ЕЛЕКТРОНИКА (ELECTRONICS)

***

Publisher

Faculty of Electrical Engineering
University of Banja Luka
Patre 5, 78000 Banja Luka
Bosnia and Herzegovina

Phone: +387 51 211-820
Fax: +387 51 211-408
Email: info@etf.unibl.org
Web: www.etf.unibl.org

Founder and Honorary Editor-in-Chief

BRANKO DOKIĆ
University of Banja Luka
Bosnia and Herzegovina
nail: branko.dokic@etf.unibl.org

Editor-in-Chief

MLADEN KNEŽIĆ
University of Banja Luka
Bosnia and Herzegovina
Email: els-eic@etf.unibl.org

Managing Editors

MITAR SIMIĆ
ALEKSANDAR PAJKANOVIĆ
University of Banja Luka
Bosnia and Herzegovina
Email: els-journal@etf.unibl.org

Editorial Board

ZDENKA BABIĆ
University of Banja Luka, Bosnia and Herzegovina

BRANKO BLANUŠA
University of Banja Luka, Bosnia and Herzegovina

OCTAVIO NIETO-TALADRIZ GARCIA
Polytechnic University of Madrid, Spain

ZORAN ĐURIĆ
University of Banja Luka, Bosnia and Herzegovina

ZORAN JAKŠIĆ
IHTM, Serbia

TOM J. KAZMIERSKI
University of Southampton, United Kingdom

DANILO MANDIĆ
Imperial College, London, United Kingdom

DARKO MARČETIĆ
University of Novi Sad, Serbia

SOHRAB MIRSAEIDI
Beijing Jiaotong University, China

PREDRAG PEJOVIĆ
University of Belgrade, Serbia

TATJANA PEŠIĆ-BRĐANIN
University of Banja Luka, Bosnia and Herzegovina

VLADIMIR RISOJEVIĆ
University of Banja Luka, Bosnia and Herzegovina

MITAR SIMIĆ
University of Banja Luka, Bosnia and Herzegovina

PAUL SOTIRIADIS
National Technical University of Athens, Greece

ROBERT ŠOBOT
Western University, Canada

VOLKER ZERBE
University of Applied Sciences of Erfurt, Germany

Honorary Editorial Board

GOCE ARSOV
St. Cyril and Methodius University
North Macedonia

MILORAD BOŽIĆ
University of Banja Luka
Bosnia and Herzegovina

VLADIMIR KATIĆ
University of Novi Sad, Serbia

VANČO LITOVSKI
University of Niš, Serbia

BRATISLAV MILOVANOVIĆ
University of Niš, Serbia

VOJIN OKLOBDŽIJA
University of Texas at Austin, USA

SLOBODAN VUKOSAVIĆ
University of Belgrade, Serbia

***

Layout Editor

Dragana Pupac

Publication frequency: Two issues a year
ISSN: 1450-5843 (Print), 2831-0128 (Online)
UDC: 621.38
Number of printed copies: 50

ELECTRONICS, VOL. 28, NO. 1, JUNE 2024

EDITOR'S COLUMN 1
Mladen Knezic

AN APPROACH TO DESIGN A LOW POWER HIGH-SPEED FULL ADDER CIRCUIT BASED ON LOGICAL EFFORT 3
S. Goswami, V. Chandana, and A. Dandapat

A NOVEL SEPARABLE CONVOLUTION ARCHITECTURE FOR IMAGE PROCESSING APPLICATIONS 12
Jaiza Hassan and Burhan Khurshid

ACCELERATED CURVE FITTING APPROACH FOR DENOISED NQR SIGNAL PARAMETER ESTIMATION 22
K. Thulasiram Varma and Rangababu Peesapati

Editor's Column

Mladen Knezic

> To have friends means to accept that there are more beautiful, smarter, and better people than you. Those, who cannot accept that, have no friends.
>
> *Duško Radović*

Editorial Letter
DOI: 10.53314/ELS2428001K

THIS year and June issue of the journal we start with three papers focused on design of digital electronics circuits.

In the first paper, entitled "An Approach to Design a Low Power High-Speed Full Adder Circuit Based on Logical Effort," authored by S. Goswami, V. Chandana, and A. Dandapat, authors proposed a low-power full adder circuit and compared it with three established low power full adders. All the circuits were designed utilizing Cadence Virtuoso software and GPDK 45nm technology and comparison was based on delay, average power, power-delay product, and transistor count. Presented results demonstrates that the proposed design brings better performance than other designs in terms of average power, delay of sum, and delay of carry for 34.23%, 26.81%, and 4.33%, respectively.

The paper entitled "A Novel Separable Convolution Architecture for Image Processing Applications," by J. Hassan and B. Khurshid, presents a novel separable Convolution architecture based on the folding transformation that is suitable for image applications, which inherently require efficient two-dimensional (2D) convolution calculation (e.g., image smoothening, image sharpening, feature extraction, image enhancement, object recognition, etc.). The proposed architecture has been implemented in hardware using Xilinx Artix-7 xc7a35tcpg236-1 FPGA device. The results indicate that the implemented solution is beneficial over the existing architectures regarding on-chip resource utilization, power consumption, critical path delay, and external memory bandwidth (EMB).

The last paper, entitled "Accelerated Curve Fitting Approach for Denoised NQR Signal Parameter Estimation," authored by K. Thulasiram Varma and R. Peesapati, demonstrates a feasible approach to extract parameters from exponential decay Nuclear Quadrupole Resonance (NQR) signals using derivative-free optimization approaches on a PYNQ-Z2 FPGA board. As supported by the presented results obtained using the hardware implementation, 0.12 seconds is required to extract the amplitude, frequency, decay constant, and phase from $5 \cdot 10^{-3}$ sec signals with the residual error in the order of 10^{-5}.

Finally, I thank the authors for their contribution to this issue of the journal and send great appreciation to all the reviewers who participated in the editorial process by providing valuable comments in timely manner to the editors and authors.

An Approach to Design a Low Power High-Speed Full Adder Circuit Based on Logical Effort

S. Goswami, V. Chandana, and A. Dandapat

***Abstract*—The logical effort method is a technique for accurately estimating the delay of a CMOS circuit. This method is computed as a ratio of capacitances. In this work we propose a low-power full adder circuit and compare it with three established low power full adder circuits. The circuits were designed utilizing Cadence Virtuoso software and GPDK 45nm technology. A comparison was conducted based on delay, average power, power delay product, and transistor count. In this work, all circuits are resized to achieve minimum delay according to the logical effort. It has been shown that the proposed design (design 4) is showing better performance in terms of average power, delay of sum and delay of carry than other designs by 34.23%, 26.81%, and 4.33%, respectively.**

***Index Terms*—CMOS logic, Full adder, GPDK45, Logical effort, Low power.**

Original Research Paper
DOI: 10.53314/ELS2428003G

I. Introduction

THE field of logical effort calculation plays a pivotal role in the design and optimization of CMOS logic gates. The logical effort of a gate is a critical metric that determines the relative performance of different gate configurations. In this context, it is essential to build upon the established methodologies and insights garnered from prior research [1], [2]. Designers are aiming to create full adders with as few transistors as feasible to get the best power consumption, delay, and area. Many full adders with different logic types were published [3]–[6]. Chip designers must decide on the appropriate circuit layout, transistor size, and logic stage count to minimize delay in digital circuits. Logical effort addresses all these concerns. The aforementioned is the proportion of capacitances at the gate that generate an equivalent current to that of an inverter [7].

A. Logic Gate Delay

The logical effort of a circuit can be defined by the following equations.

$$d = \frac{dabs}{\tau} \tag{1}$$

where $dabs$ denotes the absolute delay and $\tau = 3RC$. Delay comes with two components

$$d = f + p \tag{2}$$

Manuscript received on April 11th, 2023. Received in revised form on September 28th, November 4th, and November 30th, 2023. Accepted for publication on December 29th, 2023.

S. Goswami, V. Chandana, and A. Dandapat is with the Department of Electronics and Communication Engineering, National Institute of Technology, Meghalaya, Shillong, 793003 India (e-mail: shyamosreegoswami@gmail.com; anup.dandapat@gmail.com).

Effort delay is $f = gh$ (also known as stage effort), so that the delay is expressed as

$$d = gh + p \tag{3}$$

where:

- g is logical effort, which measures relative ability of a gate to deliver current,
- h is electrical effort, which represents output to input capacitance ratio (sometimes termed as fan-out) [7], [8]

$$h = \frac{Cout}{Cin}, \tag{4}$$

- p is parasitic delay, which represents delay of a gate driving no load (it is set by internal parasitic capacitance).

The definitions of all terms in logical effort are stated in Table I.

B. Logical Effort Calculation

Several seminal works, including references [7] and [11], have laid the groundwork for logical effort calculations. These contributions have elucidated the intricate relationship between gate characteristics, such as capacitance ratios, and the resulting logical effort. Moreover, the fundamental equation, $C_g = L \cdot W \cdot C_{ox}$, establishes a crucial link between capacitance, oxide capacitance, and width, further underlining the significance of width adjustments in logical effort optimization.

In the realm of CMOS processes, the choice of transistors, with NMOS and PMOS serving as pull-down and pull-up transistors, respectively, is a well-established convention. This practice is underpinned by the inherent disparities in carrier mobility, with electrons dominating in NMOS and holes in PMOS. The implications of these distinctions are encapsulated

TABLE I
Definitions Of All Terms In Logical Effort [7]

Term	Stage	Path
Number of Stage	1	N
Logical-Effort	g	$G = \pi g_i$
Electrical-Effort	$h = \frac{C_{out}}{C_{in}}$	$H = \frac{C_{outpath}}{C_{inpath}}$
Branching-Effort	$b = C_{onpath}$ $+C_{offpath}/C_{onpath}$	$B = \pi b_i$
Effort [7] [9]	$f = gh$	$F = GBH$
Effort-Delay	f	$D_F = \sum g_i h_i$
Parasitic-Delay	p	$P = \sum p_i$
Delay [10]	$d = f + p = gh + p$	$D = \sum d_i = D_F + P$

in the current equations (5) and (6), governing the behavior of MOSFETs in their linear operating zones.

Thus, NMOS mobility μ_n is greater than PMOS mobility μ_p. In the linear zone, for N-channel MOSFET the current equation is

$$I_{DS} = \mu_n \cdot C_{ox} \cdot \frac{W_n}{L_n} \cdot [(V_{GS} - V_T) \cdot V_{DS} - \frac{V_{DS}^2}{2}] \quad (5)$$

For a P-channel MOSFET operating in the linear region, the governing equation is

$$I_{SD} = \mu_p \cdot C_{ox} \cdot \frac{W_p}{L_p} \cdot [(V_{GS} - V_T) \cdot V_{SD} - \frac{V_{SD}^2}{2}] \quad (6)$$

When comparing PMOS with NMOS, we often assume the same voltage and constant current source. Thus, if equations (5) and (6) of the current formulas are equivalent, C_{ox} and voltages that are common on both sides are eliminated. After canceling, we get the equation

$$\mu_p \cdot \frac{W_p}{L_p} = \mu_n \cdot \frac{W_n}{L_n} \quad (7)$$

Keeping the length in (7) constant is a necessary condition for any practical circuit. This yields

$$W_p = W_n \cdot \frac{\mu_n}{\mu_p} \quad (8)$$

W_p is bigger than W_n because the mobility of n-type electrons is higher than the mobility of p-type holes. PMOS is, therefore, always larger than NMOS.

The inverter, which is shown in Fig. 1(a), contains a 2W wide pull-up transistor and a pull-down transistor whose width is W, since PMOS is wider than NMOS. Therefore, the total input capacitance will be $W + 2W = 3W$. Logical effort of the inverter is $3W/3W = 1$. The pull-down transistors for the NAND gate in Fig. 1(b) are connected in series, giving each transistor a conductance that is double that of the pull-down transistor for the reference inverter. In order to achieve the same pull-down drive as the pull-up transistor inverter, the pull-up transistors would need to be as wide as the inverter. The aggregate input capacitance of each input signal is equivalent to the summation of the widths of the pull-down and pull-up transistor, i.e., $C_i = 2 + 2 = 4$. $C_{inv} = 1 + 2 = 3$ is the input capacitance of the inverter with equal output drive. The logical effort per input of the 2-input NAND gate is, thus, $g = \frac{4}{3}$.

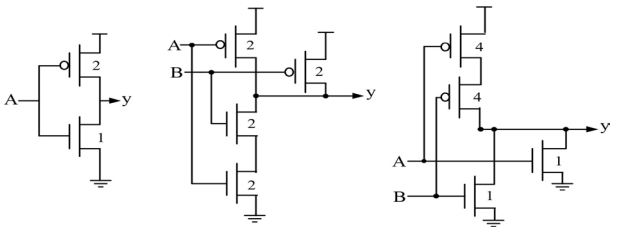

Fig. 1. Logic gates: (a) inverter, (b) two-input NAND, and (c) two-input NOR gate [2], [5].

Fig. 1(c) depicts the NOR gate in a comparable manner. In order to attain an equivalent pull-down drive to that of the inverter, the pull-down transistors must have a width of one unit. It is necessary to have four units between the pull-up transistors in a series configuration to achieve an equivalent pull-up drive as that of the inverter transistors. We can calculate the logical effort of the NOR gate as $g = \frac{5}{3}$ by adding up the input capacitance.

In the process of logic gate design, CMOS transistors are represented as ideal resistors in order to produce an output drive that is equivalent to that of the *Reference Inverter*. In the absence of transistor activation, the resistor exhibits zero conductance. However, when the transistor is activated, the conductance of the resistor is commensurate with its width. The evaluation of transistor conductance involves the amalgamation of the conductance of the transistor network, utilizing established principles for measuring the conductance of a resistor network that comprises parallel as well as series connections to the resistor. Whilst this model may be a rough estimation, it effectively characterizes the efficiency of gate logic, thereby, enabling the construction of expeditious structures.

The three low-power full adder circuits previously disclosed are compared to the new low-power single-bit full adder proposed in this paper [5]–[13]. The circuits are simulated using the Cadence Virtuoso tool and GPDK 45nm technology, and their performance is evaluated based on average power, delay, power delay product, and transistor count. The remaining sections are organized as follows.

In Section II, the design of a low-power full adder circuit and the computation of logical effort are presented. Simulations to determine logical effort and its calculation from a graph are shown in Section III. Section IV contains the results discussion. Finally, Section V presents the conclusion in its final position.

II. Design Of A Low-Power Full Adder And Calculation Of Logical Effort

A low-power full adder circuit is proposed and compared with the three published circuits. The proposed low power full adder circuit shown in Design-4 and the three published circuits are shown in Design-1, 2, and 3, which are from [5] to [14], respectively, and the logical effort values are given for the designs. Design-1 [5], which is shown in Fig. 2, is a hybrid full adder circuit designed by combining CMOS and pass-transistor logic styles. This design is divided into three modules: XOR, inverter, and multiplexer. The XOR module (based on pass transistor logic) generates intermediate signal $A \oplus B$ that is fed to an inverter, which gives $A \odot B$. These signals are afterwards used as selection lines in the multiplexer module [15], [6] (based on pass-transistor logic style) that includes two transmission gate based multiplexers to generate sum and carry. The logical effort in Fig. 2 has four stages, i.e., inverter (used for inputs A and B) for which we know $g = 1$, XOR which gives $g = \frac{2+1+2+1}{3} = 2$, inverter (used to get XNOR output) with $g = 1$, and multiplexer with $g = 2$ for both sum and carry, so that the g value will be the same.

