Aliasing* phenomenon will be present. [131, 130].

The majority of current signal acquisition devices have been designed in accordance with the Nyquist-Shannon theorem. However, signal acquisition can also take place in a sub-Nyquist Compressive Sensing framework [11], where signal reconstruction relies on finding solutions to under-determined linear systems [132]. In this thesis, both approaches are considered with the focus on energy efficiency, starting in this chapter with Nyquist-Shannon rate sampling ADCs.

5.1.1 Analog to digital conversion in embedded systems

Energy dissipation in ADCs is a prominent design concern for devices that rely on tight energy budgets. To illustrate the state-of-the-art, the STM32 ARM Cortex-M0

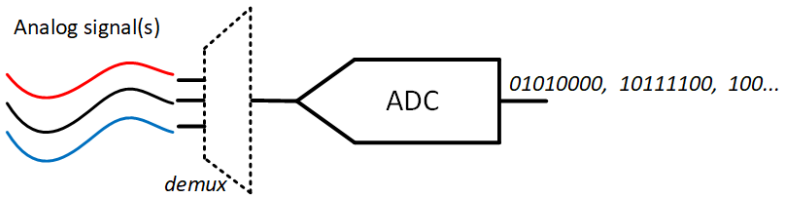

Fig. 35. Analog signals are converted into digital signals by the ADC.

microcontroller [133] draws 105 μW/MHz, while its embedded 12-bit ADC requires 60 μW at 10 kS/s and 240 μW at 1MS/s. A 16-bit ADC AD7980 from Analog Devices [134] dissipates 7 μW and 7 mW at 10 kS/s and 1 MS/s, respectively.

A simple 4-tap fixed-point FIR filter, in addition to a few comparisons of signal level against a pre-determined threshold, requires around 65 machine instructions on the STM32 ARM Cortex-M0. Therefore, for 10 kS/s data, the MCU must be clocked at around 65 kHz, translating into energy dissipation of 7 μW in the processor and 60 μW for the ADC. Moreover, at such a low operating frequency, the supply voltage of the processor can safely be reduced to save energy, while, due to the analog nature of ADCs, reduced voltage is not a readily available option [135].

Commodity ADCs are designed for a wide range of applications, so in many cases their design specifications exceed the requirements. While options such as auto-shutdown and burst-mode operation are occasionally provided [133], they do not represent architectural approaches that would enable reducing energy dissipation based on the characteristics of the input signal.

5.2 Successive approximation register ADCs

Justified by the energy efficiency and flexible sampling rate and the mixed-signal nature a Successive Approximation Register (SAR) ADCs are widely included in low power MCUs. SAR ADCs are especially useful in sensor readout interfacing, where the ADC needs to be time-multiplexed to capture input signals of different bandwidths at varying resolutions.

Modern off-the-shelf Integrated Circuits for SAR ADCs typically provide resolutions from 8 bits to 18 bits, with sampling rates up to several MHz [136, 137, 7].

The typical application ranges for commercial ADCs [138, 139] are depicted in Fig. 36. The horizontal axis represents the resolution, and the vertical axis the sampling rate. It is worth noting that the resolution decreases regardless of the ADC's architecture[9] [139] as the sampling rate increases. Each ADC architecture is best matched with a different set of applications. Typically, SAR ADCs are used for sampling rates from a few kHz to a few MHz [136].

As shown in Fig. 38, a typical SAR ADC consists of a comparator, a Digital to Analog Converter (DAC) and a digital SAR logic that carries out a conversion algorithm. One main advantage of the SAR approach is the extensive utilization of digital and mixed-signal components in its structure. Moving more functionality from analog to

[9] Primarily due to thermal noise, aperture uncertainty, and comparator ambiguity, as shown in [138]

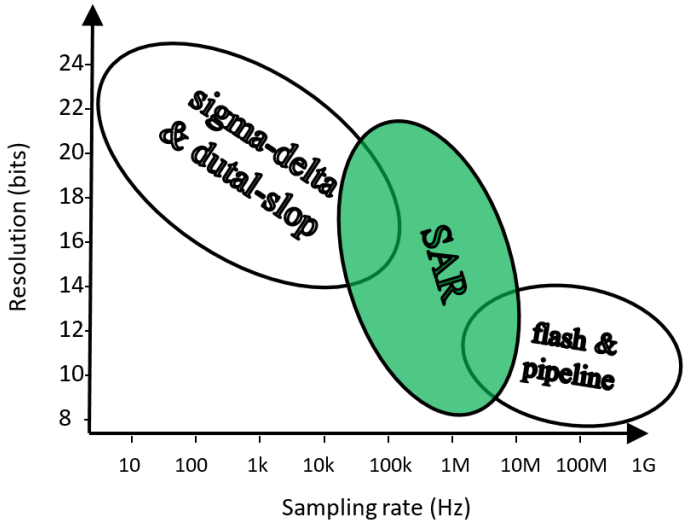

Fig. 36. Clustering ADC architectures according to resolution and sampling rate.

digital components is attractive, since the digital ones benefit more from transistor size and voltage scaling techniques.

5.2.1 Quantization by binary search

SAR ADCs essentially conduct a binary search to find the closest quantized approximation for the input analog value. The SAR logic sets the output of the DAC by changing its digital input to represent the most recent approximation of the final output from the ADC. The input of the DAC is adjusted successively until the output of the DAC is close enough to the input voltage, V_{in}. The process is illustrated in Fig. 37. The conversion is done cycle-by-cycle, starting from the Most Significant Bit (MSB) down to the Least Significant Bit (LSB). At the first “bit-cycle”, the SAR logic sets the outputs of the DAC to $\frac{V_{ref}}{2}$ by setting only the MSB bit to “1” and the rest to “0” (e.g., for a 3-bit ADC the code will be “100”).

Based on the comparison between the sampled input and the output from the DAC, the SAR logic keeps the MSB at “1”, or resets it to “0”. For the next bit-cycle, the next lower bit is set to "1", the DAC output is updated, and the process is repeated. In this manner, an N bit conversion is carried out in $N+1$ bit-cycles.

In Fig. 38 a capacitive DAC based SAR ADC is illustrated. While this scheme is common in SAR ADCs [140], several other types of DACs do exist [140].

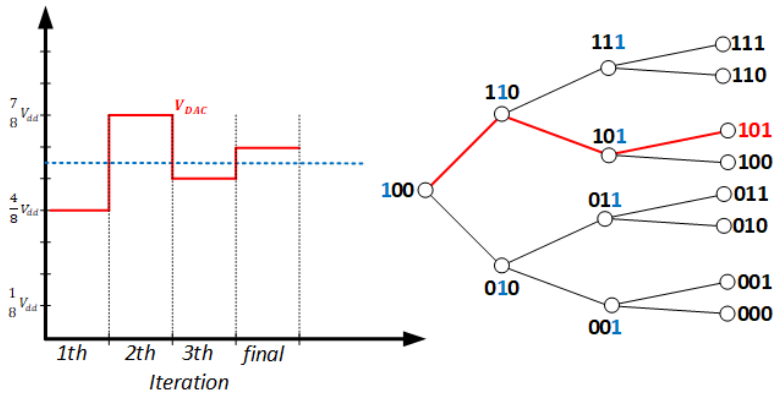

Fig. 37. Illustration of the operation of an example 3 bit SAR ADC.

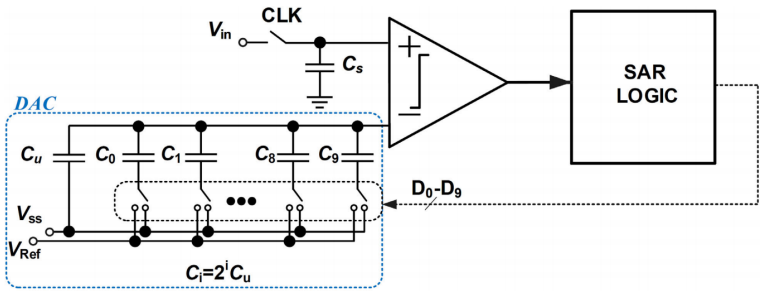

Fig. 38. General SAR architecture with capacitive DAC. Reprinted, with permission, from Paper [9] © 2020 IEEE.

5.3 Adaptive SAR ADCs

In many cases, the high end performance specifications of an ADC are only needed for a small set of applications [9]. In most other cases, a priori knowledge of the characteristics of the input signal can be leveraged into energy efficient designs that match the application's requirements.

In SAR ADCs, each bit-cycle carries an energy cost due to *i*) switching in the DAC, *ii*) digital logic, and *iii*) comparator activity. In particular, the first conversions incur the highest energy dissipation due to the larger capacitors for the MSBs in the DAC.

Generally, capacitive DACs do not consume static power; however , the SAR logic and the comparator do. Therefore, power dissipation can be reduced by adopting conversion algorithms that minimize the switching activity of the DAC, and perhaps intermittently turn off the comparator and the SAR logic. Such schemes suit applications

that deal with low-activity, or periodically low-activity input signals. In the following, we review such strategies and present a novel scheme.

5.3.1 Special SAR ADC approaches

In low activity digital signals, the majority of variations take place within the least significant bits. Conversion strategies that resolve only the LSBs reduce the number of bit-cycles, and the energy dissipation at the DAC, as the capacitors for the MSBs are the largest ones. Several signal-specific SAR ADCs proposed for low-activity signals are based on this observation.

Yaul et al. [141] have proposed an "LSB-first" SAR ADC scheme. Their SAR algorithm keeps the output from the previous sample, and starts resolving from the LSB toward the MSB. When the rate of change is very small, e.g., the input is a DC signal, an N bit LSB-first ADC needs only a single bit-cycle. The energy efficiency gains of the approach suffer and the conversion cycles grow when the signal activity increases. An abrupt change might incur up to $(2 \times N + 1)$ bit-cycles.

Yim et al. [142] propose a 10 bit SAR ADC that only resolves 5 LSBs. Full-range conversion is done in the case of an incorrect prediction, i.e. when a sample to sample variation exceeds 5 bits. A similar approach by Chen et al. proposes a more adaptive algorithmic SAR ADC that retains the electrical charge in the K MSB capacitors, and attempts to approximate the signal by modifying the $N-K$ LSB bits [143]. The number of LSBs to be resolved, $N-K$, changes according to the sample to sample variations. For low-activity signals, the ADC requires only $(K+1)+2$ bit cycles per conversion, while for high-activity signals it requires $(N+1)+2$ bit-cycles [144].

In some applications, the low-activity periods of the signal are not of interest. For such uses, Nasserian et al. [145] propose a lossy ADC scheme in which the resolution of the ADC alternates between 5 and 10 bits, according to the signal activity. In a resembling scheme, instead of resolution, Chen et al. [146] propose to adaptively adjust the sampling rate according to the signal activity. Generally, such trade-offs between precision and power efficiency are not tolerated by all applications.

5.4 Novel arithmetic tracking SAR ADC

Most tracking ADCs, including the ones introduced in section 5.3.1, employ modified SAR techniques, relying on the assumption of sample to sample adjacency in binary representation, i.e., they have a small Hamming distance.

However, the Euclidian distance of two consecutive samples might actually be small, while the binary Hamming distance is large. For example, in the case of values "1100 0111" and "1100 1000" the Hamming distance is four LSBs, while the Euclidean distance is only 1 LSB. Many tracking SAR ADCs do not verify whether the initial tracking guess was correct or if the overheads of verification are non-trivial, e.g., the scheme proposed in [143] consumes 2 additional cycles for verification. Every extra cycle dissipates energy.

To allow for more efficient signal tracking, an *arithmetic* successive approximation approach was adopted in this thesis. It verifies the correctness of conversion, without incurring any extra bit-cycles.

5.4.1 The arithmetic tracking SAR principle

The conceptual organization of the proposed SAR ADC is presented in Fig. 39. The delay block keeps the output code from the previous cycle. The comparator determines if the shift register value needs to be added to or subtracted from the value held by the delay block. The initial value for the shift register is determined by finding the binary difference of the two consecutive conversions, and rounding up to the next highest power of two value.