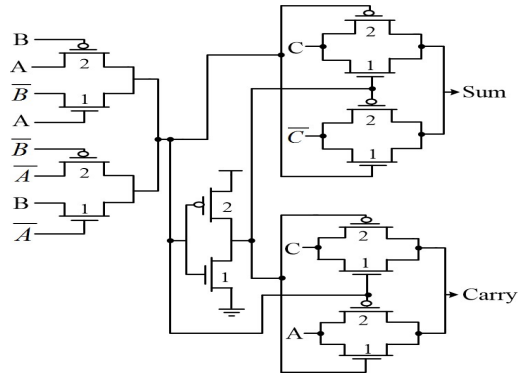

Fig. 2. Low-Power Hybrid Full Adder (Design-1 [5]).

From Fig. 2, it is clear that we need to multiply the values of logical effort for all four stages to get the final g value for sum and carry:

- Sum: $g = 1 \cdot 2 \cdot 1 \cdot 2 = 4$
- Carry: $g = 1 \cdot 2 \cdot 1 \cdot 2 = 4$

Design-2 [14], which is shown in Fig. 3, uses a modified XNOR circuit where the power consumption is reduced significantly by deliberate use of weak inverter and the same XNOR circuit is again used to produce sum. Then, a multiplexer with select lines of XOR and XNOR is used to produce carry. The circuit on Fig. 3 has 4 stages for both sum and carry, but for sum it has inverter (used for input B) with $g = 1$, XOR which gives $g = \frac{2+1+2+1}{3} = 2$, inverter (used to get XNOR output) with $g = 1$, and XOR (for generating sum), which gives $g = \frac{2+1+2+1}{3} = 2$. For carry, it has inverter (used for input B) for which we know that $g = 1$, XOR which gives $g = \frac{2+1+2+1}{3} = 2$, inverter (used to get XNOR output) with $g = 1$, and multiplexer with $g = 2$. Therefore, the final logical effort values for sum and carry for this design are:

- Sum: $g = 1 \cdot 2 \cdot 1 \cdot 2 = 4$
- Carry: $g = 1 \cdot 2 \cdot 1 \cdot 2 = 4$

Design-3 [13], which is shown in Fig. 4, is the same as Design-1, but the only difference is that it uses a 4T XOR logic, which has lesser power dissipation compared to other XOR gates, thus, reducing the total power dissipation compared to Design-1. The circuit on Fig. 4 has four stages

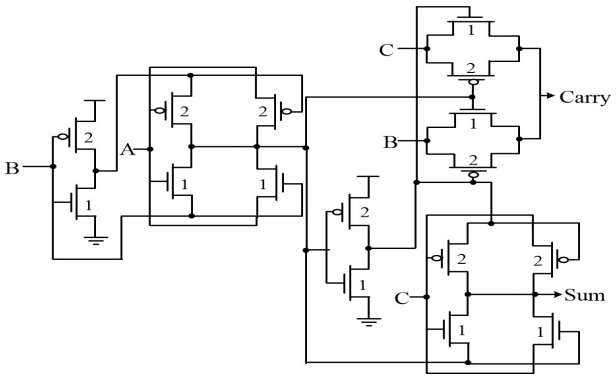

Fig. 3. Low-Power Hybrid Full Adder (Design-2 [14]).

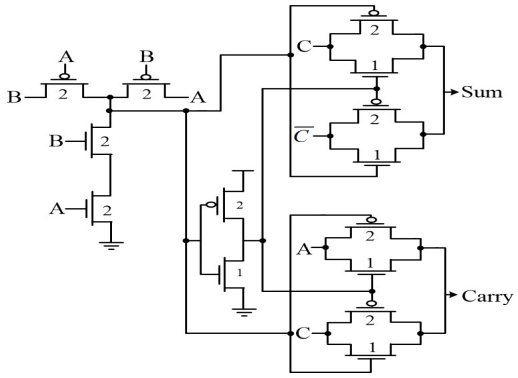

Fig. 4. Low-Power Hybrid Full Adder (Design-3 [13]).

for sum, i.e., XOR with $g = 4/3$, inverter (used for XNOR output) with $g = 1$, inverter (used for input C) with $g = 1$, and multiplexer with $g = 2$. Similarly, it has three stages for carry, i.e., XOR with $g = 4/3$, inverter (used for XNOR output) with $g = 1$, and multiplexer with $g = 2$. Therefore, the final logical effort values for sum and carry are:

- Sum: $g = 1.33 \cdot 1 \cdot 1 \cdot 2 = 2.66$
- Carry: $g = 1.33 \cdot 1 \cdot 2 = 2.66$

As mentioned above, we proposed a new circuit (Fig. 4) in order to get the best delay and much less power dissipation. This design used a 4T XOR gate, and the output of it is connected to an inverter to produce XNOR output. A multiplexer (based on pass transistor logic style) with XOR and XNOR as select lines are used produce carry output, and an XNOR module from Design-3 is used to produce output of sum. This design has three stages for both sum and carry, but sum has XOR which gives $g = \frac{4}{3} = 1.33$, inverter (used to get XNOR output) with $g = 1$, and XOR (for generating sum) which gives $g = \frac{(2+1+2+1)}{3} = 2$. For carry, it has XOR which gives $g = \frac{4}{3} = 1.33$, inverter (used to get XNOR output) with $g = 1$, and multiplexer with $g = 2$. So the final logical effort values for sum and carry for this design are:

- Sum: $g = 1.33 \cdot 1 \cdot 2 = 2.66$
- Carry: $g = 1.33 \cdot 1 \cdot 2 = 2.66$

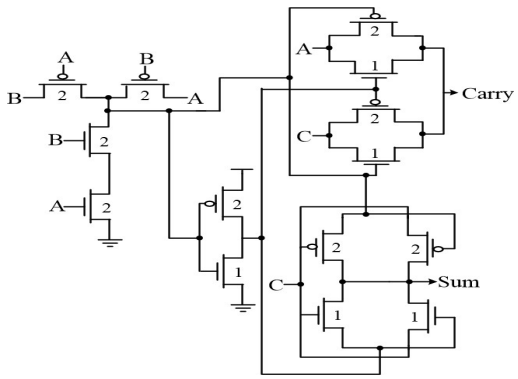

Fig. 5. Proposed Low-Power Hybrid Full Adder (Design-4).

All four designs are simulated using Cadence Virtuoso tool, and their performance is analyzed in terms of power dissipation and delay. The delay of sum and carry for every change

in output (i.e., $0 \rightarrow 1$ and $1 \rightarrow 0$ transitions), average power, power delay product of the sum and carry, and transistor count of all four full adder circuits before resizing (i.e., by taking the default width which is $W = 120nm$ for both NMOS and PMOS), as well as after resizing (i.e., by taking the width $W = 120nm$, where $g = 1$ and $W = 240nm$ with $g = 2$ for both NMOS and PMOS) are listed in Table II. From Table II, it can be observed that the average power, delay, power delay product of sum and carry, and transistor count is minimum for Design-4 before resizing.

Now when, according to logical effort, the full adders are resized, we can see that average power is minimum for the proposed design compared to other three designs, but is higher than the values shown before resizing. This is because, after resizing, the width is increased and so are R and C values. This leads to the power increase.

III. Simulation To Identify Logical Effort And Calculation From Graph

In this investigation, sum and carry for all four architectures are separated and simulated using the 45nm Cadence Virtuoso tool. Initially, the delay is determined from the Cadence Virtuoso tool simulated graph for each circuit and fan-out setting. All the circuits are constructed with the default PMOS and NMOS widths ($W = 120nm$ and $L = 45nm$, respectively). Fan-out from 0 to 4 is taken and the absolute delay for each circuit is determined. Absolute delay is the ratio of any circuit's real delay to the inverter fan-out-0 delay. Table III lists the delay values for logic gates such as inverter, NAND, and NOR (before/after resizing) and for the four designs.

In Table III, the delay values of all circuits are given for fan-out from 0 to 4. From these, we need to calculate absolute delay for all the circuits. The absolute delay values for fan-out 0-4 are listed in Table IV. Now the graph is plotted for each circuit separately. Fan-out from 0 to 4 is plotted in X-axis, and absolute delay values in ps are plotted in Y-axis. The slope is calculated for sum and carry separately in the graph.

Now, if we compare equation (5) with the straight-line equation, which is $y = mx + c$, we can see that absolute delay (d) is plotted in Y-axis, and the electrical effort (h) called fan-out is plotted in X-axis, and the slope (m) gives the logical effort (g) value. The graphs for logic gates and all four designs, including straight-line equations for sum and carry separately, are shown in Figs. 6 and 7. Here, Fig. 6(a) shows the graph of Inverter, NAND and NOR, Fig. 6(b) shows the graph of Design-1, Fig. 6(c) shows the graph of Design-2, Fig. 6(d) shows the graph of Design-3, and Fig. 7 shows the graph of Design-4. As we said, slope value in the straight line equation gives the logical effort value, and the slope is calculated by simulating all the circuits for different fan-out conditions. These slope values are stated as simulated logical effort values. The simulated logical effort values are stated in Table V by comparing them with the calculated logical effort values.

As previously stated, the logical effort value is determined by the slope, and this information is recorded in Table V through comparison with the corresponding calculated values.

In Table V, we can see that the simulated and calculated values have a lot of variations. So, according to the logical effort, the transistors are resized. As the default width for both NMOS and PMOS in 45nm technology is 120nm, the width of PMOS and NMOS are resized according to logical effort values, i.e., width of PMOS is taken as $W = 240nm$ where $g = 2$, and $W = 480nm$ where $g = 4$ the same is done for NMOS. Now, we have to follow the same procedure: the first delay is noted from simulated graph, which is shown in Table V, and absolute delay values are shown in Table IV.

Now the graphs are plotted for all the circuits, which are shown in Figs. 8 and 9. Here, Fig. 8(a) shows the graph of Inverter, NAND and NOR, Fig. 8(b) shows the graph of Design-1, Fig. 8(c) shows the graph of Design-2, Fig. 8(d) shows the graph of Design-3, and Fig. 9 shows the graph of Design-4.

The presented graphs indicate that the slope values have been identified as simulated logical effort values after resizing in Table V in comparison with calculated values of logical effort.

From Table V we can see that the simulated values before resizing have a lot of variations compared to the values after resizing. After resizing the gates, we can see that the calculated and simulated values are almost getting the same.

We have conducted a comparative analysis of the logical effort values and delay times of the fundamental gates after resizing the transistor sizes, as demonstrated in Tables VI and VII in reference to the existing work [16]–[18]. In reference to [17], the author did not provide separate calculations for the PDP and power of the 2-input NAND, NOR, and inverter. Instead, they focused on deriving expressions for the proposed Equal Effect Delay and Voltage Scaling (EEDVS) technique's power, delay, and PDP in comparison to other technologies, specifically UMC 130nm and P45-nm. It is noteworthy that within this technique, the supply voltage has been scaled in accordance with the employed logical effort stages. It is important to note that due to disparities in the underlying technologies utilized in these designs, a direct one-to-one comparison may not be straightforward. Consequently, we have presented the values as reported in the literature. Given that these are fundamental gates, any disparities arising from technological differences are anticipated to be relatively minor, thus preserving the validity of our comparative assessment.

IV. Results Analysis and Discussion

In our work, we utilized the Cadence GPDK 45nm technology node, a widely accepted and extensively verified standard for semiconductor process technology. Its well-established status ensures a high level of confidence in the accuracy and reliability of the provided model libraries. The model libraries associated with the GPDK 45nm technology node have undergone rigorous testing and verification procedures. This extensive validation process covers a wide range of operating conditions and scenarios, bolstering the credibility of our simulations. In our study, we decided to employ the GPDK 45nm technology, primarily due to its wide acceptance and robust design ecosystem, particularly within academic settings.

TABLE II
COMPARISON OF DELAY, AVG. POWER, PDP (POWER DELAY PRODUCT) AND TRANSISTOR COUNT OF ALL FOUR DESIGNS BEFORE/AFTER RESIZING

	Design-1 [5]		**Design-2 [14]**		**Design-3 [13]**		**Design-4 [proposed]**	
	Before	After	Before	After	Before	After	Before	After
Delay of Sum(ns)								
001-(0 $\rightarrow$ 1)	0.039	0.038	0.043	0.044	0.034	0.034	0.033	0.029
011-(1 $\rightarrow$ 0)	0.045	0.056	0.023	0.024	0.050	0.046	0.053	0.047
100-(0 $\rightarrow$ 1)	0.005	0.005	0.006	0.005	0.005	0.004	0.004	0.004
101-(1 $\rightarrow$ 0)	0.029	0.030	0.032	0.044	0.028	0.030	0.040	0.048
111-(0 $\rightarrow$ 1)	0.035	0.043	0.032	0.035	0.040	0.035	0.041	0.037
Delay of Carry(ns)								
011-(0 $\rightarrow$ 1)	0.0344	0.0419	0.0326	0.0340	0.0394	0.0343	0.0461	0.040
100-(1 $\rightarrow$ 0)	0.0049	0.0053	0.0114	0.0159	0.0048	0.0050	0.0048	0.005
101-(0 $\rightarrow$ 1)	0.0061	0.0062	0.0272	0.0308	0.0052	0.0050	0.0054	0.005
Average Power(nW)	75.97	112.5	55.36	82.37	50.42	77.71	47.61	73.32
PDP of Sum ($\times 10^{-18} J$)	0.3798	0.54	0.3266	0.4118	0.2470	0.3496	0.2094	0.3010
PDP of Carry ($\times 10^{-18} J$)	0.3722	0.5962	0.6311	1.3096	0.2420	0.3885	0.2285	0.3812
Transistor Count	20	20	16	16	16	16	14	14

TABLE III
DELAY OF CIRCUITS (PS) FOR FAN-OUT 0-4 BEFORE /AFTER RESIZING

h	**0**		**1**		**2**		**3**		4	
	Before	After	Before	After	Before	After	Before	After	Before	After
Inverter	6.5	6.9	15.8	16.4	19.9	20.6	23.8	24.7	27.7	28.7
NAND	12.6	12.6	30.4	30.9	40.6	41.8	50.5	52.2	60.2	62.5
NOR	5.7	7.2	32.3	25.4	38.7	32.6	45.4	39.5	52.2	46.2
Design-1 SUM [5]	11.1	11.6	63	63.9	94.4	91.3	123.2	117.7	155.4	141.8
Design-1 CARRY [5]	10.9	7.7	62.2	49.9	90.7	73	117.7	94.6	143.6	116.4
Design-2 SUM [14]	11.4	7.8	55.3	53.5	74.4	78.1	91	102	109.5	125.3
Design-2 CARRY [14]	14.8	13.2	65.7	72	87.6	95.9	103.2	127	122.3	151.2
Design-3 SUM [13]	9.2	10.2	42.2	42.8	60	60	77	76.6	92.9	92.7
Design-3 CARRY [13]	9.4	10.3	42.5	42.2	59.4	58.5	75.8	74	91.1	89
Design-4 SUM [Proposed]	7.8	9.2	36.4	38.3	48.7	51.9	63.7	65.2	74.9	78
Design-4 CARRY [Proposed]	9.4	10.3	42.5	422	59.4	58.5	75.8	74	91.1	89

Transistor resizing plays a crucial role in optimizing the delay of the proposed hybrid full adder circuit within the context of the chosen technology.

- By resizing transistors, we effectively alter their width-to-length ratios. This adjustment directly impacts the intrinsic capacitance of the transistors. Specifically, increasing the width relative to the length can result in a reduction of intrinsic capacitance, which, in turn, accelerates the charging and discharging processes during signal propagation.
- Transistor resizing allows for a careful balance between driving strength and capacitive load.
- In advanced technology nodes, interconnect resistance (R) and capacitance (C) effects can significantly contribute to signal delays. Proper transistor resizing helps to mitigate these RC delays by adjusting the driving strength to adequately overcome the capacitive load while minimizing resistance-induced delays.
- It's important to note that while transistor resizing can improve delay, there is typically a trade-off with power consumption. Over-sizing transistors may lead to higher static power dissipation, necessitating a careful optimization to achieve a balance between performance and power efficiency.
- As technology nodes continue to scale down, transistor dimensions shrink, leading to higher channel resistance. Transistor resizing allows for the adjustment of channel dimensions to maintain optimal performance in the face of these scaling trends.

Table II presents the outcomes acquired prior to and subsequent to the alteration in size of the transistors. The Full-Adder Design-4, as proposed, has demonstrated a minimal power dissipation of 47.61nW prior to resizing. Upon resizing, the power dissipation increased to 73.32nW. Notably, Design-4 employs the fewest number of transistors, totaling 14, in comparison to the other three designs.

Table III gives the delay values before and after resizing the transistors for fan-out 0 to 4 of logic gates like inverter, NAND, NOR and sum, carry of four designs. We can observe that as the fan-out is increasing from 0 to 4, the delay values are also getting increased. Even the absolute delay values before and after resizing the transistors are given in Table IV

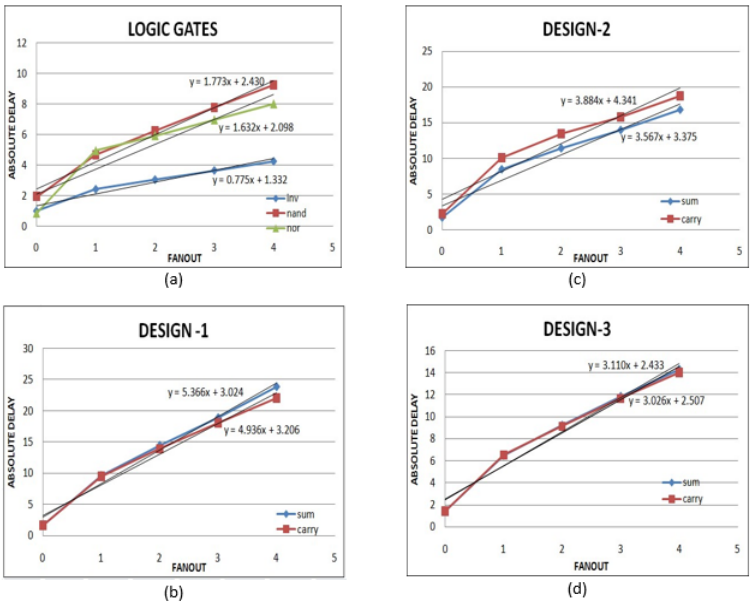

Fig. 6. (a) Fan-out plot vs. Absolute Delay of logic gates; (b) Fan-out plot vs. Absolute Delay of Design-1 [3]; (c) Fan-out plot vs. Absolute Delay of Design-2 [4]; (d) Fan-out plot vs. Absolute Delay of Design-3 [5].

TABLE IV
ABSOLUTE DELAY OF CIRCUITS (PS) FOR FAN-OUT 0-4 BEFORE/AFTER RESIZING

h	0		1		2		3		4	
	Before	After	Before	After	Before	After	Before	After	Before	After
Inverter	1	1	2.43	2.37	3.06	2.98	3.66	3.57	4.26	4.15
NAND	1.93	1.82	4.67	4.47	6.24	6.05	7.76	7.56	9.26	9.05
NOR	0.87	1.04	4.96	3.68	5.95	4.72	6.98	5.72	8.03	6.69
Design-1 SUM [5]	1.70	1.68	9.69	9.26	14.52	13.23	18.95	17.05	23.90	20.55
Design-1 CARRY [5]	1.67	1.11	9.56	7.23	13.95	10.57	18.10	13.71	22.09	16.68
Design-2 SUM [14]	1.75	1.13	8.50	7.75	11.44	11.31	14	14.78	16.84	18.15
Design-2 CARRY [14]	2.27	1.91	10.10	10.43	13.47	13.89	15.87	18.40	18.81	21.91
Design-3 SUM [13]	1.41	1.47	6.49	6.20	9.23	8.69	11.84	11.10	14.29	13.43
Design-3 CARRY [13]	1.44	1.49	6.53	6.11	9.13	8.47	11.66	10.72	14.01	12.89
Design-4 SUM [Proposed]	1.20	1.33	5.60	5.55	7.49	7.52	9.80	9.44	11.52	11.30
Design-4 CARRY [Proposed]	1.44	1.49	6.53	6.11	9.13	8.47	11.66	10.72	14.01	12.89

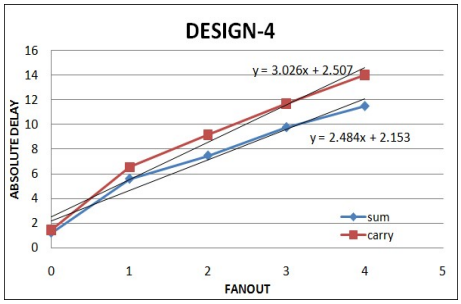

Fig. 7. Fan-out plot vs. Absolute Delay of Design-4 [Proposed].

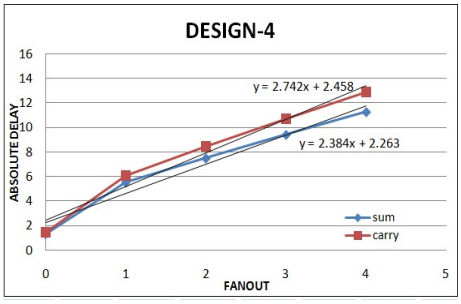

Fig. 8. Fan-out plot vs. Absolute Delay of Design-4 [Proposed].

for fan-out 0 to 4 and the graphs for fan-out vs. absolute delay is plotted for the logic gates, and four full adder designs which are shown in Figs. 6 and 7, and for each graph the line equations are shown. The slope values are noted as simulated logical effort values. Table V gives the comparison of calculated and simulated logical effort values before and after resizing. We can see that there is not much variation between calculated and simulated values of logical effort.

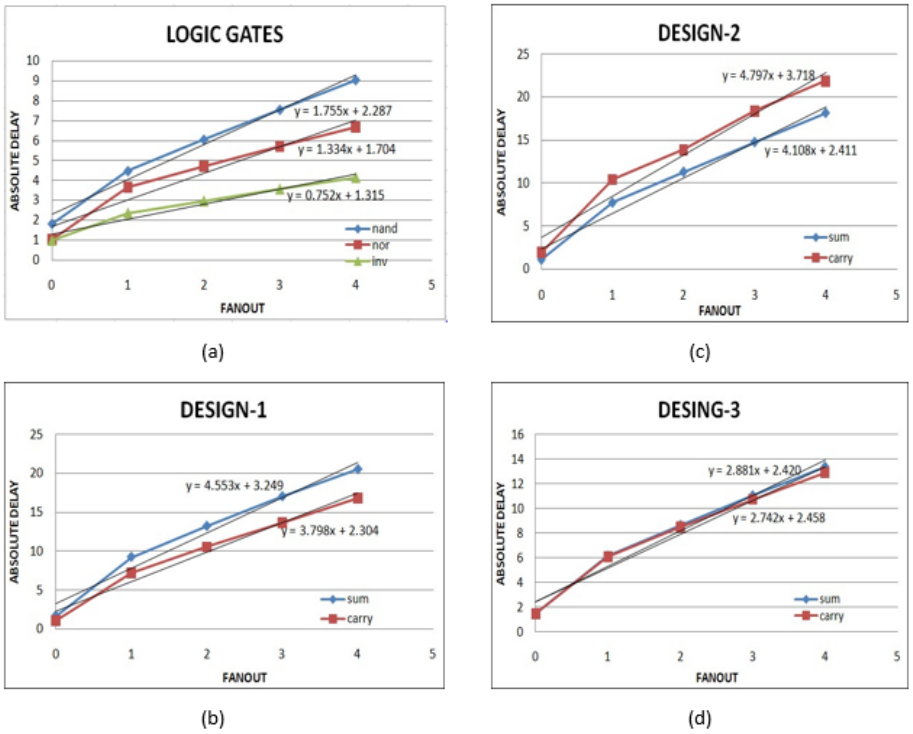

Fig. 9. (a) Fan-out plot vs. Absolute Delay of logic gates; (b) Fan-out plot vs. Absolute Delay of Design-1 [3]; (c) Fan-out plot vs. Absolute Delay of Design-2 [4]; (d) Fan-out plot vs. Absolute Delay of Design-3 [5].

TABLE V
COMPARISON OF CALCULATED AND SIMULATED LOGICAL EFFORT VALUES BEFORE/AFTER RESIZING

Circuit	Calculated Values		Simulated Values (resizing)	
	Before	After	Before	After
Inverter	1	1	0.775	0.752
NAND	1.33	1.33	1.773	1.755
NOR	1.66	1.66	1.632	1.334
Design-1 SUM [5]	4	4	5.366	4.553
Design-1 CARRY [5]	4	4	4.936	3.789
Design-2 SUM [14]	4	4	3.567	4.108
Design-2 CARRY [14]	4	4	3.884	4.797
Design-3 SUM [13]	2.66	2.66	3.110	2.881
Design-3 CARRY [13]	2.66	2.66	3.026	2.742
Design-4 SUM [Proposed]	2.66	2.66	2.484	2.384
Design-4 CARRY [Proposed]	2.66	2.66	3.026	2.742

TABLE VI
COMPARISON OF LOGICAL EFFORT

Logical effort					
	Proposed	[16]	[17]		[18]
Technology	GPDK 45nm	UMC 90nm	UMC 130nm	P45	UMC 90nm
Inverter	0.752	0.9714	0.95	1.57	0.9714
NAND	1.755	1.242	1.2	2.1	1.242
NOR	1.334	1.294	1.6	2.6	1.294

TABLE VII
COMPARISON OF DELAY, AVERAGE POWER, AND PDP

	Delay(pS)					Avg.Power(nW)	PDP
	Proposed	[16]	[17]		[18]	Proposed	Proposed nS x nW
Technology	GPDK 45-nm	UMC 90nm	UMC 130nm	P45	UMC 90nm	GPDK 45-nm	GPDK 45-nm
Inverter	4.15	11.81	21.6	40.2	12.42	15.49	0.064
NAND	9.05	-	30.1	56.4	15.79	12.89	0.116
NOR	6.69	-	35.5	66.0	18.39	23.38	0.156
SUM	26.53nS	-	-	-	-	135.33	3590.30
CARRY	38.06nS	-	-	-	-	130.86	4980.53

The graphs are shown in Figs. 8 and 9. The slope values are noted as simulated logical effort values. We can see that the calculated and simulated values of logical effort after resizing the transistors are almost getting matched.

A. Scaling the supply voltage

To ensure a fair comparison, all the existing and proposed (present work) circuits are designed using Generic Process Design Kit (GPDK) with 45 nm CMOS technology and simulated in Cadence Virtuoso using the same test inputs. The ability of design to operate in a wide variety of supply voltages, from 0.8 V to 1.1 V at typical corners (TT) at 27°C, was investigated.The Fig. 10(c) shows all the results. It has been seen that the proposed design (design 4) is showing better performance in terms of average power, delay of sum and delay of carry than other designs by 34.23%, 26.81% and 4.33%, respectively.

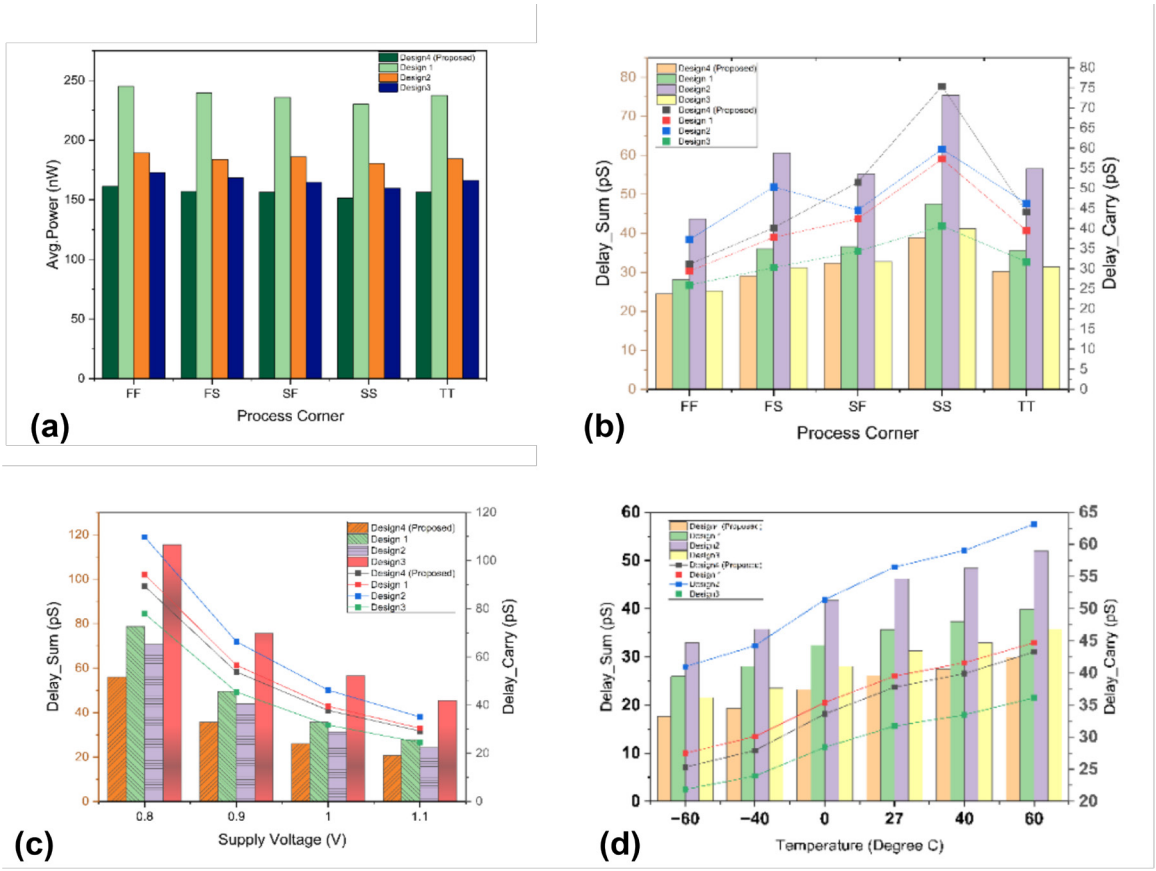

Fig. 10. PVT analysis graphs: (a) average power variation with process corner for all the compared designs, (b) Sum and Carry delay variation with process corner for all the compared designs, (c) sum and carry delay variation with different supply voltage for all the compared designs, and (d) sum and carry delay variation with different temperature for all the compared designs.