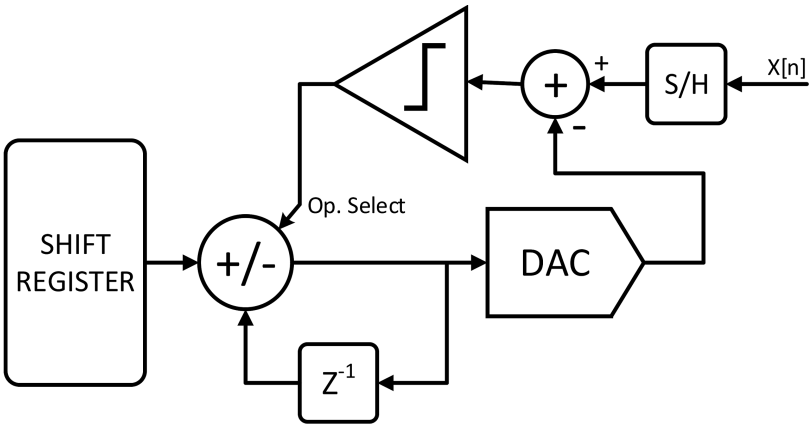

Fig. 39. A block-diagram representation of the arithmetic tracking ADC. Reprinted, with permission, from Paper [9] © 2020 IEEE.

The binary output code is resolved by arithmetic successive approximations as shown in the in Fig. 40. The digital output code is represented by D. At the first conversion cycle, D is updated with the resolved digital code of the previous sample.

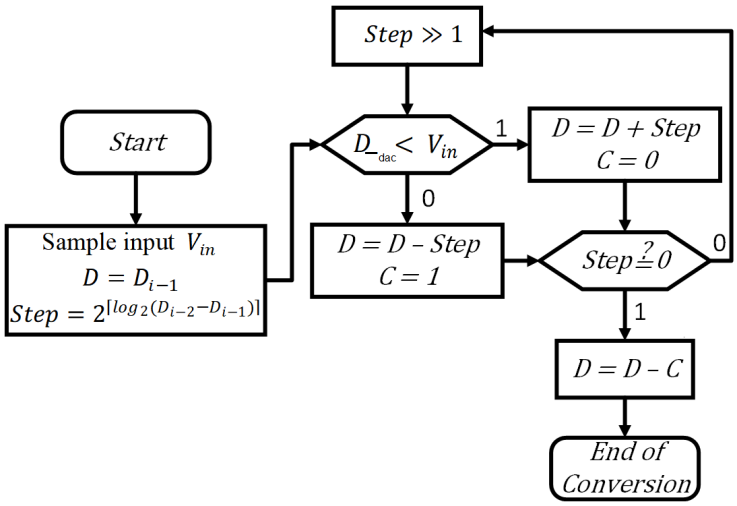

Fig. 40. The algorithm for the proposed arithmetic tracking SAR ADC. Reprinted, with permission, from Paper [9] © 2020 IEEE.

The analog equivalent of the digital code at each cycle, i.e. the output of the DAC in the SAR circuit, is compared to the input. The result from the comparator defines whether a step value is added to or subtracted from D. Notice, for the first conversion, D, is randomly initialized. The step value is shifted to the right, and the process continues until all bits in the shift register are shifted out. The initial value on the shift register is determined by the first-order prediction logic based on the rate of the signal change. The longer the binary initial value is, the more bit-cycles will be carried out.

After completion of each conversion, the arithmetic tracking ADC determines whether the input sample was within the predicted range. For this purpose, the comparator output at consecutive bit-cycles is used as a bit sequence. When the sequence for a conversion is all '1's, e.g. "11111", the input value has exceeded the predicted range, and the conversion is erroneous. Similarly, when all elements of the bit sequence is all '0's, such as. "00000", the input value has under-shot the predicted range. The state-machine shown is in Fig. 41.

Figure 42 illustrates the cycle-by-cycle output value of the DAC for the first few conversion cycles of an arithmetic SAR. The blue circles represent raw samples, while the red plot represents the DAC voltage equivalent to the resolved digital code.

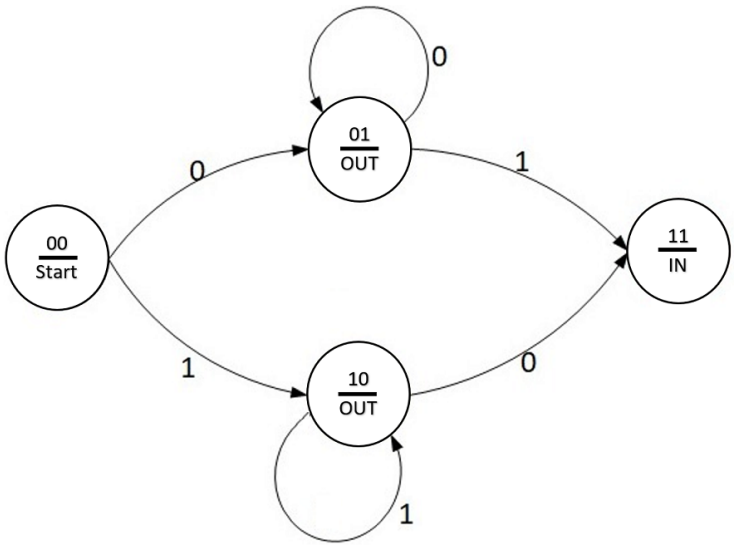

Fig. 41. The state-diagram of the out-of-range detection circuit. Reprinted, with permission, from Paper [9] © 2020 IEEE.

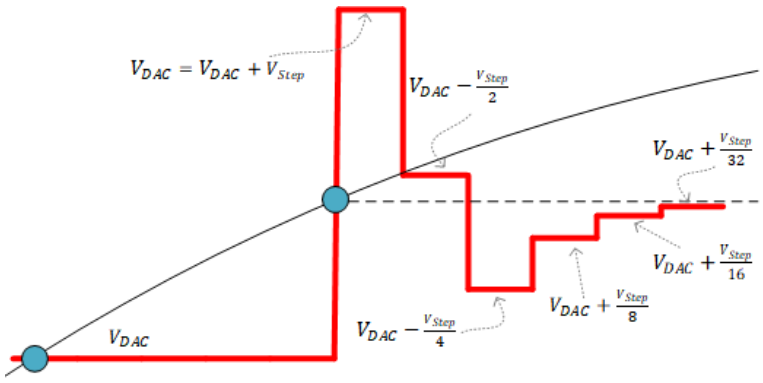

Fig. 42. Illustration of operation of the arithmetic SAR ADC. Reprinted, with permission, from Paper [9] © 2020 IEEE.

5.5 Simulations and results

A behavioral model of the proposed solution was implemented in MATLAB, and the circuit-level model simulations were carried out using HSPICE.

The average bit-cycle counts per sample in selected application cases were estimated as a proxy for energy dissipation. Important biomedical and industrial signals that are known to have periodical low activity regions were used.

Table 5 presents the average number of conversion bit-cycles and the misprediction rates for the test cases: ECG, EEG, pulse train, and image signals. The small number of bit-cycles per sample enables us to utilize power gating and clock gating techniques to further save energy.

Table 5. The average number of bit cycles and misprediction rates of the proposed arithmetic tracking SAR, and the conventional SAR approaches for different signals. Reprinted, with permission, from Paper [9] © 2020 IEEE.

Parameter	SAR type	ECG	EEG	Image	15 kHz sine	10 kHz pulse
Mean bit cycles	Prop.	4.0	7.7	3.3	4.4	3.3
	Conv.	10	10	10	10	10
Miss rate (%)	Prop.	1	12	0.9	1.5	1
	Conv.	0	0	0	0	0
Sampling rate (Hz)		1024	256	N/A	1M	100k

The number of bit-cycles per sample for the ECG signal is depicted in Fig. 43. Due to the tracking principle, the cycle count increases when the signal changes rapidly. The predictions and the initial step sizes of the conversions are illustrated in Fig. 44 for a section of the same signal.

The bit-cycle results from the simulations using the behavioral model were compared to two other tracking SAR ADC solutions, the one proposed by Chen et al. [143], and the LSB-first scheme proposed by Yaul et al. [141]. The results shown in Fig. 45 demonstrate that the proposed arithmetic tracking method on average reaches the lowest bit-cycle count. However, the energy dissipation depends on the characteristics of the logic and analog designs, and needs to be explored through circuit simulations or physical implementations.

5.5.1 Circuit-level simulations

A 90 nm CMOS HSPICE model for the arithmetic tracking SAR ADC was created, and the functionality of the ADC was verified for low-changing signals. The comparator was

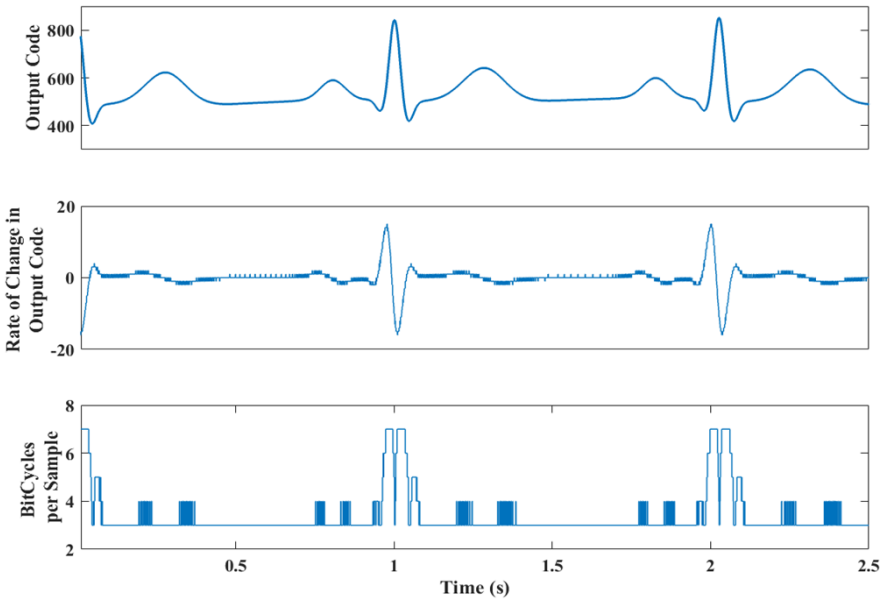

Fig. 43. The number of bit-cycles consumed for a sample ECG signal. Reprinted, with permission, from Paper [9] © 2020 IEEE.

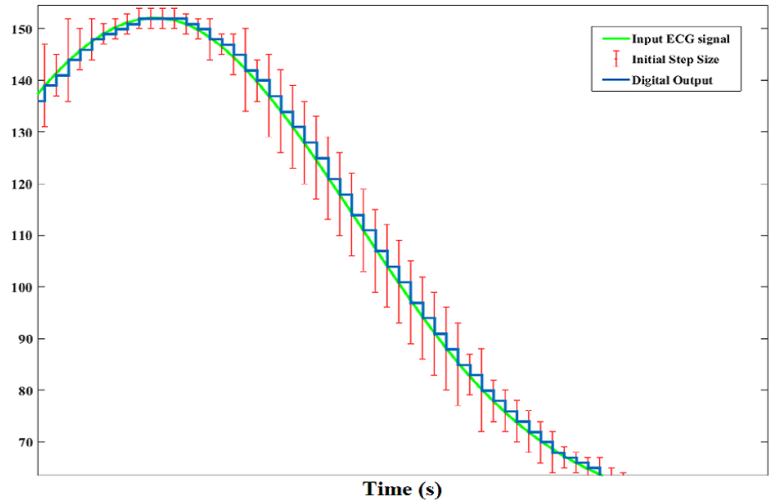

Fig. 44. The tracking window adjusts according to the sample to sample variation.