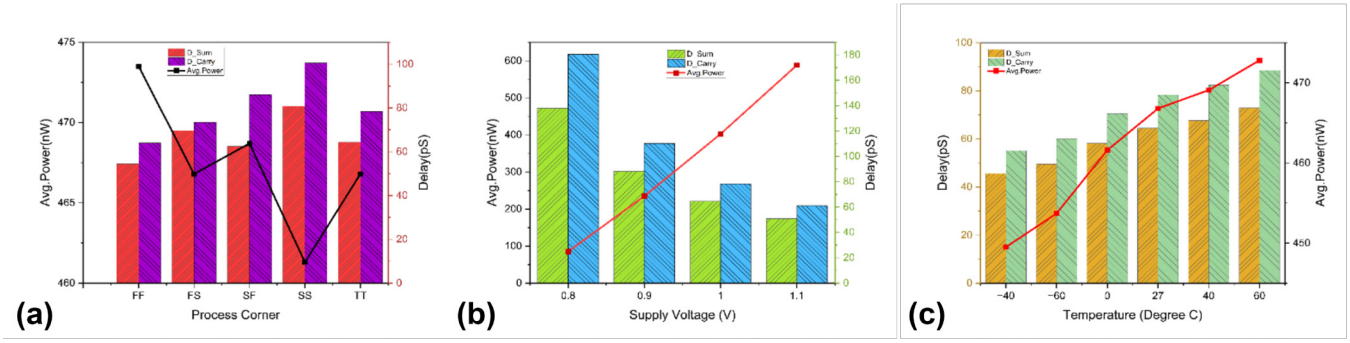

Fig. 11. Post Layout simulation result: (a) average power and Delay of Sum and Carry with variation of process corners, (b) average power and Delay of Sum and Carry with variation of supply voltage, and (c) average power and Delay of Sum and Carry with variation of temperature.

B. Variation in Temperature

The compared designs was also investigated at various temperatures. In Fig. 10(d), the plot of delay for sum and carry over a range of -60°C to +60°C is depicted. Although the power seems to increase with temperature, delay does not increase that much, and hence the PDP is seen to be improved monotonically.

TABLE VIII
COMPARATIVE RESULTS OF POST-LAYOUT MONTE CARLO SIMULATION FOR 1000 ITERATIONS

Design	Delay				SER (%)
	Min (ns)	Max (ns)	Mean (ns)	Std.Dev (ps)	
Design-4 SUM(Proposed)	55.94	83.13	65.20	3.795	0
Design-4 CARRY (Proposed)	67.01	96.77	78.88	4.296	0
Area	l= 3.945 μ m , w= 3.18 μ m,A=12.54 μ m ^{2}				

TABLE IX
COMPARATIVE RESULTS OF MONTE CARLO SIMULATION FOR 1000 ITERATIONS

Design	Delay				SER (%)
	Min (ns)	Max (ns)	Mean (ns)	Std. Dev (ps)	
Design-1 SUM [5]	31.37	43.11	35.97	1.758	0
Design-1 CARRY [5]	33.32	48.99	42.63	1.66	0
Dsign-2 SUM [14]	34.47	50.37	42.14	1.949	0
Dsign-2 CARRY [14]	28.97	38.12	33.02	1.568	0
Dsign-3 SUM [13]	27.94	36.53	31.61	1.412	0
Dsign-3 CARRY [13]	28.37	37.31	32.06	1.467	0
Design-4 SUM (Proposed)	21.86	34.22	26.53	1.76	0
Design-4 CARRY (Proposed)	32.37	45.75	38.06	1.991	0

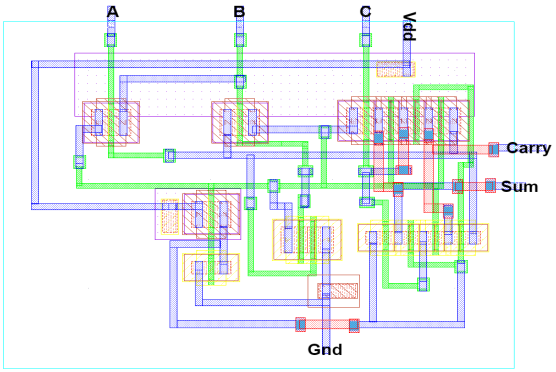

Fig. 12. Layout of proposed full adder circuit [Design-4].

C. Impact of Process Corner Variation

The designs are realized in nanoscale CMOS technology, so it is necessary to check for deviations at process corners considering typical performance metrics. All the designs are analyzed under different process corners and Fig. 10 (a) and (b) shows the result of all the analysis.

D. Verification of Stability using the Monte-Carlo Method

The functional stability of the suggested design is additionally validated through the use of a number of different mismatch and process combinations tested with the Monte Carlo sampling method across 1,000 iterations. The result of the simulation is tabulated in the Table IX.

E. Post-layout Simulation Result

The layout of the proposed design is shown in the Fig. 12 the area and post layout Monte Carlo simulation results are tabulated in the Table VIII. Fig. 11 depicts the post layout PVT simulation results. It can be seen clearly from the result that the proposed design shows better performance than existing full adder designs.

V. CONCLUSION

A low power high-speed hybrid full adder circuit is proposed in this paper and compared this circuit with three existing low power full adder circuits. The simulation was performed in cadence virtuoso tool with gpdk 45nm CMOS technology. The performance of these full adders are analyzed in terms of delay, average power, power delay product, and transistor count. The logical effort method is used to calculate the delay. Simulation is performed with fan-out values from 0 to 4 to find the logical effort required. The transistors are resized to get the best delay according to the logical effort, and comparisons are done.

REFERENCES

[1] T. Nikoubin, "Hybrid logical effort for hybrid logic style full adders in multistage structures," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 27, pp. 1138–1147, May 2019.
[2] R. Uma, P. Dhavachelvan, "Logic optimization using technology independent mux based adders in fpga," *International Journal of VLSI design & Communication Systems*, vol. 3, no. 4, pp. 135–149, Aug. 2012.
[3] M. J. A. Morad, S. R. Talebiyan, E. Pakniyat, "Design of new lowpower high-performance full adder with new xor-xnor circuit," *International Congress on Technology, Communication and Knowledge (ICTCK) IEEE*, pp. 153–158, Nov. 2015.
[4] J. Ajayan, D. Nirmal, S. Sivasankari, D. Sivaranjani, M. Manikandan, "High speed low power full adder circuit design using current comparison based domino," *2nd Intl. Conference on Devices, Circuits and Systems (ICDCS), IEEE*, pp. 1–5, 2014.
[5] M. Mewada, M. Zaveri, "A low-power high-speed hybrid full adder," *20th International Symposium on VLSI Design and Test (VDAT), IEEE*, pp. 1–2, 2016.
[6] P. Kumar, R. K. Sharma, "Low voltage high performance hybrid full adder," *Engineering Science and Technology, an International Journal*, vol. 19 , pp. 559-565, Mar. 2016.
[7] I. Sutherland, R. F. Sproull, B. Sproull, D. Harris, *Logical effort: designing fast CMOS circuits, Morgan Kaufmann*, 1999.
[8] C. C. Wang, D. Markovic, "Delay estimation and sizing of cmos logic using logical effort with slope correction," *IEEE Trans. on Circuits and Systems II: Express Briefs*, vol. 56, pp. 634–638, Aug. 2009.
[9] J. Keane, H. Eom, T. H. Kim, S. Sapatnekar,C Kim,"Subthreshold logical effort: a systematic framework for optimal subthreshold device sizing," *Proceedings of the 43rd Annual Design Automation Conference*, pp. 425-428, 2006.
[10] S. Maheshwari, J. Patel, S. K. Nirmalkar, A. Gupta,"Logical effort based power-delay-product optimization," *International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE*, pp. 565–569, 2014.
[11] R. M. Anacan, J. L. Bagay, "Logical effort analysis of various vlsi design algorithms," *IEEE International Conference on Control System, Computing and Engineering (ICCSCE), IEEE*, pp. 19–23, 2015.
[12] R. Uma, P. Dhavachelvan, "Performance evaluation of full adders in asic using logical effort calculation, "*International Conference on Recent Trends in Information Technology (ICRTIT), IEEE*, pp. 612–618, 2013.
[13] M. Devadas, K. L. Kishore, "Design topologies for low power cmos full adder," *Intl. Conference on Inventive Systems and Control (ICISC), IEEE*, pp. 1–4, 2017.
[14] P. Bhattacharyya, B. Kundu, S. Ghosh, V. Kumar, A. Dandapat, "Performance analysis of a low-power high-speed hybrid 1-bit full adder circuit," *IEEE Trans. on very large scale integration (VLSI) systems*, vol. 23, pp. 2001–2008, 2014.
[15] M. Shoba, R. Nakkeeran, "Gdi based full adders for energy efficient arithmetic applications," *Engineering Science and Technology, an International Journal*, vol. 19, pp. 485–496, 2016.
[16] S. Maheshwari, R. Raza, P. Kumar, A. Gupta, "Convex Optimization of Energy and Delay Using Logical Effort Method in Deep Sub-micron Technology," *17th International Symposium, VDAT 2013*, pp. 185–193, Springer Berlin Heidelberg, 2013.
[17] A. Kabbani, "Logical effort based dynamic power estimation and optimization of static CMOS circuits," *Integration*, vol. 43, no. 3, pp.279–288, 2010.
[18] S. Maheshwari, H. S. Raghav, A. Gupta, "Characterization of Logical Effort for Improved Delay," *VLSI Design and Test: 17th International Symposium, VDAT 2013*, pp. 108–117, Springer Berlin Heidelberg, 2013.

A Novel Separable Convolution Architecture for Image Processing Applications

Jaiza Hassan and Burhan Khurshid

***Abstract*—Two-dimensional (2D) Convolution is frequently used in many image processing applications like image smoothening, image sharpening, feature extraction, image enhancement, object recognition, etc. Although the operation of 2D Convolution is simple, its hardware implementation is quite challenging due to enormous computational and memory costs. Various 2D convolution implementations have been proposed in the literature. Among them Separable architectures stand out in terms of the reduction in the computational complexity. However, these have not been extensively explored in the literature, and more research needs to be done, especially for applications that are constrained in resources. This paper presents a novel separable Convolution architecture based on the folding transformation. The application of the folding transformation technique yields a resource-efficient architecture by time multiplexing the different functional units within the separable Convolution operation. The hardware implementation is done using Xilinx Artix-7 xc7a35tcpg236-1 FPGA device. The proposed architecture offers benefits over the existing architectures regarding on-chip resource utilization, power consumption, critical path delay, and external memory bandwidth (EMB).**

***Index Terms*—2D convolution, Folding transformation, FPGA, Partial buffering scheme, Separable filters.**

Original Research Paper
DOI: 10.53314/ELS2428012H

I. Introduction

In many image-processing applications, 2D Convolution is one of the basic operations that is frequently used. Discrete wavelet transform, object recognition, restoration, template matching, feature extraction, denoising, edge detection, enhancement, high-pass filtering (sharpening), low-pass filtering (blurring), and other image processing applications all use convolution operation [1-5]. The implementation of 2D convolution is complex, even with its straightforward form, since it incurs high computational costs and is also memory-expensive. Calculating a single output pixel using an $n \times n$ kernel necessitates n^2 image pixels per clock cycle, n^2 multiplications, and n^2-1 additions. It is evident that as the kernel size increases, so do these challenges.

Numerous architectural approaches have been suggested in the literature to overcome the aforementioned issues. These architectures primarily focus on optimizing the convolution kernel module and designing an efficient buffering scheme. Reducing on-chip resources based on a suitable EMB is achieved by using an effective buffering scheme. Full Buffering (FB) and Single Window Partial Buffering (SWPB) schemes are proposed in [6]. FB scheme stores entire *n-1* raster rows of an input image for a kernel of size $n \times n$ in row buffers, thereby achieving an EMB of one pixel/cycle but requiring a large amount of on-chip resources. SWPB scheme reduces resource utilization by storing only a small portion of the image at the cost of increased bandwidth. Multi-Window Partial Buffering (MWPB) scheme addresses these issues by reusing shared data between windows, minimizing external memory bandwidth, and reducing hardware utilization [7]. Nevertheless, this buffering strategy might not be feasible for larger kernel sizes if bandwidth is limited [8].

To optimize the Convolution unit, several approaches have been suggested in the literature. These include time-sharing [9-10], multiplier-less [5, 11-12], decomposition of the kernel into smaller parts [12-18], using structural properties of the kernel [19], pipelining [20], reconfigurable architectures [1, 21-23], and polyphase decomposition [14, 24]. Among these approaches, spatial separable convolution is one effective method for minimizing hardware utilization [12-13, 17-18, 21, 25-27], which transforms 2D convolution into the product of two one-dimensional (1D) convolutions. Separable convolution yields significant computational resource savings as the kernel size increases. Further, most 2D kernels are strictly separable; thus, convolution kernel separability is a considerable topic. However, limited study has been done on the architectural optimization of spatial separable convolutions.

Distributed arithmetic (DA) based separable Convolution FPGA architectures viz. Multi-Window Separable Convolution (MWSC) architecture and Single Window Separable Convolution (SWSC) architecture are presented in [12]. Compared to MWSC architecture, SWSC architecture offers better processing time and resource usage at the cost of throughput. In [5], two architectures viz. Separable Filter Architecture (SFA) and Generalized Filter Architecture (GFA) based on DA are proposed and compared. Non-pipelined separable Convolution architecture based on the MWPB scheme (NPSCA) is proposed in [25]. The architecture reuses the same multipliers and adders for horizontal and vertical convolution. In [27], resched-

Manuscript received on May 3rd, 2024. Received in revised form on July 3rd, 2024. Accepted for publication on July 4th, 2024.

Jaiza Hassan is a PhD student in the Department of Electronics and Communication Engineering, National Institute of Technology, Srinagar, India (e-mail: jaiza_2021phaece006@nitsri.ac.in).

Burhan Khurshid is an Assistant Professor in the Department of Electronics and Communication Engineering, National Institute of Technology, Srinagar, India (e-mail: burhan@nitsri.ac.in).

uled separable convolution architecture (RSCA) is proposed in which only one multiplier is used sequentially in the vertical convolution, reducing the overall multiplier complexity.

The abovementioned methods presented in the literature minimize resource utilization, which can be further reduced by sacrificing the throughput by a small amount. Further, with the increase in kernel size, the computational complexity and the critical path of the existing designs increase. The aim is, thus, to design a resource-efficient separable convolution architecture wherein the computational complexity and critical path are independent of kernel size. This is achieved by appropriately time-multiplexing the functional units within the separable convolution algorithm using the folding transformation [28, 29]. The folding technique is a systematic approach to designing control circuits in digital signal processing (DSP) architectures that reduce the number of functional units. Further, as mentioned earlier, many buffering schemes have been proposed to address the EMB issues. The PB scheme [6] is the most resource-efficient of the existing buffering schemes [7, 15, 25]. As designing the resource-efficient convolution architecture is the primary goal of the study, separable convolution architecture based on the PB scheme (SCA) [25] is used as a base architecture for folding transformation.

The rest of the paper is organized as follows. Section II illustrates separable convolution architecture, and Section III presents proposed area-efficient architecture. Section IV discusses hardware and timing complexities, and Section V discusses the performance analysis. Finally, the paper concludes the work in Section VI.

II. Separable Convolution

The concept behind separable convolution is that a 2D separable $n \times n$ kernel can be spatially divided into two 1D kernels: an $n \times 1$ vertical kernel v and a $1 \times n$ horizontal kernel h. Thus, just $2n\text{-}2$ additions and $2n$ multiplications are needed per pixel. The architecture of the separable convolution based on the PB scheme [25] is presented in Figure 1. For an $n \times n$ kernel, n input pixels are accessed simultaneously from the memory into n buffers. Based on whether vertical convolution is performed first or horizontal convolution, the n pixels belong to either $n \times 1$ or $1 \times n$ window. In this study, vertical convolution is performed first, followed by horizontal convolution; however, horizontal followed by vertical can be performed equivalently. The n pixels in the n buffers belonging to the $n \times 1$ window are multiplied by the corresponding vertical coefficients. The results are then added and stored in the registers. Once n intermediate results are available, they are multiplied by the n corresponding horizontal coefficients. The products are then added, and the final result is obtained. In the same manner, all output pixels are computed. The study aims to present the resource-efficient architecture obtained by applying the folding transformation [28] to the above-discussed architecture, which will be discussed in the next section.