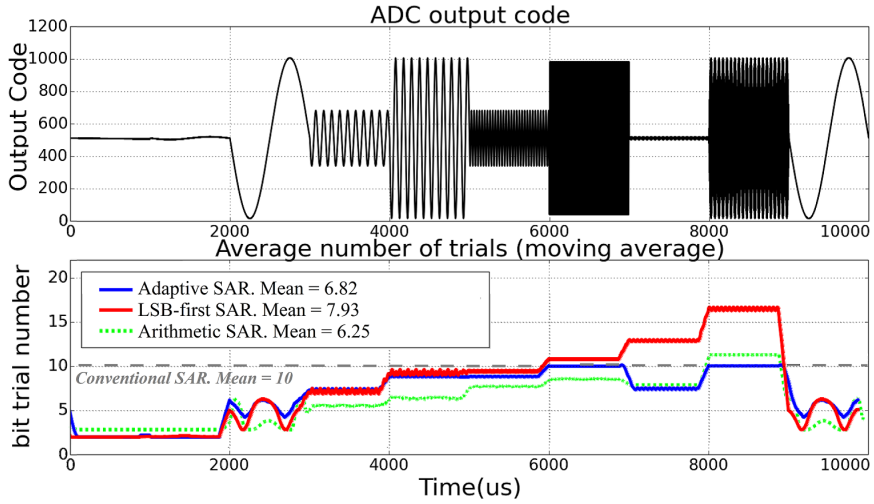

Fig. 45. Comparison of bit-cycles required by different adaptive SAR algorithms, i.e. the proposed arithmetic SAR, "Adaptive SAR" [147, 143], and "LSB-first" [141, 148]. Reprinted, with permission, from Paper [9] © 2020 IEEE, originally adapted from [143].

power-gated, while clock-gating was used in the digital section after each successful conversion. The power dissipation was 28 μW at a 800 kHz/s sampling rate. Most of the power dissipation was in the digital section, as shown in Fig. 46. This is because the number of cycles and voltage variations across the DAC are reduced, while the digital side is more complex than with the regular SAR approach. However, since the delay paths are relatively short, and the operating frequency rate is moderate, a near-threshold voltage scaling scheme could be adopted to maximize the energy efficiency of the digital logic section.

5.5.2 Oversampling use case

When the resolution of an ADC embedded in an MCU is insufficient for an application, oversampling and decimation are used to increase the Effective-Number-of-Bits. Each quadrupling of the sampling frequency adds one effective bit to the resolution, while increasing frequency is at the cost of increased power dissipation.

The arithmetic tracking SAR concept was modified for an oversampling case where the maximum sample to sample variation can be determined, based on the signal bandwidth and oversampling ratio. As with low-activity signals, in oversampling mode

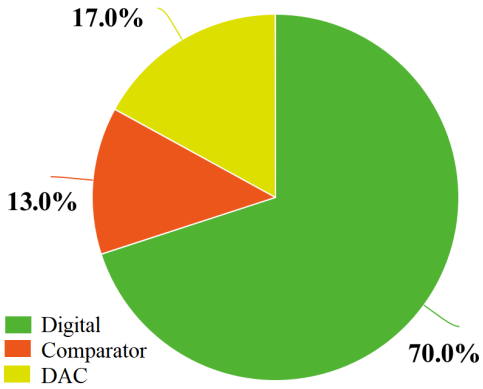

Fig. 46. Breakdown of power consumption of the proposed SAR ADC. Reprinted, with permission, from Paper [9] © 2020 IEEE.

Table 6. Estimated power consumption with respect to the oversampling ratio. Reprinted, with permission, from Paper [7] © 2019 IEEE.

Mode	OSR	Variation (LSBs)	Cycles	pW/S
Regular SAR	-	-	9	12.8
Tracking SAR	32	25	7	8.03
Tracking SAR	64	12.5	5	6.33
Tracking SAR	246	3.1	4	5.53

the sample-to-sample variations are small. A solution that leverages this observation into reduced bit-cycle conversions was implemented.

Furthermore, since the maximum input bandwidth, the largest signal amplitude, and the oversampling ratio are known, the digital circuitry acting as the predictor and out-of-range-detector was removed. Instead, the tracking ADC was initialized with a pre-calculated step that was fixed for each new conversion [7].

The energy dissipation per sample was estimated through behavioral simulations. By adapting the number of cycles to the sample-to-sample variation, the energy efficiency of each conversion improved as shown in Table 6. With larger oversampling ratios, less bit-cycles are required for each conversion, resulting in less energy dissipation per sample compared to, e.g., conventional oversampling. The resolution of the ADC was 8-bits, and the input signal was a full-scale sinusoid.

5.6 Summary

Prior knowledge about signal characteristics provides opportunities to improve the energy efficiency of analog-to-digital conversions through matching architectural approaches. The fundamental SAR ADC principle provides opportunities for algorithmic innovations that enable leveraging information about signal characteristics into reduced power dissipation.

Signals with occasional high activity periods are an attractive target for tracking SAR ADCs. The proposed arithmetic tracking ADC adaptively adjusts its operation to signal activity, exploiting low activity periods to lower the overall energy dissipation. The approach was shown to be useful for oversampled signal acquisition as well.

6 Sub-Nyquist-Shannon rate sampled signal reconstruction

In the previous chapter, an arithmetic tracking SAR ADC was proposed for exploiting signal characteristics in time and spatial domains. In battery and self-powered wireless devices, energy is then dissipated by source coding before the data is transmitted. However, when represented in an appropriate basis, many real-world signals are sparse or compressible, that is, in some domain much of their coefficients are close to or equal to zero. This provides for an alternative approach to improving the efficiency of the signal acquisition process.

Compressive sensing (CS) [149] is a signal acquisition and low complexity compression framework that is one of the means to finding energy trade-offs between sensing, source coding, and transmitting the measurements. Essentially, it enables the acquisition and reconstruction of sparse signals sampled at lower rates than the Nyquist-Shannon theorem demands.

In this chapter, an evaluation of CS theory based signal reconstruction algorithms in embedded settings is presented. Novel solutions to mitigate the identified challenges are proposed, analysed, and simulated.

6.1 Compressive sensing theory

According to the Nyquist-Shannon sampling theory, perfect reconstruction requires a sampling frequency (f_s) of at least twice the maximum bandwidth of the signal (f_{max}) [150, 151]. Instead, according to CS theory, to enable reconstruction, the number of signal samples (or "measurements") can be proportional to the information content [151]. Information content in this context refers to the share of non-zero elements or the sparsity degree of a given signal [152].

6.1.1 Signal sparsity

A signal $X \in R^N$ is defined k-sparse if there are only k non-zero coefficients in X, where $k << N$. Equivalently, the signal may have a sparse representation in a base such as Ψ where $\alpha = \Psi x$, when there are only k non-zero coefficients in α. As an example, consider Fig. 47, where a sinusoid is plotted in time and in Fourier domains. While the sinusoid is not a sparse in the time domain, in the the Fourier domain it is very sparse.

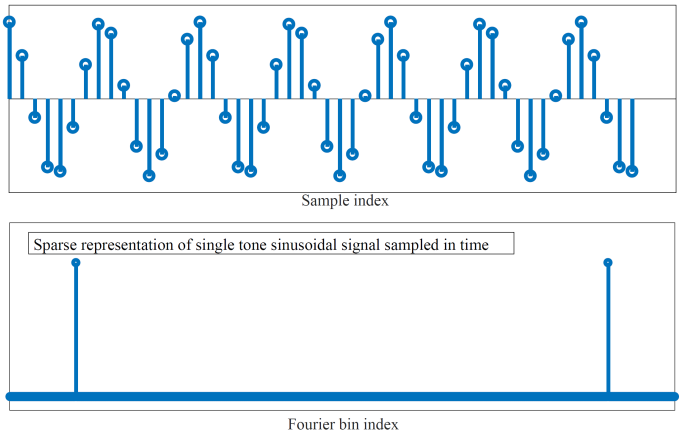

Fig. 47. Sparse representation,F , of a single-tone sinusoidal signal,X, in the Fourier domain.

Signal sparsity is widely exploited in many signal processing applications, such as image/video compression [153, 154, 152].

6.1.2 Compressive sensing framework

Based on the CS theory, exact reconstruction of a length-N signal $X \in R^N$, with a sparsity degree of k, requires only M *measurements* ($M << N$). In the CS context, the measurements refer to the linear combination of elements of the input signal, obtained through multiplying the signal vector X with the measurement matrix Φ,

$$Y_{M\times 1} = \Phi_{M\times N} \times X_{N\times 1}. \tag{13}$$

where, $Y \in R^M$ is the measurements vector (Fig. 48).

As Equation 13 indicates, the CS framework can be utilized for increasing the energy efficiency of signal acquisition by reducing the number of samples [153]. This spares data compression effort, and saves power from transmission in energy constrained sensor nodes, at the cost of signal reconstruction at the receiving end [151].

6.1.3 Sparse signal reconstruction

Based on the CS framework, the original sparse signal X can be reconstructed from the measurement vectors. At the reconstruction stage, vector X in the the system, Equation 13, contains the unknowns, Φ the coefficients of the system, and the measurement vector Y the constant terms. Since the number of measurements, equivalently

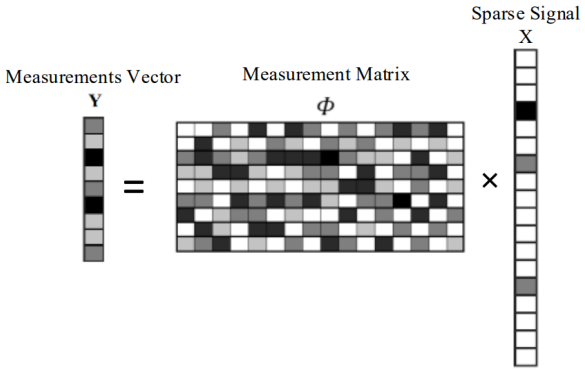

Fig. 48. CS measurement (Equation 13) of a sparse signal. Reprinted, with permission, from Paper [10] © 2018 IEEE.

the number of equations, is less than the number of signal elements or unknowns, ($M << N$), Equation 13 is under-determined [152].

An under-determined system of equations has infinitely many solutions. A closed form solution, i.e. $x^{'} = (\Phi^T \Phi)^{-1} \Phi^T y$, may return an infinite set. However, knowing that X it is sparse means that a large share of the unknowns are zero, helping to identify the desired solution [151]. The reconstruction method to leverage sparsity is the key in the CS framework.

A naive approach to solving Equation 13 is solving it multiple times, on each occasion assuming a different set of elements of X to be none-zero [153, 152]. At first,X the sparsest possible signal is assumed, i.e., all coefficients except one in vector X are assumed to be zero [10]. Then, all possible combinations of Equation 13 are considered and solved. If no solution is found, the assumed number of non-zero coefficients is increased to two, and all possible forms of Equation 13 for two non-zero elements in X are solved again. That translates into solving Equation 13 $\binom{N}{2}$ times [152].

The process continues with repeatedly solving for all possible subsets , until a fitting solution is found for a sparse signal with k non-zero elements. The worse case number of trails is shown in Equation **??**. The needed number of trials quickly render this approach computationally prohibitive [152].

The original signal X does not need to be sparse in the sampling domain allowing for reconstruction. It suffices to possess a sparse representation α in a transform domain, i.e. $X = \Psi \times \alpha$ [152]. With a small modification to Equation **??**, instead of solving X, we can solve α from Equation 14,

[10] In mathematical terms, this means minimizing for the L_0 pseudo-norm

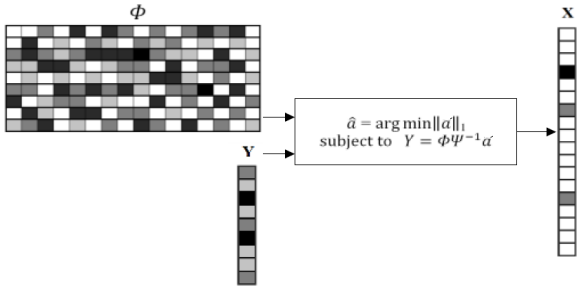

Fig. 49. CS reconstruction (Equation 14) of a sparse signal from the measurements vector. Reprinted, with permission, from Paper [10] © 2018 IEEE.

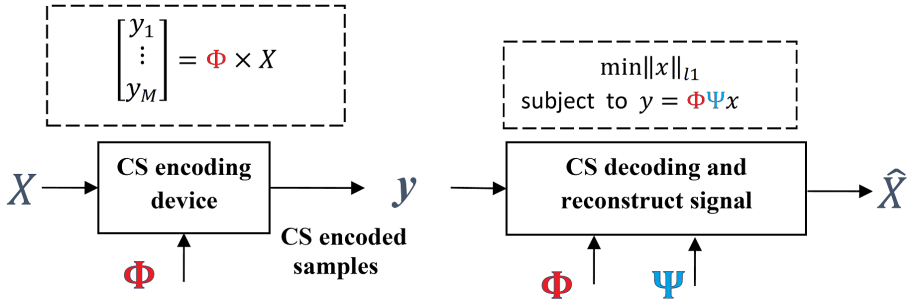

Fig. 50. Signal sensing and reconstruction in the CS framework. Redrawn from [155].

$$\hat{\alpha} = argmin(||\alpha'||_1) \;\; subject\ to \;\; Y = \Phi \times \Psi^{-1} \times \alpha'. \tag{14}$$

The respective reconstruction is depicted in Fig. 49. It is necessary in reconstruction that the columns of the measurement matrix bear the least similarity to each other, i.e. they are incoherent [151, 152].

Using random matrices is the most straightforward way to achieve incoherent measurements. Hence, in many CS applications a pseudo-random number generator is employed [10]. Fig. 50 depicts an end-to-end CS based sparse signal acquisition and reconstruction process.

6.2 CS based signal acquisition systems

For the CS framework, hardware implementations of Analog to Information Converters (AICs) have been proposed as the alternative to conventional ADCs [156, 151]. AICs

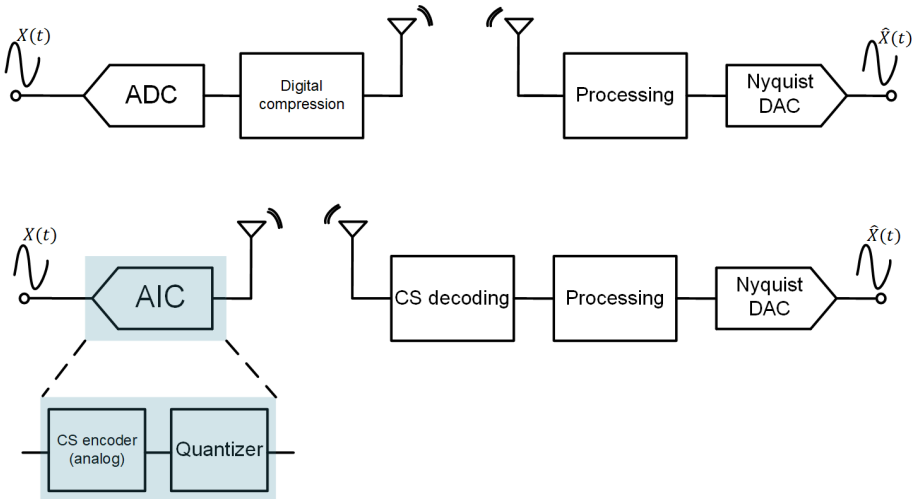

Fig. 51. Structure of an AIC and the comparison with ADC based signal acquisition and processing.

improve the overall energy efficiency [155] by taking fewer quantized samples for sparse signals.

As depicted in Fig. 51, the AIC requires a CS encoding front-end (Equation 13) and a quantizer, while reconstruction (Equation 14) is done at a decoding back-end that is not energy constrained, as shown in Fig. 51 [157].

Promising results for AICs have been reported in literature [156, 155, 11]. However, while power is saved in sampling, the total energy dissipation could increase due to the high computational cost of signal reconstruction [155] [152].

6.2.1 Non-uniform sampler

For the hardware realizations of AICs, various architectures have been proposed [156, 158, 155]. A non-uniform sampler (NUS) can be considered as the simplest AIC scheme, and has been used in the current work.

The NUS structure is shown in Fig. 52. It is basically a conventional ADC with Nyquist-Shannon rate sampling. However, only randomly chosen sample times are used, hence reducing energy dissipation. Such selection of only half of the samples results in an average sampling rate below the Nyquist-Shannon limit.

A pseudo-random bit generator is used to trigger the ADC to acquire a sample. Successive Approximation-Register (SAR) ADC was employed due to its simplicity and capability to sample multiplexed signals.

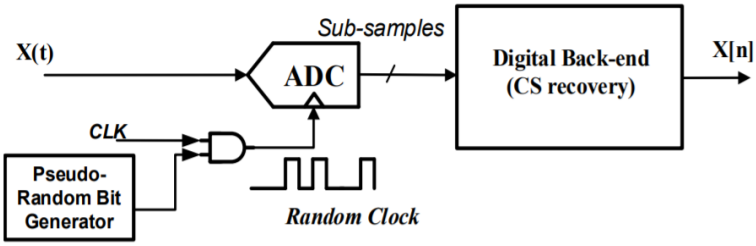

Fig. 52. Structure of a non-uniform sampler. Reprinted, with permission, from Paper [11] © 2018 IEEE.

The sampling scheme of the NUS implements the measurement matrix Φ. While the NUS does not allow reconstructing time domain sparse signals, it can be utilized for signals sparse in the frequency domain.

6.3 CS reconstruction algorithms

In the current investigation, three computationally tractable and robust CS reconstruction algorithms were used [159]: Orthogonal Matching Pursuits (OMP) [160], Approximate Matching Passing (AMP) [161], and Normalized-Iterative Hard Thresholding (N-IHT) [162, 163]. These algorithms represent computationally attractive greedy CS reconstruction approaches. The developed implementations are available from an online repository [11].

OMP is an extension of the Matching Pursuit algorithm [160]. Through finding most correlated columns of the coefficient matrix $\Phi \times \Psi^{-1}$ with measurement vectors Y, the OMP algorithm iteratively identifies non-zero elements in the sparse signal X. At each iteration one non-zero element is identified. The identified non-zero elements form a subspace of unknowns, and the rest of the unknowns are assumed to be zero, and are removed from the problem. In each iteration, the problem, now reduced to a small over-determined system of equations, is solved using Least Squares method. The iterations continue until a stopping criteria holds.

The AMP and IHT algorithms are similar, but unlike OMP do not rely on Least Squares solutions. Instead, the signal is reconstructed by vector truncation that completes the sparse signal in an iterative manner.

The performance of the AMP algorithm depends greatly on the selected parameters. In our work, those recommended in [161] were used. Empirical studies on the original

[11] `https://github.com/NeuroFan/Compressive_Sensing`

IHT algorithm indicated poor performance [163], so a different version, normalized-IHT was implemented in which the parameters[12] are updated dynamically in each iteration.

6.3.1 Evaluation of CS reconstruction algorithms

The selected CS reconstruction algorithms were evaluated with the focus on energy efficiency using three processing platforms: ARM Cortex15, NIOS II, and a Transport Triggered Architecture (TTA) based application specific instruction set processor (ASIP). The NIOS II and the TTA processors were implemented on a Cyclone IV-EP4CE115F29C7 FPGA. The algorithms were coded in C language, and optimized to maximise parallelism on each platform.

The algorithms share several common macro operations, i.e. matrix arithmetic, matrix transpose, sorting, thresholding and norm calculations. The processor simulator used[13] identified matrix multiplication and matrix inversion (used in the LS stage of the OMP algorithm) as the most resource demanding macro operation.

The results from the implementations for the TTA processor are shown in Fig. 53. For the other processors, the reconstruction time vs. Signal to Noise Ration (SNR) trends are similar. The performance of NIOS II turned out to be about a tenth of the TTA processor, while ARM beat the latter by being ≈ 30 times faster. The differences are due to the internal architectures, implementation technologies, and clock frequencies of the processors. The power consumption results are presented in Table 7.

In terms of energy per sample, the customised TTA processor implemented on Cycle IV FPGA performed less well than the ARM processor. This is attributed to underlying old process technology and high static power consumption of FPGAs. Much higher energy efficiency has been demonstrated in [165] by Hautala et al. for a somewhat similar TTA processor, with roughly the same number of gates implemented using a 28 nm process.

6.4 Challenges with the CS framework

While the CS signal acquisition framework is theoretically attractive, and appears promising, practical challenges limit its usability [166]. The substantial demand of resources for signal reconstruction is only one of the issues curbing the adoption of the CS framework in applications [167], while others include the impacts of quantization noise and spectral leakage.

[12] e.g. step size
[13] PROXIM from TCE toolset [164]

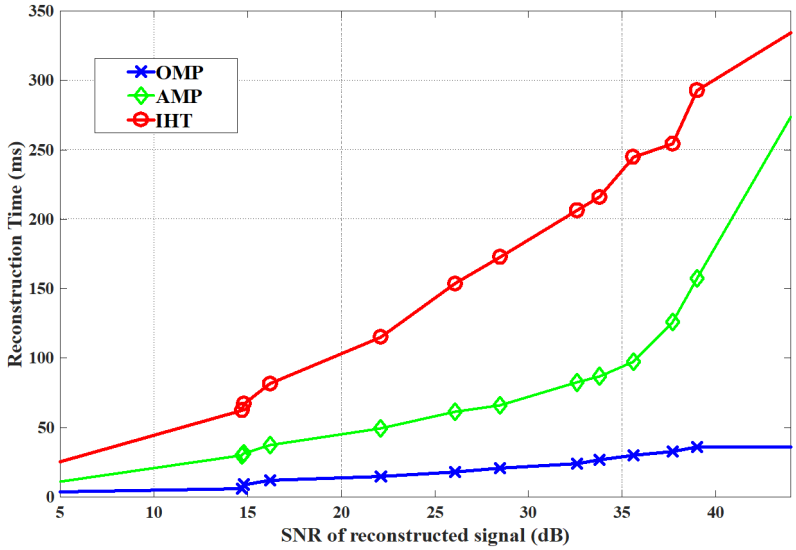

Fig. 53. Performance of the CS algorithms on an embedded platform. Reprinted, with permission, from Paper [10] © 2018 IEEE.

Table 7. Energy consumption report for running the CS algorithms. Reprinted, with permission, from Paper [10] © 2018 IEEE.

Processor	NIOS II	Customized TTA	ARM Cortex 15
Dynamic power (mW)	195	174	1700
Clock frequency (MHz)	50	50	2000
Energy/Sample (μJ/S)	390	42	15
Implementation Technology	60 nm FPGA	60 nm FPGA	28 nm ASIC

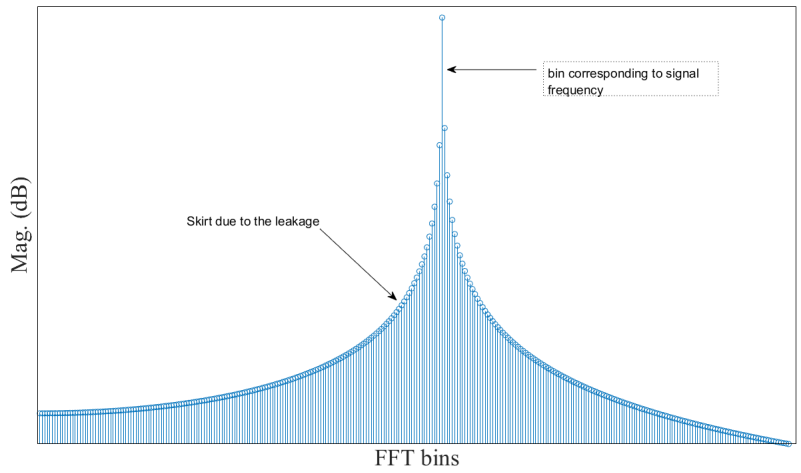

Fig. 54. A sinusoid's Fourier representation is no longer sparse due to spectral leakage.

6.4.1 Noise folding

Let us consider CS based sampling with additive noise n present in the input, so $Y_N = \Phi_{M\times N} \times (X_N + n_N)$. In CS measurements, the input noise is projected from an N-dimensional space into a lower M-dimensional space. This means that the elements of noise are multiplied by matrix Φ, added together, and stored in the elements of Y. It has been shown that noise leaked into the measurement is *amplified*, and projected back to the reconstructed vector $\widehat{X}$ [166].

Unfortunately, it is rarely the case that practical signals are be perfectly sparse and noiseless. Quantization noise is unavoidably added to the input in the CS signal acquisition [168, 167].

6.4.2 Discrete bases and leakage problem

CS assumes signal sparsity in discrete bases, such as Fourier and wavelets. etc. Due to windowing of finite signal segments [158], even a signal with very few components might not have a sparse representation at all. An extreme example of such spectral leakage [166] occurs with rectangular windowing: the DFT of a single tone sinusoid can occupy most of the frequency bins, as shown in Fig. 54.

Duarte and Baraniuk [169] have proposed an adaptive adjustable basis for CS reconstruction to mitigate the spectral leakage problem. Unfortunately, this substantially increases the already high computational complexity [166].