III. Proposed Folded Separable Convolution Architecture

Figure 2 presents the detailed schematic of the proposed Folded Separable Convolution Architecture (FSCA). The proposed architecture is obtained by time-multiplexing $2n$ multipliers and $2n\text{-}2$ adders in the separable convolution architecture by two pipelined multipliers (M_V and M_H) and two pipelined adders (A_V and A_H). In other words, separable convolution architecture is folded by folding factor n for kernel of size $n \times n$. Multiplier M_V and adder A_V perform vertical convolution, and multiplier M_H and adder A_H perform horizontal convolution. Two pipelined stages are used for multiplication and one pipelined stage for addition. The decision is based on the latency configurations of the Xilinx IP adders and multipliers in Artix-7 FPGAs that implement these functional units. Instead of n input pixels, only one pixel located in the $n \times 1$ window is accessed from the external memory at a time. This reduces the EMB to *one pixel per clock cycle.*

At every clock cycle, the input pixel is accessed from memory, and the appropriate vertical coefficient is assigned to the multiplier M_V. The product outputs are stored in the registers. A total of $n\text{-}2$ registers are used between the multiplier M_V and ad-

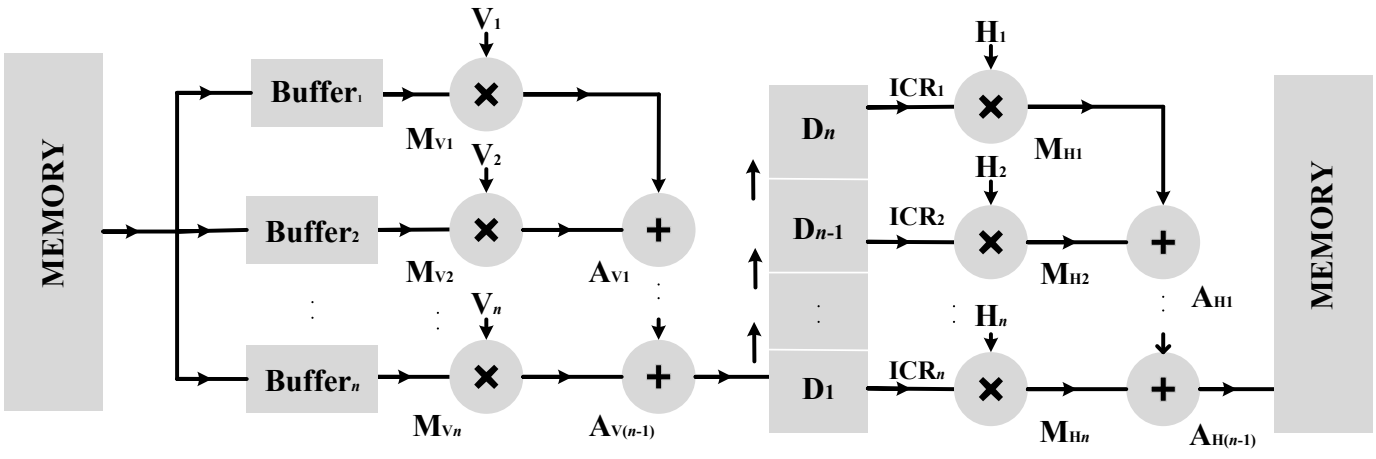

Fig. 1. Separable Convolution architecture based on PB scheme [25].

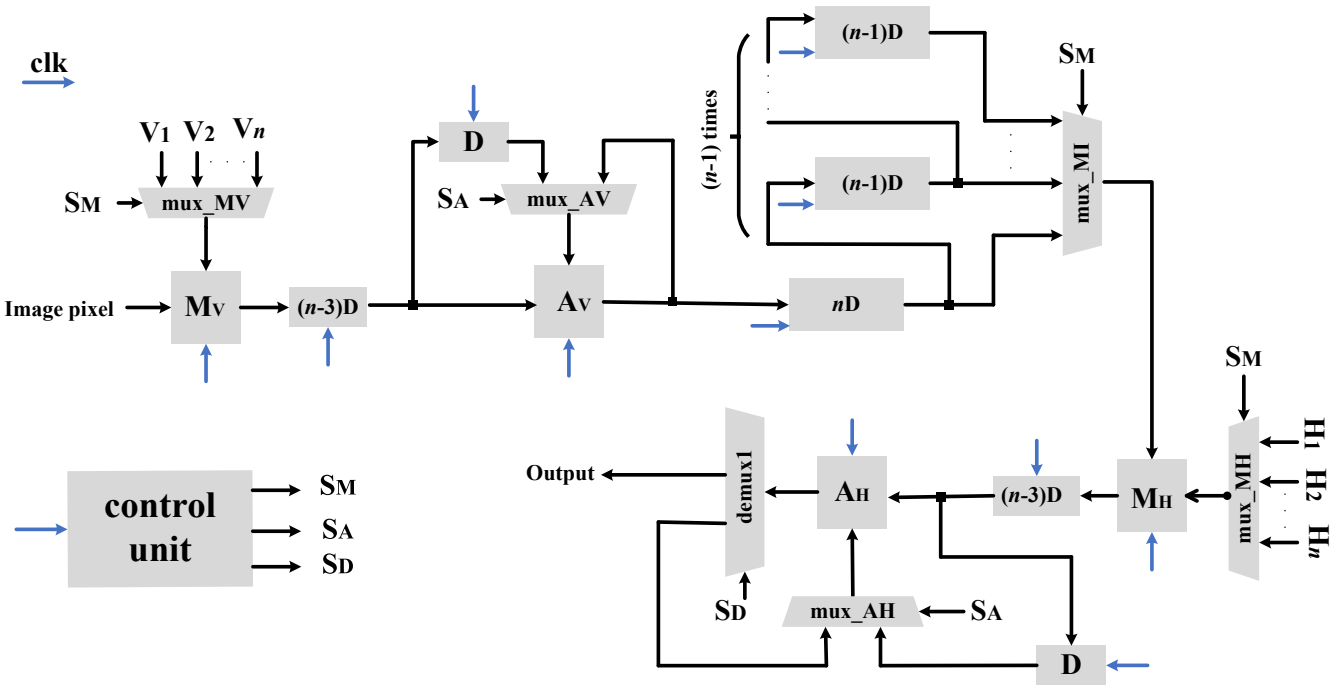

Fig. 2. Folded Separable Convolution architecture (FSCA).

der A_V. The product output is then added either to the previous product output or to the previous adder output using the adder A_V. After every n clock cycles, the intermediate result, i.e., vertical convolution output, is generated and stored in the registers. n shift registers (one shift register of size n and n-1 shift registers of size n-1) are used to store intermediate results. Among the n intermediate results from n shift registers and n horizontal coefficients, appropriate values are assigned to the multiplier M_H, and product results are stored in registers. Like vertical convolution, n-2 registers are used between the multiplier M_H and adder A_H. Adder A_H then adds product output either to the previous product output or to the previous adder output, like the adder A_V. The final output is produced after n clock cycles by the adder A_H through demultiplexer *demux1*. Each output pixel is produced similarly after every *n clock cycles.*

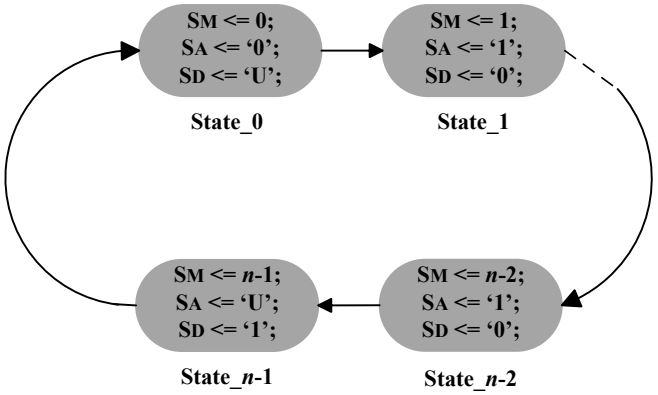

Fig. 3. Control Unit FSM.

Five multiplexers, *mux_MV*, *mux_MH*, *mux_MI*, *mux_AV*, *mux_AH,* and one demultiplexer, *demux1,* are required to perform the time multiplexing for the proposed architecture based on the control signals. Control signals, S_M, S_A, and S_D, are generated by the control unit, a modulo *n-counter* incremented for each clock cycle, where n is the size of the kernel. The control unit is designed by using the finite state machine (FSM), as shown in Figure 3. For kernel of size $n \times n$, the FSM consists of n states, *state_0* to *state_n-1*. The FSM resets to *state_0* after reaching *state_n-1*. The transition from one state to the next will occur at every clock cycle. The values on the selection lines are set according to the FSM's different states, which correspond to the different time instants. A timing diagram for the selection signals for a kernel of size 7×7 is shown in Figure 4.

Signal S_M is sent to the multiplexers, *mux_MV*, *mux_MH*, and *mux_MI,* signal S_A to multiplexers, *mux_AV,* and *mux_AH*, and signal S_D to demultiplexer *demux1*. The value of the signal S_M varies from 0 to *n-1,* incremented for each clock cycle (state). At each clock cycle based on the signal S_M, *mux_MH* and *mux_MV* decide on the n horizontal and vertical filter coefficients, respectively, while *mux_MI* decides on the n intermediate results from the vertical Convolution. Based on signal S_A, multiplexers, *mux_AV*, and *mux_AH*, select one of the inputs to adder A_V and A_H, respectively. For one clock cycle, S_A is low, and for *1* to *n-2* clock cycles, S_A is high, as described in Figure 4. When S_A is low, both inputs to the adders are products stored in registers. When S_A is high, one input to the adders is a product result, and the other is its output generated in the previous clock cycle. Module demultiplexer, *demux1,* sends data to the *mux_AH* when S_D is low and provides the final output when S_D is high. As shown in Figure 4, for *1* to *n-2* clock cycles, signal S_D is low, and at *n-1* clock cycle S_D is high. From Figure 3 and Figure 4, it can be seen that signal S_A and S_D are shown as uninitialized *'U'* (i.e., set to a "don't care" condition) at *state_n-1* and *state_0*, respectively. This is because adders, A_V and A_H, and

demultiplexer, *demux1* are not being used during these clock cycles, and it doesn't matter what value the selection line takes because the output of adders and demultiplexer is not being used at these particular instants of time.

IV. Hardware And Timing Complexity

Table I presents the hardware and timing complexities for the proposed FSCA, SCA [25], RSCA [27], and NPSCA [25] for the $n \times n$ kernel size. Hardware complexity is calculated in terms of the number of multipliers and adders and timing complexity in terms of critical path delay (CP). The proposed architecture requires only two adders and two multipliers, much less than existing architectures. Further, the number of multipliers and adders needed in the proposed design does not change with the kernel size. Thus, the number of multipliers and adders is significantly reduced compared to the existing architectures. Table II presents the resource estimation in terms of multipliers and adders for different kernel sizes for the proposed architecture, SCA [25], RSCA [27], and NPSCA [25], based on the hardware complexity presented in Table I. For a kernel size of 15×15, our proposed design requires 93.33%, 87.5%, and 86.66% fewer multipliers, and 92.85%, 92.85%, and 85.71% fewer adders when compared to SCA [25], RSCA [27], and NPSCA [25], respectively. From Figure 2, it is clear that the critical path in the proposed design will be the time required to complete the multiplication operation (T_m); however, in the proposed design, the multiplier is pipelined by two stages, thereby reducing the critical path to $T_m/2$ u.t. Compared to existing designs, the proposed design has a lesser critical path and is independent of kernel size. Even though our proposed design provides the expected performance results, the actual implementation is necessary to verify these results, as detailed in the next section.

V. Implementation Results

The proposed design is modeled using VHDL language and implemented on Xilinx Artix-7 xc7a35tcpg236-1 FPGA. All hardware modules are synthesized using Xilinx VIVADO 2023.1, and performance metrics are analyzed. Performance is

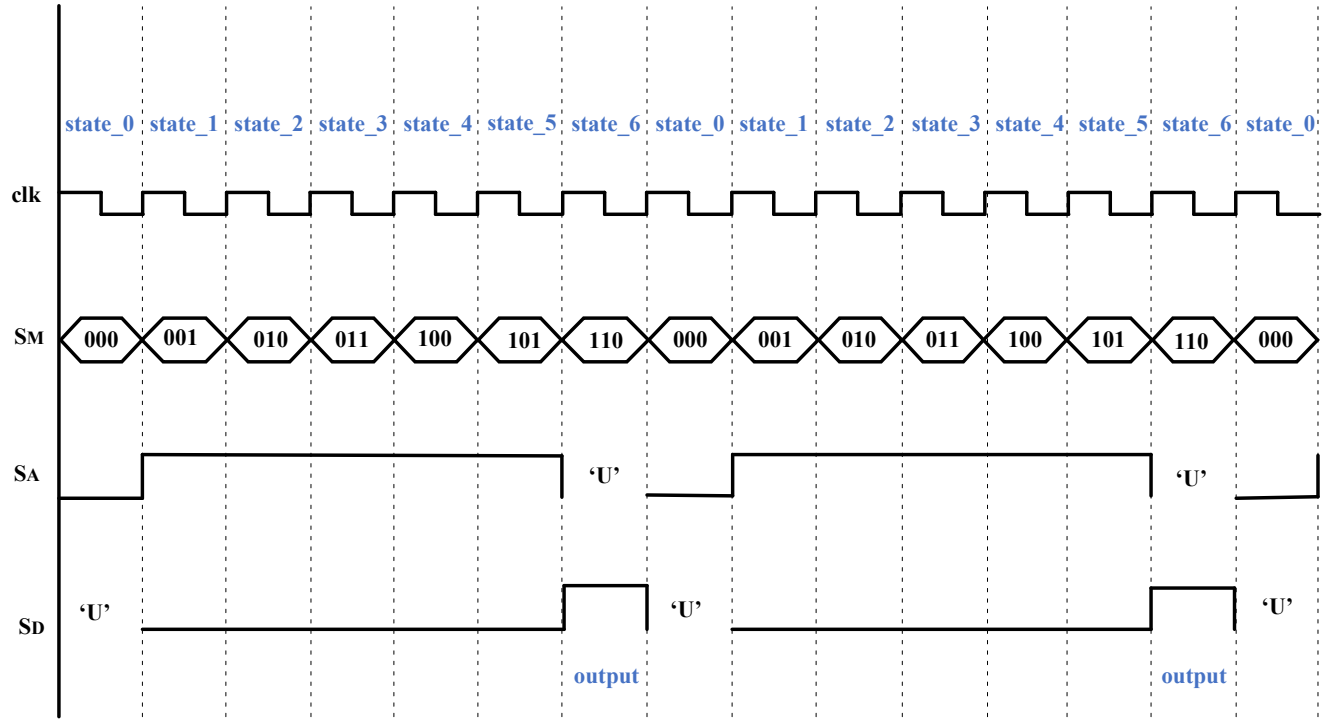

Fig. 4. Timing diagram for a kernel of size 7 x 7.

Table I
Hardware and timing complexity for proposed and existing separable convolution architectures for N×N kernel size.