6.4.3 Computational complexity of reconstruction

While the energy efficiency of AICs appears promising in comparison with using ADCs and source coding, the computational complexity of reconstruction in the back-end can be a substantial challenge [170]. Even highly optimized implementations of reconstruction algorithms require much more energy per sample than a counterpart conventional ADC. For instance, [171] reports $14nJ$ per reconstructed sample, while ADCs often demand a few pJs per sample [9, 129]. Such 2-3 order of magnitude differences can outweigh energy savings from transmitting and receiving less data.

Even highly optimized reconstruction designs, such as [172], [171] and [173] suffer from high computational costs. In the following, a solution to mitigate the computational complexity of CS reconstruction is proposed. While not a total solution, it is a step toward improving the energy efficiency.

6.5 Reducing the computational complexity of reconstruction

One way to reduce the computational complexity of CS reconstruction algorithms is to reduce the number of iterations required. This translates into somehow simplifying the CS reconstruction part. The approach adopted in this thesis compromises by adding small overheads to the CS signal acquisition and data transmission.

The proposed scheme modifies the NUS AIC structure to provide the CS reconstruction with more data samples. The extra samples have lower bit resolution, and help to identify the probable location of non-zero elements in the sparse signal X using less iterations and energy. Transistor level simulations demonstrate that the added energy overhead in signal acquisition is negligible.

6.5.1 Low resolution Nyquist-Shannon sampling aided CS

CS reconstruction algorithms aim at determining the locations of the non-zero elements in the unknown sparse signal. If they were known beforehand, e.g., the least squares method of the OMP algorithm, for example, would reconstruct the signal with a single iteration.

In the case of Fourier domain sparse signals, the position of the non-zero coefficients in the spectrum can be estimated using uniformly sampled low bit resolution measurements. The principle of the proposed signal acquisition solution is depicted in Fig. 55.

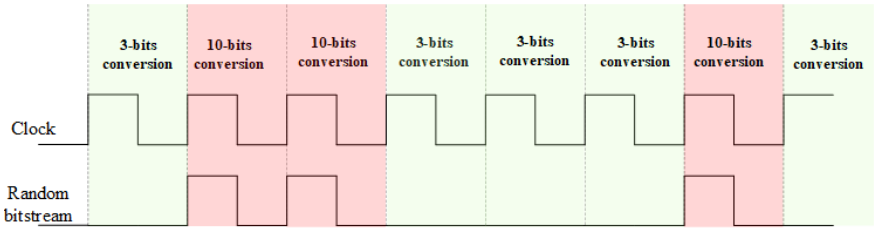

Fig. 55. Illustration of dual-mode operation of the SAR ADC.

The high resolution sub-Nyquist-Shannon rate AIC samples the sparse signal to the measurement vector Y through the measurement matrix Φ. The low-resolution Nyquist-Shannon rate ADC samples are fed to a computationally light DFT based spectral estimation [174] that detects most of the zero coefficients of the signal. When they are removed, the CS reconstruction problem boils down to an over-determined system of equations. This new reconstruction problem can efficiently be solved, e.g., using the least-squares method. Subsequently, the removed zero coefficients are inserted into their corresponding locations, and a full-length reconstructed input is obtained.

The computational complexity is now bounded to $O(NlogN)$ by the DFT algorithm. This is trivial compared to the complexity $O(NM)$ of the OMP reconstruction algorithm.

6.5.2 Proposed designs

An efficient realization of the signal acquisition concept is presented in Fig. 56. A minor modification was made to the SAR ADC to support dual mode operations, enabling sampling the input signal at two different resolutions, 3-bits and 10-bits.

Figure 57 details the organization of the dual-mode SAR ADC. Components are shared between the low-resolution ADC and the high-resolution AIC, saving both in energy dissipation and chip area. The SAR ADC is equipped with a pseudo-random clock generator based on Linear-Feedback-Shift-Register (LFSR) with adjustable distribution of 1s and 0s, enabling us to set the ratio between the high and low resolution samples. The ADC operates as a regular low-resolution SAR ADC, unless it receives a "1" from the LFSR, changing to high-resolution mode. High resolution samples are stored as CS measurement, Y, while all samples are stored for spectral estimation.

6.6 Simulations and results

Initially, the proposed system was behaviorally simulated using MATLAB. The computational complexity of the reconstruction method was measured by counting the

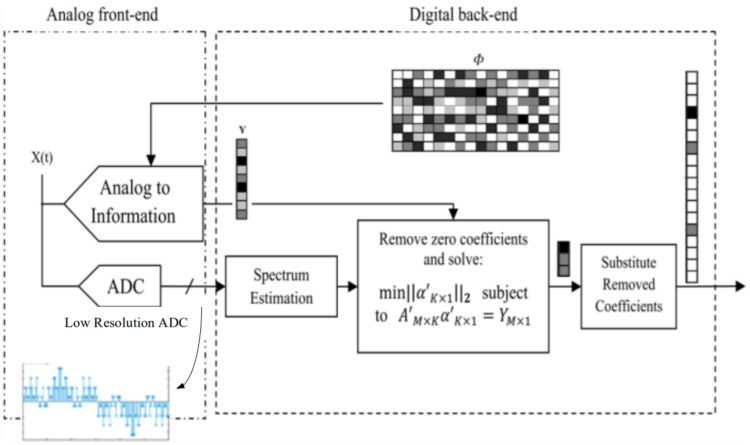

Fig. 2. Architecture of proposed acquisition system

Fig. 56. The proposed structure to combine an AIC and a low-resolution ADC. Reprinted, with permission, from Paper [11] © 2018 IEEE.

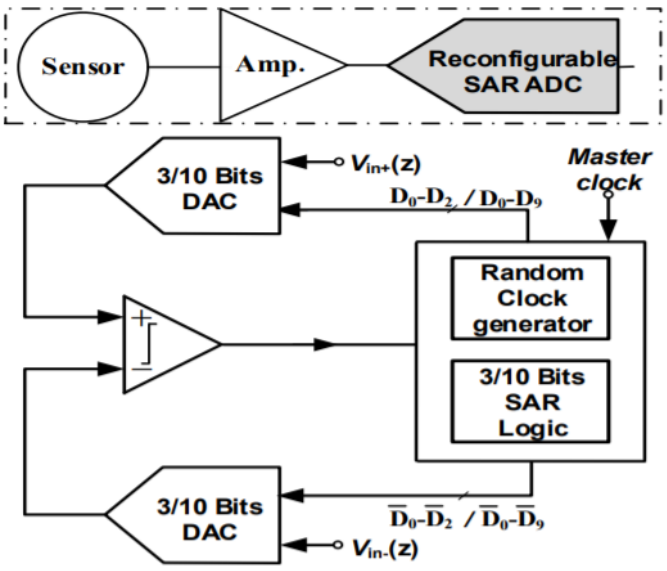

Fig. 57. Dual-Mode SAR ADC acting as a 10-bit NUS and a 3-bit ADC intermittently. Reprinted, with permission, from Paper [11] © 2018 IEEE.

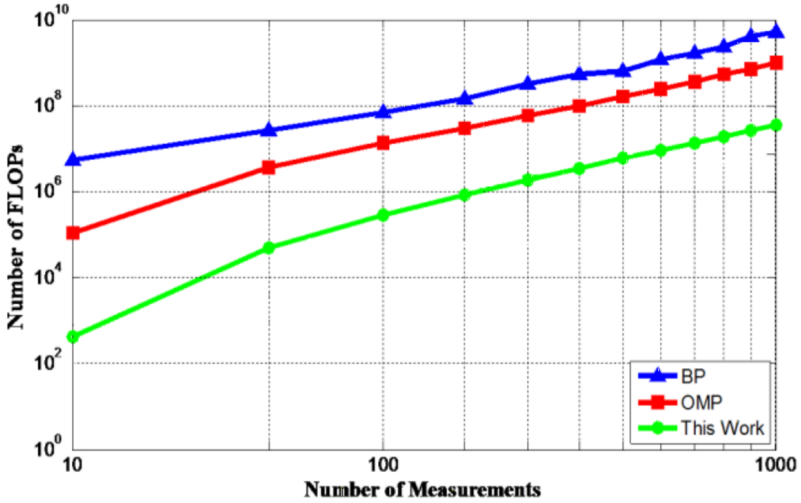

Fig. 58. FLOP counts for different reconstruction approaches. Reprinted, with permission, from Paper [11] © 2018 IEEE.

number of Floating-Point-Computations (FLOP), and comparing it against other CS reconstruction algorithms.Then, transistor level simulations of the SAR ADC-NUS were carried out.

In behavioral simulation, a sparse input signal of length $N = 1024$ for a different number of measurements M was generated. The sparsity degree k was set at 10% of the measurements M, i.e. $k = 0.1\,M$. The number of Floating Point Operations (FLOP) were measured using the LightSpeed toolbox in MATLAB. For reconstruction, the proposed approach, OMP and Basis pursuit algorithms were used. The results are in Fig. 58, demonstrating a substantial FLOPs savings with the proposed scheme.

A circuit level simulation of the proposed dual-mode SAR ADC using HSPICE in 90 nm CMOS process showed that the proposed technique increases the energy dissipation of sampling only marginally: the total power consumption is 26 μW, whereas only 6 μW came from the low-resolution mode. On average, the SNR of the reconstructed signals was 55 dB.

6.6.1 Verification

To verify the functionality of the proposed solution, it was implemented using off-the-shelf components.We employed two ADCs that operate in parallel, one emulated the high-resolution AIC, and the other acted as the low-resolution ADC. An STm ARM Cortex®-M4 microcontroller was selected, due to its built-in pseudo-random number

generator unit, and two independent programmable SAR ADCs [175]. The MCU was programmed with a firmware to emulate a low-resolution Nyquist-rate ADC and an NUS AIC by the embedded SAR ADCs. Using the setup, sparse signals were sampled and constructed successfully [11].

6.7 Summary

Sub-Nyquist-Shannon rate pseudo-random sampling offers an opportunity to save energy in signal acquisition, but moves the power efficiency challenges to reconstruction at the back-end. In that respect, none of the studied greedy reconstruction techniques was superior.

Originally inspired by the puristic CS framework, the proposed novel approach combines lower resolution Nyquist-Shannon rate sampling with high resolution pseudo-random sampling. The solution gives substantial computational complexity savings for signal reconstruction, while slightly increasing the energy dissipation of signal acquisition at the front-end.

7 Summary

The energy efficiency challenge is ubiquitous from IoT nodes and wireless communications to computer graphics and super-computing. Reduced power dissipation contributes to designs with longer battery life, smaller size, and less cooling needs, and may in the long run even help to curb global warming.

In *computing*, reducing the voltage at which a given technology is operated is an attractive means for improving the energy efficiency. While the chip manufacturers guarantee correct operation when power is drawn from the nominal voltage, this applies within a given temperature range, and all the equivalently rated chips. The results in this thesis demonstrate up to 50% lower energy dissipation from reduced voltage, without any performance loss.

If the voltage is reduced further, a performance drop is inevitable. However, substantial improvements in energy efficiency become possible, e.g. up to 20x when the device is operating at near-threshold voltages.

Identifying the optimum voltage for reaching the performance for an application, without system failure or transient errors, is a big challenge. Techniques that employ a sub-circuit as proxy for the platform or those techniques based on static characterization do not address the global and dynamic variations that occur on the fly. Timing Error Detection systems that detect errors in situ are an effective solution, but increase the development costs substantially and are not applicable if the design has already been fabricated.

This thesis represents a shift of focus from hardware to software based error detection to support controlling the operation voltage in a dynamic manner. Algorithm Based Fault Tolerance was selected for error detection due to low overheads. Its utility was demonstrated with an FPGA based matrix multiplier implementation. The energy efficiency improved by a factor of two, while the performance was maintained.

While experimenting for low-voltage solutions, the Inverse Temperature Dependence phenomenon was observed. We speculate that in technologies such as 7nm and 5nm, the impact of ITD could be more pronounced due to the proximity of the supply and threshold voltages. How ITD can be leveraged for boosting performance in practical applications, when the ambient and chip temperature is high, remains to be investigated.

As an architecture level innovation, ABFT error checking was integrated into a systolic array design. The design was simulated with supply voltages down to the threshold voltage.This exposed opportunities for using ABFT in near-threshold designs.

In IoT nodes used for *sensing*, the ADCs can constitute a large share of energy consumption. This thesis proposes novel ADC designs that are able to exploit the characteristics of the signals to reduce power dissipation.

For signals exhibiting periodically low activity, a novel arithmetic tracking SAR ADC architecture is proposed to reduce the number of bit-cycles, and energy needed for resolving each sample. The scheme conserves energy also when used for oversampling.

For acquiring sparse signals in the context of a compressive sensing framework, the proposed approach intermixes the low bit-resolution Nyquist-Shannon rate and pseudo-randomly taken higher bit-resolution samples. This scheme slightly increases the energy dissipation in signal acquisition when compared to plain CS compliant sampling, but results in substantial savings in signal reconstruction.

Addressing the energy efficiency challenges of electronic systems, from simple sensor-readout nodes to super-computers, is important due to the growing amount of information technology on the globe. Hopefully, the contributions of this thesis are long-lasting, providing means for reducing the power dissipation of systems that employ advanced chip technologies of the future.

References

[1] M. Honkala, D. Korpi, and J. M. J. Huttunen, "Deeprx: Fully convolutional deep learning receiver," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3925–3940, 2021.

[2] E. Björnson, J. Hoydis, M. Kountouris, and M. Debbah, "Massive MIMO systems with non-ideal hardware: Energy efficiency, estimation, and capacity limits," *IEEE Transactions on Information Theory*, vol. 60, no. 11, pp. 7112–7139, 2014.

[3] L. Shang and R. P. Dick, "Thermal crisis: challenges and potential solutions," *IEEE Potentials*, vol. 25, no. 5, pp. 31–35, 2006.

[4] E. Linder, A. Grote, S. Varjo, N. Linder, M. Lebbad, M. Lundin, V. Diwan, J. Hannuksela, and J. Lundin, "On-chip imaging of schistosoma haematobium eggs in urine for diagnosis by computer vision," *PLoS Negl Trop Dis*, vol. 7, no. 12, p. e2547, 2013.

[5] M. Safarpour, I. Hautala, M. B. López, and O. Silvén, "Transport triggered array processor for vision applications," in *International Conference on Embedded Computer Systems*. Springer, 2019, pp. 361–372.

[6] M. Safarpour, R. Inanlou, and O. Silvén, "Algorithm level error detection in low voltage systolic array," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2021.

[7] M. Safarpour, R. Inanlou, O. Silvén, T. Rahkonen, and O. Shoaei, "A reconfigurable dual-mode tracking sar adc without analog subtraction," in *2019 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*. IEEE, 2019, pp. 28–32.

[8] M. Safarpour, L. Xun, G. V. Merrett, and O. Silven, "A high-level approach for energy efficiency improvement of fpgas by voltage trimming," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021.

[9] R. Inanlou, M. Safarpour, and O. Silvén, "Arithmetic tracking adaptive sar adc for signals with low-activity periods," *IEEE Access*, vol. 8, pp. 211 621–211 629, 2020.

[10] M. Safarpour, I. Hautala, and O. Silvén, "An embedded programmable processor for compressive sensing applications," in *2018 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC)*. IEEE, 2018, pp. 1–5.

[11] M. Safarpour, R. Inanlou, M. Charmi, O. Shoaei, and O. Silvén, "Adc-assisted random sampler architecture for efficient sparse signal acquisition," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 8, pp. 1590–1594, 2018.

[12] C. A. Mack, "Fifty years of moore's law," *IEEE Transactions on semiconductor manufacturing*, vol. 24, no. 2, pp. 202–207, 2011.

[13] V. Saripalli, S. Datta, V. Narayanan, and J. P. Kulkarni, "Variation-tolerant ultra low-power heterojunction tunnel fet sram design," in *2011 IEEE/ACM International Symposium on Nanoscale Architectures*. IEEE, 2011, pp. 45–52.

[14] M. Horowitz, T. Indermaur, and R. Gonzalez, "Low-power digital design," in *Proceedings of 1994 IEEE symposium on low power electronics*. IEEE, 1994, pp. 8–11.

[15] R. Andri, L. Cavigelli, D. Rossi, and L. Benini, "Yodann: An ultra-low power convolutional neural network accelerator based on binary weights," in *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2016, pp. 236–241.

[16] B. Reagen, P. Whatmough, R. Adolf, S. Rama, H. Lee, S. K. Lee, J. M. Hernández-Lobato, G.-Y. Wei, and D. Brooks, "Minerva: Enabling low-power, highly-accurate deep neural network accelerators," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2016, pp. 267–278.

[17] A. Ahmad and M. A. Pasha, "Optimizing Hardware Accelerated General Matrix-Matrix Multiplication for CNNs on FPGAs," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2020.

[18] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolić, *Digital integrated circuits: a design perspective*. Pearson education Upper Saddle River, NJ, 2003, vol. 7.

[19] R. G. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253–266, 2010.

[20] B. Razavi, *Fundamentals of microelectronics*. John Wiley & Sons, 2013.

[21] Q. Wu, M. Pedram, and X. Wu, "Clock-gating and its application to low power design of sequential circuits," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, no. 3, pp. 415–420, 2000.

[22] A. Hemani, T. Meincke, S. Kumar, A. Postula, T. Olsson, P. Nilsson, J. Oberg, P. Ellervee, and D. Lundqvist, "Lowering power consumption in clock by using globally asynchronous locally synchronous design style," in *Proceedings of the 36th annual ACM/IEEE Design Automation Conference*, 1999, pp. 873–878.

[23] D. Kim and H. Hoffmann, "Racing and pacing to idle: Minimizing energy under performance constraints," *Tech. Rep. TR-2014-10*, 2014.

[24] G. Papadimitriou, M. Kaliorakis, A. Chatzidimitriou, D. Gizopoulos, P. Lawthers, and S. Das, "Harnessing voltage margins for energy efficiency in multicore cpus," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, 2017, pp. 503–516.

[25] G. Papadimitriou, A. Chatzidimitriou, D. Gizopoulos, V. J. Reddi, J. Leng, B. Salami, O. S. Unsal, and A. C. Kestelman, "Exceeding conservative limits: A consolidated analysis on modern hardware margins," *IEEE Transactions on Device and Materials Reliability*, vol. 20, no. 2, pp. 341–350, 2020.

[26] J. Leng, A. Buyuktosunoglu, R. Bertran, P. Bose, and V. J. Reddi, "Safe limits on voltage reduction efficiency in gpus: a direct measurement approach," in *2015 48th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2015, pp. 294–307.

[27] B. Salami, O. S. Unsal, and A. C. Kestelman, "Comprehensive evaluation of supply voltage underscaling in fpga on-chip memories," in *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2018, pp. 724–736.

[28] B. Salami, E. B. Onural, I. E. Yuksel, F. Koc, O. Ergin, O. Kestelman, Adrian Cristal nd Unsal, H. Sarbazi Azad, and O. Mutlu, "An experimental study of reduced-voltage operation in modern fpgas for neural network acceleration," *Proceedings of the 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, vol. 98, no. 2, pp. 253–266, 2020.

[29] B. Salami, O. S. Unsal, and A. C. Kestelman, "On the resilience of rtl nn accelerators: Fault characterization and mitigation," in *2018 30th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*. IEEE, 2018, pp. 322–329.

[30] P. Koutsovasilis, C. Antonopoulos, N. Bellas, S. Lalis, G. Papadimitriou, A. Chatzidimitriou, and D. Gizopoulos, "The impact of cpu voltage margins on power-constrained execution," *IEEE Transactions on Sustainable Computing*, 2020.

[31] D. Gizopoulos, G. Papadimitriou, A. Chatzidimitriou, V. J. Reddi, B. Salami, O. S. Unsal, A. C. Kestelman, and J. Leng, "Modern hardware margins: Cpus, gpus, fpgas recent system-level studies," in *2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS)*. IEEE, 2019, pp. 129–134.

[32] M. Turnquist, M. Hiienkari, J. Mäkipää, R. Jevtic, E. Pohjalainen, T. Kallio, and L. Koskinen, "Fully integrated dc-dc converter and a 0.4 v 32-bit cpu with timing-error prevention supplied from a prototype 1.55 v li-ion battery," in *2015 Symposium on VLSI Circuits (VLSI Circuits)*. IEEE, 2015, pp. C320–C321.

[33] R. M. Swanson and J. D. Meindl, "Ion-Implanted Complementary MOS Transistors in Low-voltage Circuits," *IEEE Journal of Solid-State Circuits*, vol. 7, no. 2, pp. 146–153, 1972.

[34] A. Wang and A. Chandrakasan, "A 180-mv subthreshold fft processor using a minimum energy design methodology," *IEEE Journal of solid-state circuits*, vol. 40, no. 1, pp. 310–319, 2005.

[35] J. Mäkipää, M. J. Turnquist, E. Laulainen, and L. Koskinen, "Timing-error detection design considerations in subthreshold: An 8-bit microprocessor in 65 nm cmos," *Journal of Low Power Electronics and Applications*, vol. 2, no. 2, pp. 180–196, 2012.

[36] B. Zhai, L. Nazhandali, J. Olson, A. Reeves, M. Minuth, R. Helfand, S. Pant, D. Blaauw, and T. Austin, "A 2.60 pj/inst subthreshold sensor processor for optimal energy efficiency," in *2006 Symposium on VLSI Circuits, 2006. Digest of Technical Papers.* IEEE, 2006, pp. 154–155.

[37] K. Singh and J. P. de Gyvez, "Twenty years of near/sub-threshold design trends and enablement," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 1, pp. 5–11, 2020.

[38] P. Rathnala, T. Wilmshurst, and A. Kharaz, "Timing error detection and correction for power efficiency: an aggressive scaling approach," *IET Circuits, Devices & Systems*, vol. 12, no. 6, pp. 707–712, 2018.

[39] K. K. Chang, A. G. Yağlıkçı, S. Ghose, A. Agrawal, N. Chatterjee, A. Kashyap, D. Lee, M. O'Connor, H. Hassan, and O. Mutlu, "Understanding reduced-voltage operation in modern dram devices: Experimental characterization, analysis, and mechanisms," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 1, pp. 1–42, 2017.

[40] H. Hanson, S. W. Keckler, S. Ghiasi, K. Rajamani, F. Rawson, and J. Rubio, "Thermal response to dvfs: Analysis with an intel pentium m," in *Proceedings of the 2007 international symposium on Low power electronics and design (ISLPED'07)*. IEEE, 2007, pp. 219–224.

[41] B. Bowhill, B. Stackhouse, N. Nassif, Z. Yang, A. Raghavan, O. Mendoza, C. Morganti, C. Houghton, D. Krueger, O. Franza *et al.*, "The xeon® processor e5-2600 v3: A 22 nm 18-core product family," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 1, pp. 92–104, 2015.

[42] P. Pillai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," in *Proceedings of the eighteenth ACM symposium on Operating systems principles*, 2001, pp. 89–102.

[43] J. Tyhach, M. Hutton, S. Atsatt, A. Rahman, B. Vest, D. Lewis, M. Langhammer, S. Shumarayev, T. Hoang, A. Chan *et al.*, "Arria™ 10 device architecture," in *2015 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 2015, pp. 1–8.

[44] I. Ahmed, "Dynamic voltage scaling for current and future fpgas," Ph.D. dissertation, University of Toronto (Canada), 2020.

[45] I. Ahmed, S. Zhao, O. Trescases, and V. Betz, "Measure twice and cut once: Robust dynamic voltage scaling for fpgas," in *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2016, pp. 1–11.

[46] I. Ahmed, L. L. Shen, and V. Betz, "Becoming more tolerant: Designing fpgas for variable supply voltage," in *2019 29th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2019, pp. 1–8.

[47] C. T. Chow, L. S. M. Tsui, P. H. W. Leong, W. Luk, and S. J. Wilton, "Dynamic voltage scaling for commercial fpgas," in *Proceedings. 2005 IEEE International Conference on Field-Programmable Technology, 2005.* IEEE, 2005, pp. 173–180.

[48] J. M. Levine, E. Stott, G. A. Constantinides, and P. Y. Cheung, "Online measurement of timing in circuits: For health monitoring and dynamic voltage & frequency scaling," in *2012 IEEE 20th International Symposium on Field-Programmable Custom Computing Machines*. IEEE, 2012, pp. 109–116.