Method	Multipliers	Adders	Critical Path
SCA [25]	*2n*	*2n-2*	$T_m + (n-1)\,T_a$
RSCA [27]	*n+1*	*2n-2*	$T_m + (n-1)\,T_a$
NPSCA [25]	*n*	*n-1*	$T_m + (n-1)\,T_a$
Proposed FSCA	*2*	*2*	$T_m/2$

Tm: Computation time for the multiplication operation.
Ta: Computation time for the addition operation.

TABLE II
NUMBER OF ADDERS AND MULTIPLIERS REQUIRED IN PROPOSED AND EXISTING SEPARABLE CONVOLUTION ARCHITECTURES.

Filter size	SCA [25]		RSCA [27]		NPSCA [25]		Proposed FSCA	
	Multiplier	Adder	Multiplier	Adder	Multiplier	Adder	Multiplier	Adder
3 x 3	6	4	4	4	3	2	2	2
5 x 5	10	8	6	8	5	4	2	2
7 x 7	14	12	8	12	7	6	2	2
9 x 9	18	16	10	16	9	8	2	2
11 x 11	22	20	12	20	11	10	2	2
13 x 13	26	24	14	24	13	12	2	2
15 x 15	30	28	16	28	15	14	2	2

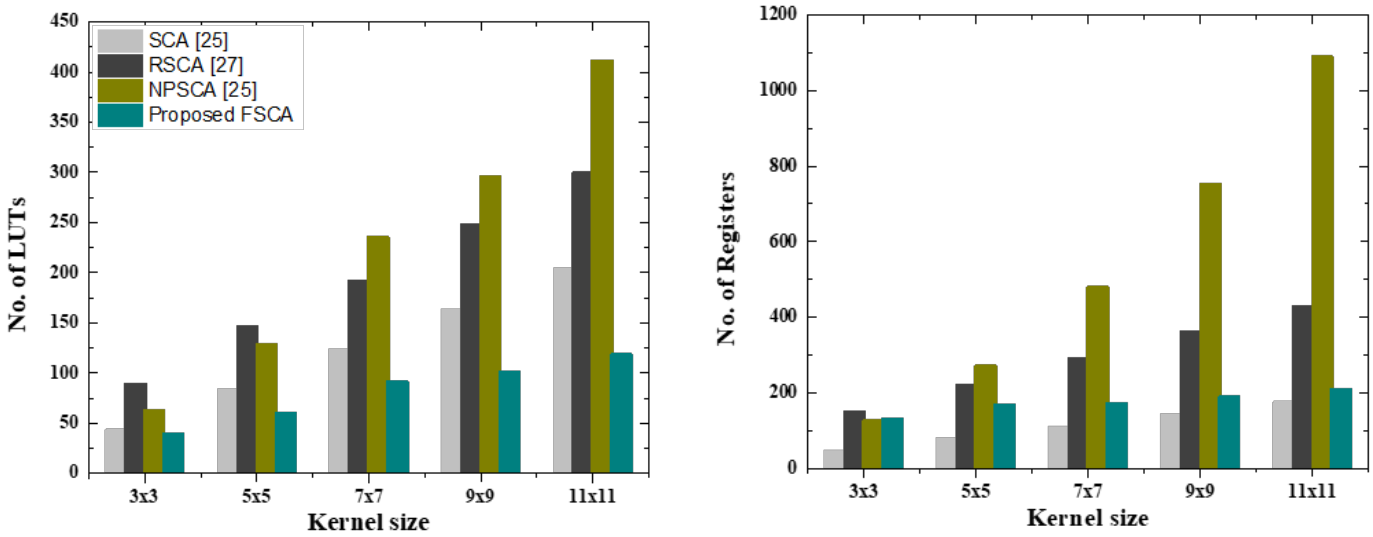

Fig. 5. LUT and register utilization for different architectures for an image size of 128×128 and varied kernel sizes.

compared against SCA [25], RSCA [27], and NPSCA [25] presented in the literature. All the architectures are implemented within the same design environment and using the same set of constraints.

Figure 5 presents the LUT and register utilization for 128 x 128 image and various kernel sizes for different architectures. It can be observed that both the LUT and register utilization in all the designs increase as the kernel size increases. However, the increase in LUT utilization with the kernel size in the proposed design is minimal compared to the other designs. This increase is due to the increase in the number of states in the FSM used to design the control unit and the size of multiplexers as the kernel size increases. For a given kernel size, the proposed design utilizes fewer LUTs than the other designs. For instance, for the kernel size of 11×11, the LUT utilization for the proposed design is 41.66%, 60.33%, and 71.11% lesser when compared to SCA [25], RSCA [27], and NPSCA [25], respectively. Register utilization for the proposed design is comparatively higher than that of the SCA. It is evident that since the proposed design is obtained by time-multiplexing the functional units it requires more registers at different time instants to store the intermediate results. However, an increase in the registers to reduce the functional units, especially the multipliers, is advantageous as multipliers are the main computational burden in the design. Furthermore, register utilization does not result in additional hardware costs since each FPGA logic slice has a large number of registers that will otherwise remain unutilized. For a kernel of size 3×3 and 11×11, our proposed design utilizes 179.16% and 19.3% more registers than SCA [25]. Therefore, as the kernel size increases, the percentage increase in the register utilization decreases. Further, compared to RSCA [27] and NPSCA [25], register utilization for the proposed design is lesser. For a kernel size of 11×11, register utilization for the proposed design is 51.27% and 80.75% less than that of RSCA [27] and NPSCA [25].

Figure 6 provides the timing results regarding the minimum attainable clock period (MCP), maximum operating frequency (F), and critical path delay (CPD) for various kernel sizes. As expected, the proposed design, when compared to SCA [25], RSCA [27], and NPSCA [25], has a lesser minimum attainable clock period and critical path. A lower minimum attainable clock period and critical path result from the reduced number of functional units and the usage of pipelined units in the proposed architecture. It can be pointed out that a gradual increase in the minimum attainable clock period and the critical path occurs with the increase in kernel size. This is due to the increased complexity of the design as the kernel size increases. Nevertheless, as anticipated, in SCA [25], RSCA [27], and NPSCA [25], as kernel size increases, a prominent increase in the minimum attainable clock period and critical path is reported. For a kernel

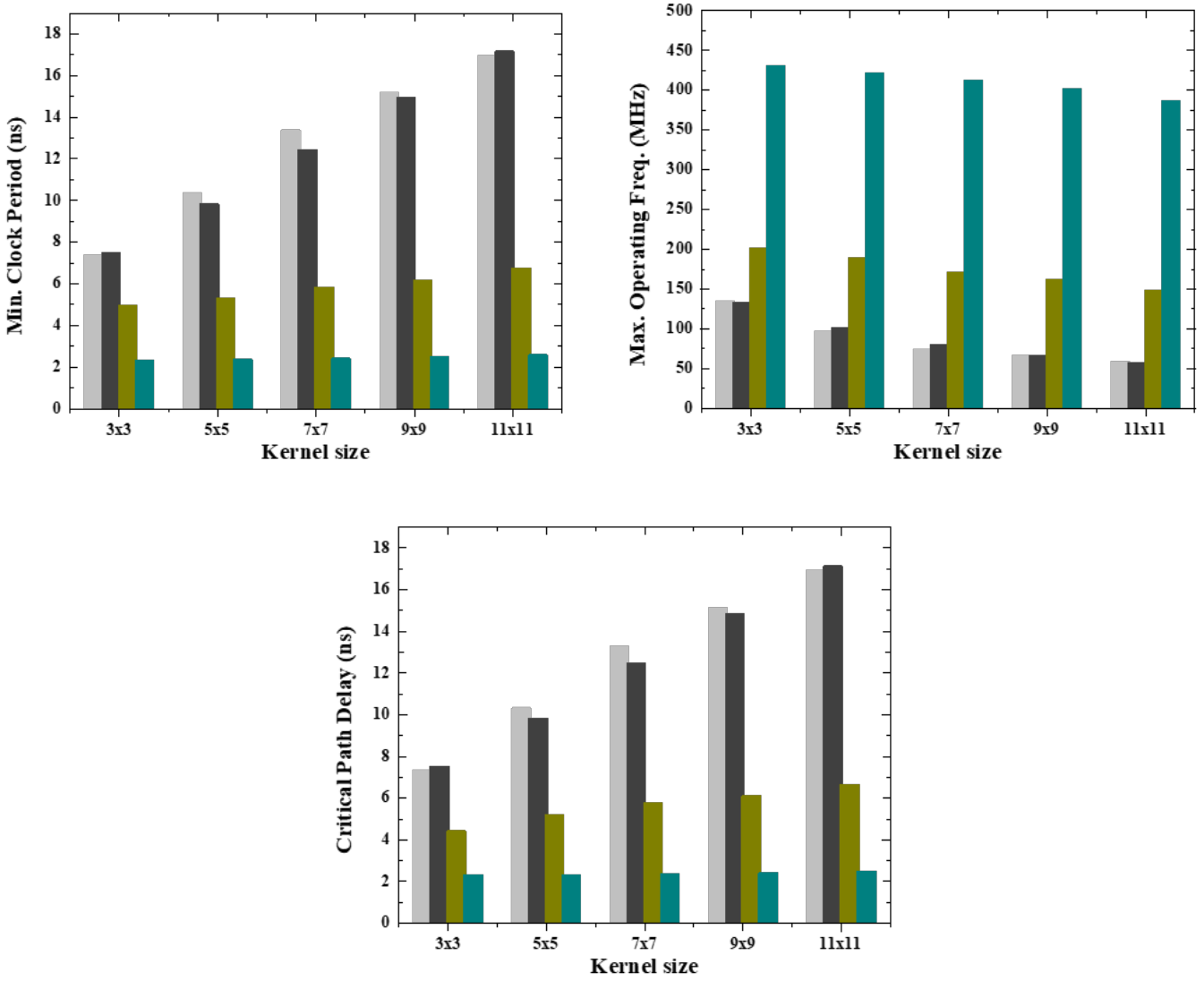

Fig. 6. Timing analysis for different architectures for an image size of 128×128 and varied kernel sizes.

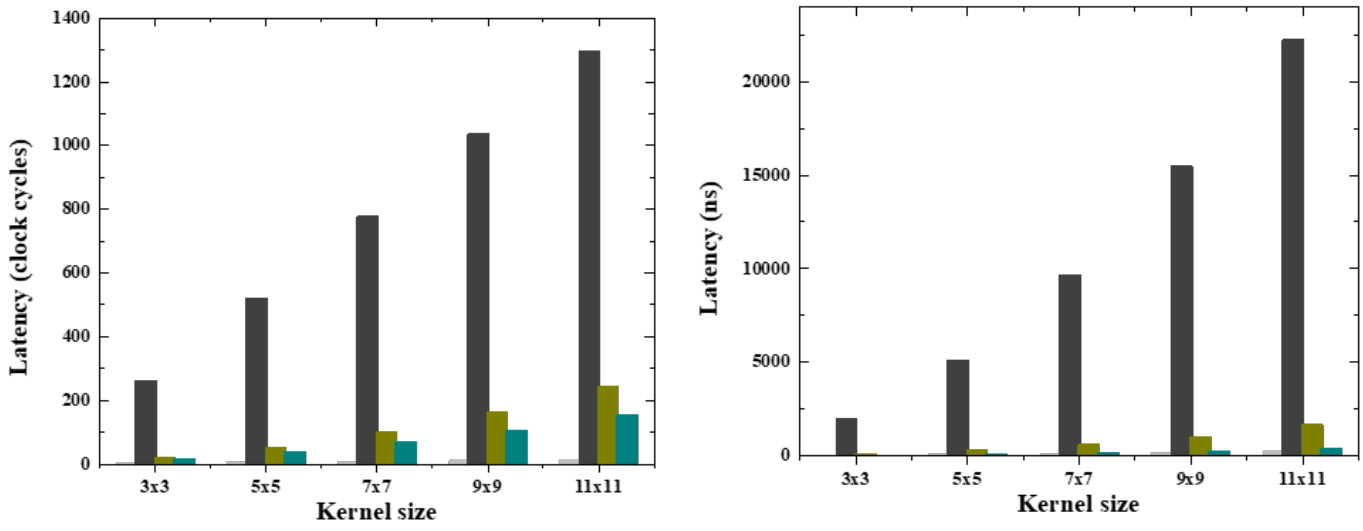

Fig. 7. Latency for different architectures for an image size of 128×128 and varied kernel sizes.

size of 11×11, the minimum achievable clock period and critical path for the proposed design is 84.72%, 84.92%, 61.63%, and 85.17%, 85.34%, 62.37% lesser when compared to SCA [25], RSCA [27], and NPSCA [25], respectively. Additionally, Figure 7 compares the latency of different architectures. Latency is calculated in terms of the clock cycles required to process one output pixel. As the proposed architecture is based on folding transformation, latency will increase as compared to the SCA [25]. This is obvious because the EMB of SCA [25] is *n* times higher than the proposed design. However, higher EMB leads to higher memory requirements and higher power dissipation, limiting the usability of SCA [25] for large kernel sizes. When compared to RSCA [27] and NPSCA [25], the latency of the proposed design is much less. Further, as mentioned earlier, the clock period of the proposed design is less than that of the other designs. Thus, for a fair comparison, latency is also calculated in *ns* by multiplying the MCP by the number of clock cycles. For a kernel size of 11×11, latency for the proposed design is 98.22% and 75.84% less than that of RSCA [27] and NPSCA [25] and 94.70% higher than that of SCA [25].

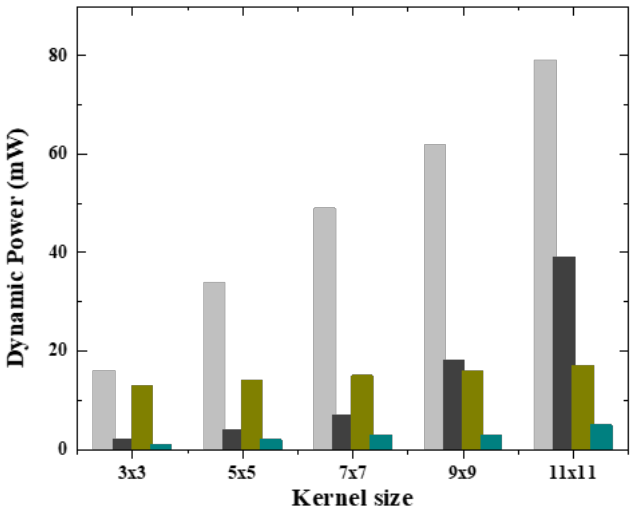

Fig. 8. Power analysis in mW for different architectures for an image size of 128×128 and varied kernel sizes.

Figure 8 presents the power analysis for the different architectures. The proposed design reports a lesser dynamic power dissipation when compared to SCA [25], RSCA [27], and NPSCA [25]. The reduction in power dissipation is expected due to the decreased functional units and the switching activity in the proposed design. Further, the large power dissipation in SCA [25] is also due to multiple pixel access, and in RSCA [27], it is due to the large buffer lines. For a kernel of size 11×11, the power dissipation in the proposed design is 93.67%, 87.17%, and 70.58% less when compared to SCA [25], RSCA [27], and NPSCA [25], respectively.

VI. Conclusion

In this study, the folded separable convolution architecture is proposed that requires only two adders and two multipliers, irrespective of the kernel size. Thus, for a kernel of size $n \times n$, the computational complexity is approximately reduced by factor *n*. In addition to the functional units, since only one pixel is accessed per clock cycle, the EMB of the proposed design is *one pixel/clock cycle*. Experimental results demonstrated that the proposed architecture utilizes fewer LUT slices and has reduced critical path and power dissipation compared to existing designs. The future discourse of the research will focus on designing time-multiplexed architectures for the non-separable convolution architecture.