[49] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner *et al.*, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proceedings. 36th Annual IEEE/ACM International Symposium on Microarchitecture, 2003. MICRO-36.* IEEE, 2003, pp. 7–18.

[50] "Dynamic Margining: The Minima Approach to Near-threshold Design," Minima Processor Oy , Tech. Rep., December 2020. [Online]. Available: https://minimaprocessor.com/wp-content/uploads/2017/11/minima-margining-white-paper.pdf

[51] E. Stott, J. M. Levine, P. Y. Cheung, and N. Kapre, "Timing fault detection in fpga-based circuits," in *2014 IEEE 22nd Annual International Symposium on Field-Programmable Custom Computing Machines*. IEEE, 2014, pp. 96–99.

[52] H. Bolcskei, "MIMO-OFDM wireless systems: basics, perspectives, and challenges," *IEEE wireless communications*, vol. 13, no. 4, pp. 31–37, 2006.

[53] J. Xu, "Systolic array for universal matrix arithmetic," Ph.D. dissertation, Northeastern University, 2020.

[54] J. He, H. Wymeersch, L. Kong, O. Silvén, and M. Juntti, "Large intelligent surface for positioning in millimeter wave MIMO systems," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.

[55] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE communications magazine*, vol. 52, no. 2, pp. 186–195, 2014.

[56] Z. Wang, I. Stupia, and L. Vandendorpe, "Energy efficient precoder design for MIMO-OFDM with rate-dependent circuit power," *International Communications Conference*, pp. 1897–1902, 2015.

[57] K. Guo, S. Zeng, J. Yu, Y. Wang, and H. Yang, "A survey of fpga-based neural network accelerator," *arXiv preprint arXiv:1712.08934*, 2017.

[58] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[59] Z. Li, Y. Wang, T. Zhi, and T. Chen, "A survey of neural network accelerators," *Frontiers of Computer Science*, vol. 11, no. 5, pp. 746–761, 2017.

[60] A. Vasudevan, A. Anderson, and D. Gregg, "Parallel multi channel convolution using general matrix multiplication," in *2017 IEEE 28th international conference on application-specific systems, architectures and processors (ASAP)*. IEEE, 2017, pp. 19–24.

[61] N. P. Jouppi, C. Young, N. Patil, A. Patterson *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 2017, pp. 1–12.

[62] V. Strassen, "Relative bilinear complexity and matrix multiplication." *Journal für die reine und angewandte Mathematik*, vol. 375, pp. 406–443, 1987.

[63] K.-H. Huang and J. A. Abraham, "Algorithm-based fault tolerance for matrix operations," *IEEE transactions on computers*, vol. 100, no. 6, pp. 518–528, 1984.

[64] K. Zhao, S. Di, S. Li, X. Liang, Y. Zhai, J. Chen, K. Ouyang, F. Cappello, and Z. Chen, "Algorithm-based fault tolerance for convolutional neural networks," *arXiv preprint arXiv:2003.12203*, 2020.

[65] G. Bosilca, R. Delmas, J. Dongarra, and J. Langou, "Algorithm-based fault tolerance applied to high performance computing," *Journal of Parallel and Distributed Computing*, vol. 69, no. 4, pp. 410–416, 2009.

[66] M. Vijay and R. Mittal, "Algorithm-based fault tolerance: a review," *Microprocessors and Microsystems*, vol. 21, no. 3, pp. 151–161, 1997.

[67] L. Fiore, "Design of a fault tolerant risc-v instruction execute stage for safety critical applications," Ph.D. dissertation, Politecnico di Torino, 2021.

[68] M. Valinataj, A. Mohammadnezhad, and J. Nurmi, "A low-cost high-speed self-checking carry select adder with multiple-fault detection," *Microelectronics Journal*, vol. 81, pp. 16–27, 2018.

[69] Z. Zhao, G. Min, W. Gao, Y. Wu, H. Duan, and Q. Ni, "Deploying edge computing nodes for large-scale iot: A diversity aware approach," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3606–3614, 2018.

[70] S. B. Wicker and V. K. Bhargava, *Reed-Solomon codes and their applications*. John Wiley & Sons, 1999.

[71] M. Lentmaier and K. S. Zigangirov, "On generalized low-density parity-check codes based on hamming component codes," *IEEE communications letters*, vol. 3, no. 8, pp. 248–250, 1999.

[72] P. Prata and J. G. Silva, "Algorithm based fault tolerance versus result-checking for matrix computations," in *Digest of Papers. Twenty-Ninth Annual International Symposium on Fault-Tolerant Computing (Cat. No. 99CB36352)*. IEEE, 1999, pp. 4–11.

[73] R. F. Molanes, K. Amarasinghe, J. Rodriguez-Andina, and M. Manic, "Deep learning and reconfigurable platforms in the internet of things: Challenges and opportunities in algorithms and hardware," *IEEE industrial electronics magazine*, vol. 12, no. 2, pp. 36–49, 2018.

[74] R. Hameed, W. Qadeer, M. Wachs, O. Azizi, A. Solomatnikov, B. C. Lee, S. Richardson, C. Kozyrakis, and M. Horowitz, "Understanding sources of inefficiency in general-purpose chips," in *Proceedings of the 37th annual international symposium on Computer architecture*, 2010, pp. 37–47.

[75] N. Tarafdar, T. Lin, E. Fukuda, H. Bannazadeh, A. Leon-Garcia, and P. Chow, "Enabling flexible network fpga clusters in a heterogeneous cloud data center," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2017, pp. 237–246.

[76] J. M. Mbongue and C. Bobda, "Accommodating multi-tenant fpgas in the cloud," in *2020 IEEE 28th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*. IEEE, 2020, pp. 214–214.

[77] K. Chang, O. Mutlu, A. G. Yaglikçi, S. Ghose, A. Agrawal, N. Chatterjee, A. Kashyap, D. Lee, M. O'Connor, and H. Hassan, "Understanding reduced-voltage operation in modern dram devices: Experimental characterization, analysis, and mechanisms," *ACM SIGMETRICS Performance Evaluation Review*, vol. 44, pp. 52–52, 06 2017.

[78] W. Jiang, H. Yu, J. Zhang, J. Wu, S. Luo, and Y. Ha, "Optimizing energy efficiency of cnn-based object detection with dynamic voltage and frequency scaling," *Journal of Semiconductors*, vol. 41, no. 2, p. 022406, 2020.

[79] K. Maragos, G. Lentaris, and D. Soudris, "A pvt-aware voltage scaling method for energy efficient fpgas," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2021, pp. 1–5.

[80] M. Hosseinabady and J. L. Nunez-Yanez, "Run-time power gating in hybrid arm-fpga devices," in *2014 24th International Conference on Field Programmable Logic and Applications (FPL)*. IEEE, 2014, pp. 1–6.

[81] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for deep-submicron FPGAs*. Springer Science & Business Media, 2012, vol. 497.

[82] Xilinx Inc., "7000 All Programmable SoC ZC702 Evaluation Kit," Xilinx Inc., Tech. Rep., 2015. [Online]. Available: http://www.xilinx.com/products/boards-and-kits/ek-z7-zc706-g.html

[83] "Python productivity for zynq (pynq)," 2021. [Online]. Available: https://pynq.readthedocs.io/en/v2.6.1/

[84] Z. Navabi, *Digital design and implementation with field programmable devices*. Springer Science & Business Media, 2004.

[85] Xilinx, "Vivado design suite user guide-high-level synthesis," 2014.

[86] V. Kathail, J. Hwang, W. Sun, Y. Chobe, T. Shui, and J. Carrillo, "Sdsoc: A higher-level programming environment for zynq soc and ultrascale+ mpsoc," in *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, 2016, pp. 4–4.

[87] S. Ahmad, S. Subramanian, V. Boppana, S. Lakka, F.-H. Ho, T. Knopp, J. Noguera, G. Singh, and R. Wittig, "Xilinx first 7 nm device: Versal ai core (vc1902)." in *Hot Chips Symposium*, 2019, pp. 1–28.

[88] N. Persson, "Dc/dc pmbus compliance tester," Master's thesis, Chalmers University, 2012.

[89] R. V. White, "Pmbus: A decade of growth: An open-standards success," *IEEE Power Electronics Magazine*, vol. 1, no. 3, pp. 33–39, 2014.

[90] Xilinx Inc., "7000 All Programmable SoC ZC702 Evaluation Kit," Xilinx Inc., Tech. Rep., 2015. [Online]. Available: http://www.xilinx.com/products/boards-and-kits/ek-z7-zc706-g.html

[91] D. Bagni, A. Di Fresco, J. Noguera, and F. Vallina, "A zynq accelerator for floating point matrix multiplication designed with vivado hls," *Application note, January*, 2016.

[92] Xilinx, "Systolic array implementation," https://github.com/Xilinx/SDSoC_Examples/tree/master/cpp/getting_started/systolic_array, 2019.

[93] K. Neshatpour, W. Burleson, A. Khajeh, and H. Homayoun, "Enhancing power, performance, and energy efficiency in chip multiprocessors exploiting inverse thermal dependence," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 4, pp. 778–791, 2018.

[94] J. M. Daga, E. Ottaviano, and D. Auvergne, "Temperature effect on delay for low voltage applications [cmos ics]," in *Proceedings Design, Automation and Test in Europe*. IEEE, 1998, pp. 680–685.

[95] F. Lima, L. Carro, and R. Reis, "Designing fault tolerant systems into sram-based fpgas," in *Proceedings of the 40th annual Design Automation Conference*, 2003, pp. 650–655.

[96] I. Ahmed, S. Zhao, J. Meijers, O. Trescases, and V. Betz, "Froc 2.0: Automatic bram and logic testing to enable dynamic voltage scaling for fpga applications," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 12, no. 4, pp. 1–28, 2019.

[97] L. Koskinen, M. Hiienkari, J. Mäkipää, and M. J. Turnquist, "Implementing minimum-energy-point systems with adaptive logic," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 4, pp. 1247–1256, 2015.

[98] K. Neshatpour, H. Homayoun, A. Khajeh, and W. Burleson, "Revisiting dynamic thermal management exploiting inverse thermal dependence," in *Proceedings of the 25th edition on Great Lakes Symposium on VLSI*, 2015, pp. 385–390.

[99] M. Safarpour, T. Deng, J. Massingham, L. Xun, M. Sabokrou, and O. Silvén, "Low-voltage energy efficient neural inference by leveraging fault detection techniques," in *2021 IEEE Nordic Circuits and Systems Conference (NORCAS): NORCHIP and International Symposium of System-on-Chip (SoC)*. IEEE, 2021, pp. 1–5.

[100] L. K. Draghetti, F. F. dos Santos, L. Carro, and P. Rech, "Detecting errors in convolutional neural networks using inter frame spatio-temporal correlation," in *2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS)*. IEEE, 2019, pp. 310–315.

[101] W. Wang, K. Guo, M. Gu, Y. Ma, and Y. Wang, "A universal fpga-based floating-point matrix processor for mobile systems," in *2014 International Conference on Field-Programmable Technology (FPT)*. IEEE, 2014, pp. 139–146.

[102] H. Qi, O. Ayorinde, and B. H. Calhoun, "An ultra-low-power fpga for iot applications," in *2017 IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S)*. IEEE, 2017, pp. 1–3.

[103] J. M. Levine, E. Stott, and P. Y. Cheung, "Dynamic voltage & frequency scaling with online slack measurement," in *Proceedings of the 2014 ACM/SIGDA international symposium on Field-programmable gate arrays*, 2014, pp. 65–74.

[104] T. Marty, T. Yuki, and S. Derrien, "Safe overclocking for cnn accelerators through algorithm-level error detection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 12, pp. 4777–4790, 2020.

[105] K. Singh, B. de Bruin, H. Jiao, J. Huisken, H. Corporaal, and J. P. de Gyvez, "Converter-free power delivery using voltage stacking for near/subthreshold operation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2021.

[106] M. Horowitz, "Computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE, 2014, pp. 10–14.

[107] O. Silven and K. Jyrkkä, "Observations on power-efficiency trends in mobile communication devices," *EURASIP Journal on Embedded Systems*, vol. 2007, pp. 1–10, 2007.