References

[1] S. Perri, M. Lanuzza, P. Corsonello, G. Cocorullo, "A high-performance fully reconfigurable FPGA-based 2D convolution processor," *Microprocess. Microsy.*, vol. 29, no. 8-9, pp. 381–391, Nov. 2005.

[2] R. C. Gonzalez, "Digital image processing", *Pearson education India*, 2009.

[3] S. Sadangi, S. Baraha, D. K. Satpathy, P. K. Biswal, "FPGA implementation of spatial filtering techniques for 2D images," in *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, 2017, pp. 1213–1217.

[4] J. Lee, H. Tang, J. Park, "Energy efficient canny edge detector for advanced mobile vision applications," *IEEE T. Circ. Syst. Vid.*, vol. 28, no. 4, pp. 1037–1046, Dec. 2018.

[5] D. Mukherjee, S. Mukhopadhyay, "Fast hardware architecture for fixed-point 2D Gaussian filter," *AEU-Int. J. Electron. C.*, vol. 105, pp. 98–105, Jun. 2019.

[6] B. Bosi, G. Bois, Y. Savaria, "Reconfigurable pipelined 2-D convolvers for fast digital signal processing," *IEEE T. VLSI Syst.*, vol. 7, no. 3, pp. 299–308, Sep. 1999.

[7] H. Zhang, M. Xia, G. Hu, "A multiwindow partial buffering scheme for FPGA-based 2-D convolvers," *IEEE T. Circuits-II*, vol. 54, no. 2, pp. 200–204, Feb. 2007.

[8] M. Kazmi, A. Aziz, H. R. Khan, S. A. Qazi, and L. K. Stergioulas, "Resource-Efficient Image Buffer Architecture for Neighborhood Processors," *IEEE Access*, vol. 8, pp. 181 964–181 975, Sep. 2020.

[9] V. Sriram, D. Kearney, "A FPGA implementation of variable kernel convolution," in *Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2007)*, Adelaide, SA, Australia, 2007, pp. 105–110.

[10] M. Kalbasi, H. Nikmehr, "A classified and comparative study of 2-D convolvers," in *2020 International Conference on Machine Vision and Image Processing (MVIP)*, Iran, 2020, pp. 1–5.

[11] G. D. Licciardo, C. Cappetta, L. Di Benedetto, A. Rubino, R. Liguori, "Multiplier-less stream processor for 2D filtering in visual search applications," *IEEE T. Circ. Syst. Vid.*, vol. 28, no. 1, pp. 267–272, Aug. 2018.

[12] D. Mukherjee, S. Mukhopadhyay, "Fast hardware architecture for 2-D separable convolution operations," *IEEE T. Circuits-II*, vol. 65, no. 12, pp. 2042–2046, Mar. 2018.

[13] F. Mamalet, C. Garcia, "Simplifying convnets for fast learning," in *International Conference on Artificial Neural Networks*, Berlin, Heidelberg, 2012, pp. 58–65.

[14] J. Wang, J. Lin, Z. Wang, "Efficient convolution architectures for convolutional neural network," in *2016 8th International Conference on Wireless Communications & Signal Processing (WCSP)*, Yangzhou, China, 2016, pp. 1–5.

[15] Z.-B. Ma, Y. Yang, Y.-X. Liu, A. A. Bharath, "Recurrently decomposable 2-D convolvers for FPGA-based digital image processing," *IEEE T. Circuits-II*, vol. 63, no. 10, pp. 979–983, Feb. 2016.

[16] J. Chang, J. Sha, "An efficient implementation of 2D convolution in CNN," *IEICE Electron. Expr.*, vol. 14, no. 1, pp. 20161134–20161134, 2017.

[17] S. I. Cho, S.-J. Kang, "Gradient prior-aided CNN denoiser with separable convolution-based optimization of feature dimension," *IEEE T. Multimedia*, vol. 21, no. 2, pp. 484–493, Jul. 2019.

[18] M. Kr´alik, L. Lad´anyi, “Canny edge detector algorithm optimization using 2D spatial separable convolution,” *Acta Electrotech. Inform.*, vol. 21, no. 4, pp. 36–43, 2021.

[19] J. Y. Mori, C. H. Llanos, P. A. Berger, “Kernel analysis for architecture design trade off in convolution-based image filtering,” in *2012 25th Symposium on Integrated Circuits and Systems Design (SBCCI)*, Brasilia, Brazil, 2012, pp. 1–6.

[20] M. Kalbasi and H. Nikmehr, “A Fine-Grained Pipelined 2-D Convolver for High-Performance Applications,” *IEEE T. Circuits-II*, vol. 66, no. 1, pp. 146–150, May 2018.

[21] L. Rao, B. Zhang, J. Zhao, “Hardware implementation of reconfigurable 1D convolution,” *J. Signal Process.*, vol. 82, pp. 1–16, Jan. 2016.

[22] A. Dehghani, A. Kavari, M. Kalbasi, K. RahimiZadeh, “A new approach for design of an efficient FPGA-based reconfigurable convolver for image processing,” *J Supercomput.*, vol. 78, pp. 2597–2615, Feb. 2022.

[23] L. Rao, B. Zhang, J. Zhao, “Hardware implementation of reconfigurable separable convolution,” in *2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, Hong Kong, China, 2018, pp. 232–237.

[24] A. Arumalla and M. L. Makkena, “Efficient Architecture for Block Parallel Convolution using Two-Dimensional Polyphase Decomposition,” *Circ. Syst. Signal. Pr.*, vol. 41, no. 2, pp. 1166–1186, Feb. 2022.

[25] A. K. Joginipelly, D. Charalampidis, “Efficient separable convolution using field programmable gate arrays,” *Microprocess. Microsy.*, vol. 71, pp. 102852, Nov. 2019.

[26] Y. Hu, V. K. Prasanna, “Energy-efficient parameterized 2-D separable convolution on FPGA,” in *International Green Computing Conference*, Dallas, TX, USA, 2014, pp. 1–10.

[27] H. Kim, H. Yoo, “Area-efficient two-dimensional separable convolution structure,” *J. Imaging. Sci. Techn.*, vol. 63, no. 5, pp. 50404–1, Sep. 2019.

[28] K. K. Parhi, C.-Y. Wang, A. P. Brown, “Synthesis of control circuits in folded pipelined DSP architectures,” *IEEE J. Solid-St. Circ.*, vol. 27, no. 1, pp. 29–43, Jan. 1992.

[29] K. K. Parhi, “VLSI digital signal processing systems: design and implementation,” *John Wiley & Sons*, 2007.

Accelerated Curve Fitting Approach for Denoised NQR Signal Parameter Estimation

K. Thulasiram Varma and Rangababu Peesapati

***Abstract*—Signal parameters such as amplitude, frequency, decay constant, and phase play an important role in identifying the signal's origin in many real-life applications. Standard gradient-based curve-fitting approaches are computationally complex to implement on hardware because of large matrix multiplication and partial derivatives. The current work demonstrates a feasible approach to extract parameters from exponential decay Nuclear Quadrupole Resonance (NQR) signals using derivative-free optimization approaches on a PYNQ-Z2 FPGA board. The hardware implementation results indicate 0.12 seconds is required to extract the amplitude, frequency, decay constant, and phase from $5x10^{-3}$ sec signals with the residual error in the order of 10^{-5}.**

***Index Terms*—RF-interference, frequency-domain, NQR spectroscopy, System on Chip (SoC), PYNQ-Z2 FPGA.**

Original Research Paper
DOI: 10.53314/ELS2428022V

I. Introduction

OUTPUT signal parameter estimation is a successive step after capturing the signal from the real-time sensors to infer the scenario of signal origin. This is an area of interest in many applications like mechanical engineering, biological signal processing, spectroscopy technology, etc. In these domains, parameter estimation can provide deeper insights into the system's functionality and operation. Nuclear Quadrupole Resonance (NQR) spectroscopy is one such field where parameter estimation plays a critical role. NQR spectroscopy falls under the category of non-destructive testing methods. It identifies and classifies chemical substances by examining their nuclear quadrupole resonance properties. It involves exiting the substance under the test with an external Radio Frequency (RF) pulse. In NQR spectroscopy, signal parameters give us an insight into the material and the composition used to make it. This capability of identifying chemical substances helps in the files of pharmaceuticals, security screening, and material science. In this paper, we worked on the estimation of output signal parameters specifically related to NQR spectroscopy.

Parameter estimation is usually performed using a curve-fitting approach, where a mathematical function mimics the actual signal with the least residual error. It involves an iterative approach that converges the solution to a mathematical function. There are two main optimization approaches to converge the solution: gradient and non-gradient. Gradient approaches are popular for their fast convergence rate and better approximation. On the other hand, non-gradient-based methods are simple to implement but require significant iterations to obtain a global minimum. Both approaches are capable of approximating linear and nonlinear-based systems of equations.

Nonlinear least square regression is a widely used gradient-based approach for approximating nonlinear curves with exponential and periodic properties. Gradient descent and the Gauss-Newton algorithm are used to implement nonlinear least square regression. The Levenberg-Marquardt method is a hybrid between Gradient descent and the Newton method. It acts like gradient descent when the optimal values are far from the global minimum and works as the Newton method when the global minima are near [1] [2].

Initial guess vector, step size, and computation complexity are critical bottlenecks to be addressed while implementing a gradient-based approach on hardware. Small step sizes will lead to slow convergence, while larger ones will overshoot the convergence area. On the other hand, the initial guess vector, subject to the developer and not data-driven, leads to uncertainty for algorithms in finding global minima. Unlike linear regeneration, which requires Multiply-Accumulate Operation (MAC) and division operations, as well as nonlinear least square regression, consume a significant amount of computation resources to perform matrix multiplication, inversion, and partial derivatives.

This work presents an optimal solution with a data-driven approach to find the region of interest and global minima. Once the area of interest is identified, non-gradient-based systems are introduced to optimize the curve fit with the least residual error. This will eliminate the initial guess procedure and an undefined number of iterations to converge the solution with lesser computation resources.

Related works in curve fitting and parameter estimation are presented in section two. The third section explains the nonlinear least square regression theory and the proposed work concepts. The fourth section consists of the hardware setup to implement the proposed algorithm. The fifth section presents the results and discussion, followed by a conclusion and references.

II. Related Works

Multiple research studies were carried out on curve fitting and parameter estimation using gradient and non-gradient approaches. Jim et al., [3], and his team worked on parameter

Manuscript received on October 8th, 2023. Received in revised form on April 26th, June 27th, and July 17th, 2024. Accepted for publication on July 18th, 2024.

K. Thulasiram Varma is with the Department of Electronics and Communication Engineering, National Institute of Technology Meghalaya, Shillong 793003, India. (e-mail: kthulasivarma@nirtm.ac.in; phone: +917888967498).

Rangababu Peesapati is with the Department of Electronics and Communication Engineering, Indian Institute of Information Technology, Design and Manufacturing, Kurnool, 518 008, India (e-mail: p.rangababu@iiitk.ac.in).

estimation of decaying signals using dense autoencoder neural networks. The study is carried out on an exponential decay envelope, and decaying oscillation signals of length 5 μs, 200, and 1000 such signals were considered in the training data set. The output obtained from the auto-encoder method is compared with the least-square fit and observed at a similar estimation error rate.

Boston et al., [4] work down Cavity Rain Down Spectroscopy (CRDS), where exponential decay constant plays a vital role in identifying the interaction of light with the gas in the cavity. The 8192 points Fast Fourier Transform (FFT) was implemented on a Virtex-4 Field Programmable Gate Array (FPGA) at the rate of 4.4 kHz. The residual error between the actual and estimated signal is in the range of - 3 to 3 mV.

Abba et al., [5], and their team implemented nonlinear least square fitting on FPGA using both Gauss-Newton and Levenberg-Marquardt approaches. The experimental results suggest that the proposed work required 60 μs to estimate the amplitude and occurrence time of the event of signals.

Although the literature suggests multiple works to estimate the signal properties using gradient-based and other approaches, there is a need to develop simple and computationally inexpensive methods to extract signal parameters.

As a continuation of previous work where a noisy NQR signal is denoised using Singular Spectral Analysis (SSA) and observed 32.4 dB gain in Signal to Noise Ratio (SNR) [6]. We carry forward the SSA output signal to this present work to estimate the signal parameters using the proposed method.

The proposed method is compared with existing Nonlinear Least-Square Fit (NLLSF) on software, and similar results were observed. In addition, the proposed work provides the simplest solution to implement the parameters estimation algorithm on FPGA hardware.

III. Curve Fitting and Parameters Extraction

Curve fitting finds its application in many engineering and applied science domains, which helps in approximating the desired signal mathematically [7]. The functional coefficients, which approximate the signal, further help identify the signal properties and origin. Curve fitting is an iterative approach that involves optimization techniques to converge the solution faster. In general, gradient and non-gradient are two optimization approaches, also known as derivative and derivative-free optimization [8].

A. Nonlinear Least-Square Fit (NLLSF)

Non-linearly Least Square Fit (NLLSF) try to estimate the unknown parameters of a model function and approximate the actual signal with the least residual error. Equation (1) corresponds to residual error R, the difference between the observed signal y and the estimated signal with C unknown parameters. The relationship between unknown parameters C with all the samples in the signal is shown in (2). $C^{'}$ corresponding to the new estimated value of the unknown parameter. By solving (2) and (3), the function of $C^{'}$ and C are simplified and result in (4), corresponding to the rate of change in the difference of unknown parameters (ΔC).

$$R = \sum \|F(x_i, C_j) - y_i\|^2 \tag{1}$$

$$F(x_i, C_j^{'}) = F(x_i, C_j) + \frac{\partial F}{\partial C_j}(C_j^{'} - C_j) \tag{2}$$

$$Y_i = Y_i - R_i + \frac{\partial F}{\partial C_j}(C_j^{'} - C_j) \tag{3}$$

$$R_i = \frac{\partial F}{\partial C_j}(\Delta C_j) \tag{4}$$

The above equations result in a Jacobian matrix A of size i, j, where i corresponds to the length of the signal and j is equal to the number of unknown parameters of the model function, as shown in (5). The Jacobian matrix is multiplied with the column vector of (ΔC), resulting in residual error R. Operations in (7) are performed to converge the solution of unknown parameters, which involves matrix multiplication and inversion.

$$\begin{bmatrix} \frac{\partial F_1}{\partial C_1} & \frac{\partial F_1}{\partial C_2} & \cdots & \frac{\partial F_1}{\partial C_j} \\ \frac{\partial F_2}{\partial C_1} & \frac{\partial F_2}{\partial C_2} & \cdots & \frac{\partial F_2}{\partial C_j} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial F_i}{\partial C_1} & \frac{\partial F_i}{\partial C_2} & \cdots & \frac{\partial F_i}{\partial C_j} \end{bmatrix}_{i,j} \cdot \begin{bmatrix} \Delta C_1 \\ \Delta C_2 \\ \cdots \\ \Delta C_j \end{bmatrix}_{j,1} = \begin{bmatrix} R_1 \\ R_2 \\ \cdots \\ R_i \end{bmatrix}_{i,1} \tag{5}$$

$$A.\Delta C = R \tag{6}$$

$$\Delta C = (A^T A)^{-1} A^T R \tag{7}$$

B. Parameters Estimation and Optimization

In the current work, curve fitting is performed on the NQR signal, which is a dampened sinusoidal signal with four fundamental parameters (Amplitude, Frequency, Decay constant, and phase), unlike a simple sinusoidal signal with amplitude, frequency and phase as a fundamental components decay parameter make the NQR signal more complex to approximate (Fig. 1). Based on these parameters, the cure fitting algorithm is designed. Parameters estimation and optimization start with exponential curve fitting. It tries to estimate the decay rate in the exponential envelope using standard linear straight-line fitting. In (8), A and D correspond to amplitude and decay constant, respectively. As shown in (9), using the natural log, the exponential form of the equation is converted to a linear equation. The unknown parameters Amplitude (A) and Decay constant (D), which are in straight line form as y-intercept and slope, is calculated using (10) [9].