[108] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "Eie: Efficient inference engine on compressed deep neural network," *ACM SIGARCH Computer Architecture News*, vol. 44, no. 3, pp. 243–254, 2016.

[109] H. T. Kung and C. E. Leiserson, "Systolic arrays for (vlsi)." Carnegie-mellon university, dept. of computer science, Pittsburgh, Tech. Rep., 1978.

[110] M. Arnold and H. Corporaal, "Designing domain-specific processors," in *Proceedings of the ninth international symposium on Hardware/software codesign*, 2001, pp. 61–66.

[111] H.-W. Lang, "The instruction systolic array—a parallel architecture for vlsi," *Integration*, vol. 4, no. 1, pp. 65–74, 1986.

[112] O. Mutlu, "Design of digital circuits lecture 23a: Systolic arrays and beyond," *Lecture notes*, 2018.

[113] J. Xu and M. Leeser, "High-level and compact design of cross-channel lte downlink channel encoder," in *International Conference on Cognitive Radio Oriented Wireless Networks*. Springer, 2018, pp. 15–24.

[114] M. Leeser, S. Handagala, M. Mohamed, J. Xu, and M. Onabajo, "An fpga design technique to receive multiple wireless protocols with the same rf front end," in *2019 Wireless Days (WD)*. IEEE, 2019, pp. 1–6.

[115] H.-T. Kung, "Why systolic architectures?" *Computer*, vol. 15, no. 1, pp. 37–46, 1982.

[116] C. S. Wallace, "A suggestion for a fast multiplier," *IEEE Transactions on electronic Computers*, no. 1, pp. 14–17, 1964.

[117] N. Burgess, "Fast ripple-carry adders in standard-cell cmos vlsi," in *2011 IEEE 20th Symposium on Computer Arithmetic*. IEEE, 2011, pp. 103–111.
[118] B. Razavi, *Fundamentals of microelectronics*. John Wiley & Sons, 2021.
[119] Y. Cheng and C. Hu, *MOSFET modeling & BSIM3 user's guide*. Springer Science & Business Media, 1999.
[120] R. J. Baker, *CMOS: mixed-signal circuit design*. John Wiley & sons, 2008.
[121] M. Rathore, P. Milder, and E. Salman, "Error probability models for voltage-scaled multiply-accumulate units," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2020.
[122] M. Hiienkari, N. Gupta, J. Teittinen, J. Simonsson, M. Turnquist, J. Eriksson, R. Anttila, O. Myllynen, H. Rämäkkö, S. Mäkikyrö *et al.*, "A 0.4-0.9 v, 2.87 pj/cycle near-threshold arm cortex-m3 cpu with in-situ monitoring and adaptive-logic scan," in *2020 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS)*. IEEE, 2020, pp. 1–3.
[123] P. N. Whatmough, S. K. Lee, H. Lee, S. Rama, D. Brooks, and G.-Y. Wei, "14.3 a 28nm soc with a 1.2 ghz 568nj/prediction sparse deep-neural-network engine with> 0.1 timing error rate tolerance for iot applications," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2017, pp. 242–243.
[124] J. Zhang, K. Rangineni, Z. Ghodsi, and S. Garg, "Thundervolt: enabling aggressive voltage underscaling and timing error resilience for energy efficient deep learning accelerators," in *Proceedings of the 55th Annual Design Automation Conference*, 2018, pp. 1–6.
[125] N. D. Gundi, T. Shabanian, P. Basu, P. Pandey, S. Roy, K. Chakraborty, and Z. Zhang, "Effort: Enhancing energy efficiency and error resilience of a near-threshold tensor processing unit," in *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2020, pp. 241–246.
[126] A. F. Timan, *Theory of approximation of functions of a real variable*. Elsevier, 2014.
[127] T. Nyländen, J. Janhunen, J. Hannuksela, and O. Silvén, "Fpga based application specific processing for sensor nodes," in *2011 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation*. IEEE, 2011, pp. 118–123.
[128] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad hoc networks*, vol. 7, no. 3, pp. 537–568, 2009.
[129] B. Murmann, "The race for the extra decibel: A brief review of current adc performance trajectories," *IEEE Solid-State Circuits Magazine*, vol. 7, no. 3, pp. 58–66, 2015.
[130] H. Nyquist, "Certain topics in telegraph transmission theory," *Transactions of the American Institute of Electrical Engineers*, vol. 47, no. 2, pp. 617–644, 1928.
[131] R. G. Lyons, *Streamlining Digital Signal Processing*. Wiley Online Library, 2012.
[132] M. Mishali and Y. C. Eldar, "Sub-nyquist sampling," *IEEE Signal Processing Magazine*, vol. 28, no. 6, pp. 98–124, 2011.
[133] K. Georgiou, Z. Chamski, K. Nikov, and K. Eder, "A comprehensive and accurate energy model for arm's cortex-m0 processor," *arXiv preprint arXiv:2104.01055*, 2021.
[134] D. S. AD7980, "16-bit, 1 msps pulsar adc in msop/lfcsp."
[135] W. Kester, "Understand sinad, enob, snr, thd, thd+ n, and sfdr so you don't get lost in the noise floor," *MT-003 Tutorial, www. analog. com/static/importedfiles/tutorials/MT-003. pdf*, 2009.
[136] B. Razavi, "A tale of two adcs: Pipelined versus sar," *IEEE Solid-State Circuits Magazine*, vol. 7, no. 3, pp. 38–46, 2015.
[137] W. Kester and A. D. I. Engineeri, *Data conversion handbook*. Newnes, 2005.
[138] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on selected areas in communications*, vol. 17, no. 4, pp. 539–550, 1999.
[139] W. Kester, *Mixed-signal and DSP design techniques*. Newnes, 2003.

[140] M. Saberi, R. Lotfi, K. Mafinezhad, and W. A. Serdijn, "Analysis of power consumption and linearity in capacitive digital-to-analog converters used in successive approximation adcs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 58, no. 8, pp. 1736–1748, 2011.

[141] F. M. Yaul and A. P. Chandrakasan, "A 10 bit sar adc with data-dependent energy reduction using lsb-first successive approximation," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 12, pp. 2825–2834, 2014.

[142] S. Yim, Y. Park, H. Yang, and S. Kim, "Power efficient sar adc adaptive to input activity for ecg monitoring applications," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2017, pp. 1–4.

[143] B. Chen, F. Yaul, Z. Tan, and L. Fernando, "An adaptive sar adc for dc to nyquist rate signals," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2018, pp. 1–5.

[144] N. Wood and N. Sun, "Predicting adc: A new approach for low power adc design," in *2014 IEEE Dallas Circuits and Systems Conference (DCAS)*. IEEE, 2014, pp. 1–4.

[145] M. Nasserian, A. Peiravi, and F. Moradi, "An adaptive-resolution signal-specific adc for sensor-interface applications," *Analog Integrated Circuits and Signal Processing*, vol. 98, no. 1, pp. 125–135, 2019.

[146] S.-L. Chen, J. F. Villaverde, H.-Y. Lee, D. W.-Y. Chung, T.-L. Lin, C.-H. Tseng, and K.-A. Lo, "A power-efficient mixed-signal smart adc design with adaptive resolution and variable sampling rate for low-power applications," *IEEE Sensors Journal*, vol. 17, no. 11, pp. 3461–3469, 2017.

[147] B. Chen, L. D. Fernando, and Z. Tan, "Method of performing analog-to-digital conversion," Sep. 3 2019, uS Patent 10,404,264.

[148] F. M. Yaul and A. P. Chandrakasan, "11.3 a 10b 0.6 nw sar adc with data-dependent energy savings using lsb-first successive approximation," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*. IEEE, 2014, pp. 198–199.

[149] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[150] Y. Chen, Y. C. Eldar, and A. J. Goldsmith, "Shannon meets nyquist: Capacity of sampled gaussian channels," *IEEE Transactions on Information Theory*, vol. 59, no. 8, pp. 4889–4914, 2013.

[151] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE signal processing magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[152] M. Safarpour, "Design and simulation of analog to information convertors based on RMPI structure," Master thesis, Ministry of science and technolgoy, university of Zanjan, Faculty of Engineering, 1394. [Online]. Available: https://ganj.irandoc.ac.ir

[153] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[154] G. K. Wallace, "The jpeg still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[155] F. Chen, A. P. Chandrakasan, and V. M. Stojanovic, "Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 3, pp. 744–756, 2012.

[156] M. Verhelst and A. Bahai, "Where analog meets digital: Analog-to-information conversion and beyond," *IEEE Solid-state circuits magazine*, vol. 7, no. 3, pp. 67–80, 2015.

[157] F. Pareschi, P. Albertini, G. Frattini, M. Mangia, R. Rovatti, and G. Setti, "Hardware-algorithms co-design and implementation of an analog-to-information converter for

biosignals based on compressed sensing," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 10, no. 1, pp. 149–162, 2016.

[158] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on information theory*, vol. 56, no. 4, pp. 1982–2001, 2010.

[159] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and computational harmonic analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[160] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on information theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[161] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "From denoising to compressed sensing," *IEEE Transactions on Information Theory*, vol. 62, no. 9, pp. 5117–5144, 2016.

[162] T. Blumensath and M. E. Davies, "Normalized iterative hard thresholding: Guaranteed stability and performance," *IEEE Journal of selected topics in signal processing*, vol. 4, no. 2, pp. 298–309, 2010.

[163] A. Kyrillidis and V. Cevher, "Recipes on hard thresholding methods," in *2011 4th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 2011, pp. 353–356.

[164] P. Jääskeläinen, T. Viitanen, J. Takala, and H. Berg, "Hw/sw co-design toolset for customization of exposed datapath processors," in *Computing platforms for software-defined radio*. Springer, 2017, pp. 147–164.

[165] I. Hautala, J. Boutellier, and O. Silven, "Programmable 28nm coprocessor for hevc/h. 265 in-loop filters," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2016, pp. 1570–1573.

[166] M. A. Davenport, J. N. Laska, J. R. Treichler, and R. G. Baraniuk, "The pros and cons of compressive sensing for wideband signal acquisition: Noise folding versus dynamic range," *IEEE Transactions on Signal Processing*, vol. 60, no. 9, pp. 4628–4642, 2012.

[167] T. Strohmer, "Measure what should be measured: progress and challenges in compressive sensing," *IEEE Signal Processing Letters*, vol. 19, no. 12, pp. 887–893, 2012.

[168] E. Arias-Castro and Y. C. Eldar, "Noise folding in compressed sensing," *IEEE Signal Processing Letters*, vol. 18, no. 8, pp. 478–481, 2011.

[169] M. F. Duarte and R. G. Baraniuk, "Spectral compressive sensing," *Applied and Computational Harmonic Analysis*, vol. 35, no. 1, pp. 111–129, 2013.

[170] M. Wakin, S. Becker, E. Nakamura, M. Grant, E. Sovero, D. Ching, J. Yoo, J. Romberg, A. Emami-Neyestanak, and E. Candes, "A nonuniform sampler for wideband spectrally-sparse environments," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 516–529, 2012.

[171] T.-S. Chen, H.-C. Kuo, and A.-Y. Wu, "A 232–1996-ks/s robust compressive sensing reconstruction engine for real-time physiological signals monitoring," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 1, pp. 307–317, 2018.

[172] Y.-Z. Wang, Y.-P. Wang, Y.-C. Wu, and C.-H. Yang, "A 12.6 mw, 573–2901 ks/s reconfigurable processor for reconstruction of compressively sensed physiological signals," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 10, pp. 2907–2916, 2019.

[173] V. Q. Nguyen, W. H. Son, M. Parfieniuk, L. T. N. Trung, and S. Y. Park, "High-throughput and low-area implementation of orthogonal matching pursuit algorithm for compressive sensing reconstruction," *Etri Journal*, vol. 42, no. 3, pp. 376–387, 2020.

[174] K. Adhikari and J. R. Buck, "Spatial spectral estimation with product processing of a pair of colinear arrays," *IEEE Transactions on Signal Processing*, vol. 65, no. 9, pp. 2389–2401, 2017.

[175] H. Seo and R. Azarderakhsh, "Curve448 on 32-bit arm cortex-m4," in *International Conference on Information Security and Cryptology*. Springer, 2020, pp. 125–139.