$$y_i = Ae^{Dx_i} \tag{8}$$

$$\ln(y_i) = \ln(A) + Dx_i \tag{9}$$

$$\begin{aligned} D &= \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2} \\ A &= \frac{\sum x_i^2 \sum y_i - \sum x_i \sum x_i y_i}{n\sum x_i^2 - (\sum x_i)^2} \end{aligned} \tag{10}$$

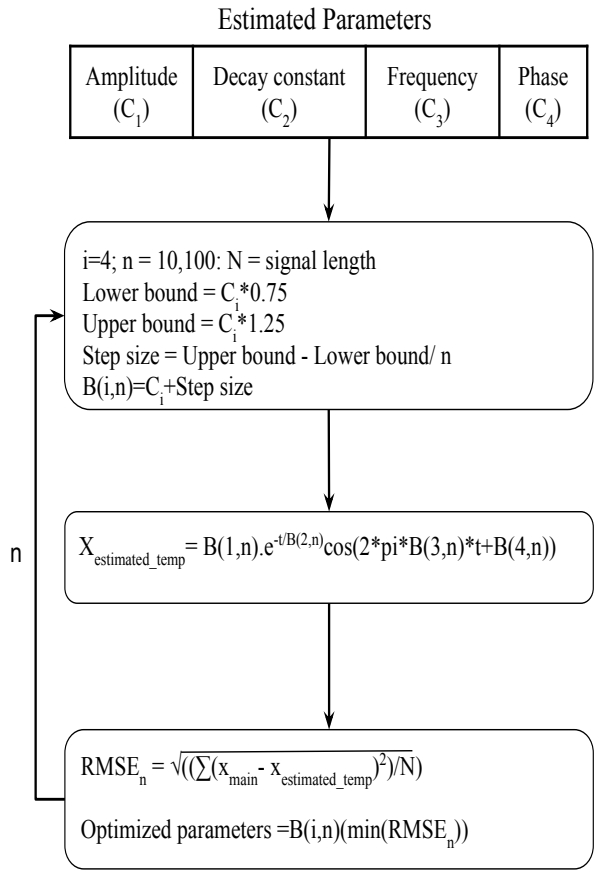

Fig. 1. Proposed curve fit algorithm alternative to NLLSF.

Once the exponential curve fitting is performed, the frequency and phase of the signal are calculated using (11), where pt_2-pt_1 corresponds to the different times between two successive peaks and t_2-t_1 represents the sampling interval.

$$\begin{aligned} Frequency &= 1/pt_2 - pt_1 \\ Phase &= 360 * (t_2 - t_1/pt_2 - pt_1) \end{aligned} \tag{11}$$

The estimated parameters act as a region of interest and non-gradient-based optimization is used to find the global minima. The main objective of this technique is to find the minimum value of an unimodal function. As shown in the flow chart, an iterative approach was modelled with the help of both exhaustive search and region elimination rules. Lower and upper bounds were initially calculated, defining the region of interest for each estimated parameter. The signal model function is passed through all new estimated parameters at each step in the region of interest, and Root Mean Square Error (RMSE) is calculated with respect to the actual signal. The parameters with the least RMSE are considered, and the remaining parameters are discarded; this step further narrows the search area. The above-mentioned steps are performed again by considering the least RMSE parameters as new estimated parameters with a smaller step size [10].

Algorithm 1 FINDING ENVELOP OF THE SIGNAL

Input : signal [N], time[N]
Output : peaks[s], pk_time[s]
Ensure: $i, s = 0$
  for all $i < N - 1$ **do**
    if $signal[i] > 0$ **then**
      if $signal[i-1] < signal[i] and signal[i] > signal[i+1]$ **then**
        $peaks[s] = signal[i]$
        $pk_time[s] = time[i]$
        $s = s + 1$
      end if
    end if
  end for

Algorithm 2 CALCULATE SIGNAL PARAMETERS

Input : pk_time[s], peaks[s],time[N]
Output : est_parameters[4]
Ensure: $x, x_2, y, xy, i, temp = 0$
  for all $i < s$ **do**
    $iny[i] = log(peaks[i])$
    $x = x + pk_time[i]$
    $y = y + lny[i]$
    $x_2 = x_2 + pk_time[i]\text{^}2$
    $xy = xy + pk_time[i] * lny[i]$
  end for
  $est_parameters[2] = (s * xy - x * y)/(s * x_2 - x * x)$
  $temp = (s * xy - x * y)/(s * x_2 - x * x)$
  $est_parameters[1] = e\text{^}temp$
  $td = pk_time[1] - pk_time[0]$
  $td_1 = time[1] - time[0]$
  $est_parameters[3] = 1/td$
  $est_parameters[0] = (360 * (td_1/td))$

Algorithm 3 OPTIMIZATION

Input : est_parameters[4],signal[N],pk_time[s]
Output : opt_parameters[4]
Ensure: $n_o = 0.1, N_o = 0.01, i, RMSE[n], s = 0$
  for all $j < 4$ **do**
    $lower_bound = est_parameters[i] * 0.75$
    $upper_bound = est_parameters[i] * 1.25$
    $step_size = (upper_bound - lower_bound)/n_0$
    for all $i < n_o$ **do**
      $opt_parameters[j] = lower_bound$
      for all $k < N$ **do**
        $temp_signal = (opt_parameters[1]) * (e\text{^}time[k] * opt_parameters[2]) * (cos(2 * pi * opt_parameters[3] * time[k] + opt_parameters[0])$
      end for
      $RMS_error[i] = RMSE(signal - temp_signal)$
      $opt_parameters[j] = lower_bound + step_size$
    end for
    $min_RMSE = min(RMS_error)$
    $new_opt_parameters = opt_parameters(min_RMSE)$
  end for

IV. Hardware Implementation

The Parameter estimation and optimization algorithm are implemented on the PYNQ-Z2 FPGA board, which has a 650MHz dual-core ARM Cortex-A9 processor and 512MB DDR3 RAM. This section details the implementation process and optimization techniques used for the FPGA-base solution.

Our design consists of two main functional blocks: region of interest calculation and parameters optimization. Both functions are implemented on a PYNQ-Z2 FPGA board using development tools such as Vivado HLS, Vivado and the algorithms were initially developed in MATLAB.

The region of interest for all the parameters is calculated using (8) to (11), explained in algorithms 1 and 2. The exponential decaying NQR signal can be divided into two stages: exponential and periodic. Using algorithm 1, all the periodic peaks were identified and interpolated to capture the signal's exponential property. Later, in algorithm 2, with the help of exponential curve fitting, the non-linear equation is converted into a linear form. All these calculations are performed, as shown in (8) to (10), to capture the amplitude and decay constant of the signal. Finally, Equation (11) calculates the signal's frequency and phase. Estimated_parameters [0:3] correspond to phase, amplitude, decay contact, and frequency.

The estimated parameters are further optimized to fit the actual signal in the second functional block. As explained in Fig. 1 and algorithm 3, the lower and upper bounds were initially calculated by considering a step size of 0.1, and RMSE is calculated with respect to the actual signal. The parameters with the least error are considered local minima. The second optimization set is performed by considering 0.01 as the step size around these new parameters, which results in a globally optimal solution. All the optimized parameters are appended onto a vector and pushed back to PS using DMA. In algorithm 3, temp_signal corresponds to the NQR model function, as (12). (N) represents the legend of the signal, i.e., 4868, and (n) is equal to the number of unknown parameters. The detailed implementation flow of the proposed parameter extraction algorithm is illustrated in Fig. 2. The process begins with implementing the algorithm in MATLAB to identify any bottlenecks that could affect the hardware implementation. After identifying these bottlenecks, the entire algorithm is translated into an Intellectual Property (IP) core. This IP core is then used to generate a bitstream for the Programmable Logic (PL) section of the Field-Programmable Gate Array (FPGA). To compare the time and power consumption between the PL and the Processing System (PS) of the FPGA, we also translated the IP code into Python and executed it on the PS.

Coming to the entire system architecture, as shown in Fig. 3, the denoised signal from the SSA algorithm is collected and fed as input to the parameter estimation block on PL. 4868 sampled denoised NQR signal is passed from the Processing System (PS) to Programmable Logic (PL) using Direct Memory Access(DMA). Timing constraints were defined and met to ensure the design operates at the required clock frequency, as shown in the implementation timing diagram (Fig. 4).

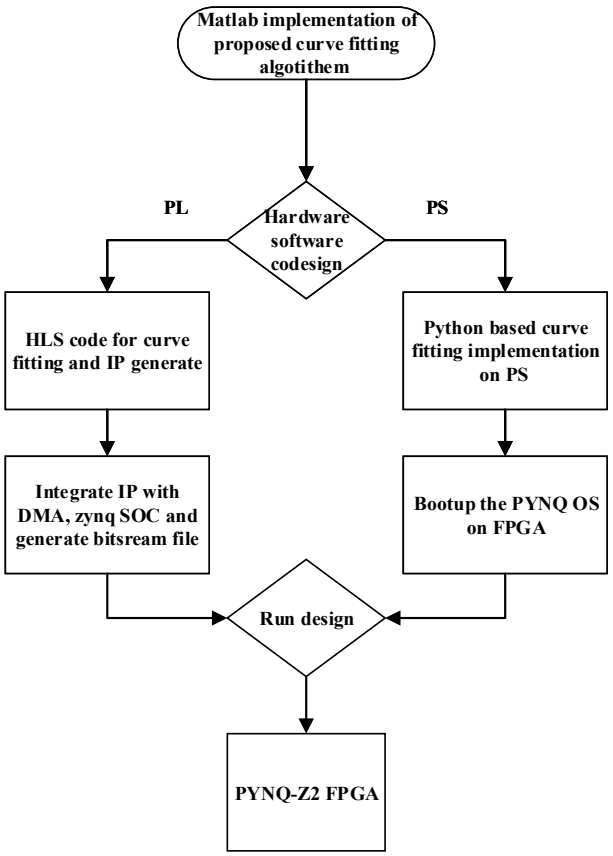

Fig. 2. Design implementation flow.

The outcome metrics, such as resource utilization, power consumption, and speed, are provided in the results section.

$$S = A.e^{-t/T}cos(2\pi * Fc * t) \tag{12}$$

V. Results and Discussion

All four unknown parameters are extracted with the help of the Matlab curve fitting toolbox and signal modelling equation, as shown in (12). The residual error calculates the difference between the estimated and actual signals in the order of 10^{-5}. To implement the same on an FPGA, the computational complexity comes into the picture, as explained in section two of this paper; the size of the Jacobin Matrix will increase with respect to signal length, and we need to perform an inversion operation for the whole matrix. From [11], we can observe that the PYNQ-Z2 board can implement only matrix inversion to the order of 100x100. Apart from this, the NLLSF curve fitting method requires multiple partial derivative operations. The timing diagram in Fig. 4 provides a clear picture of data transfer from PS to PL via master Advanced eXtensible Interface (AXI) stream and parameters estimation at the (IP) accelerator using algorithms 1, 2 and 3 and getting the data back to the PS using slave AXI stream. It also details the pipeline and loop unrolling stages used in the proposed work.

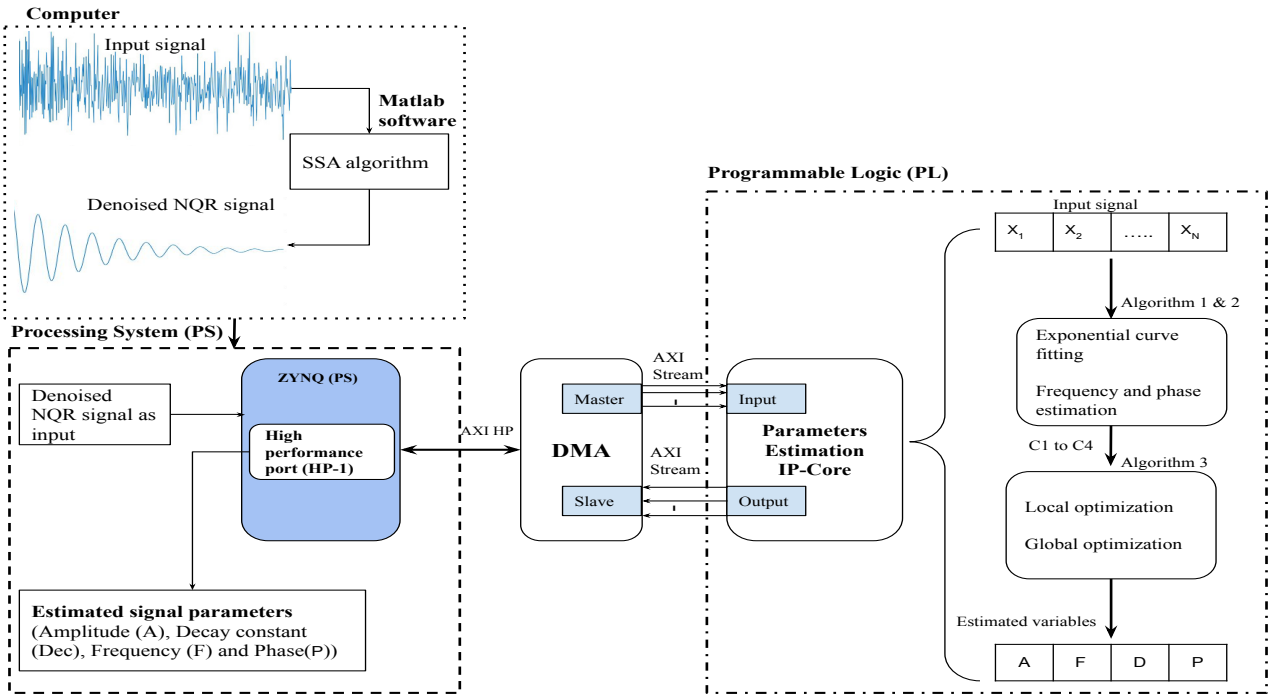

Fig. 3. SoC block diagram for proposed curve fit hardware.

The total clock cycles consumed by the proposed design is 12 M at the clock frequency of 100 Mhz.

To avoid this mathematical complexity, an optimal and simple non-gradient-based approach is explored, as explained in the previous section, and the same is implemented on PYNQ-Z2 FPGA. The proposed approach to extract signal parameters is coded using sequential and pipeline fashions on Vivado HLS. Table I depicts the performance of the proposed method on different NQR signals collected from the sensor. Without optimization, the column corresponds to RMS error calculation based on Algorithms 1 and 2, where standard exponential curve fitting, frequency, and phase estimation are performed. RMSE is further drawdown by using proposed non-gradient-based optimization techniques as explained in Algorithm 3 and reported the error in the order of 10^{-4} and 10^{-5}, which is similar to NLLSF in software. As shown in Table I, the Pearson Correlation Coefficient (PCC) metric indicates the correlation between the actual and estimated signal. Value 1 is highly correlated, and 0 represents no correlation [12]. Fig. 5 showcases the comparison plot between the SSA output signal and the approximate signal using estimated parameters.

We implemented the proposed method using sequential and pipeline approaches. The resources and performance measures of both approaches are tabulated in Table II. In the pipeline approach of implementing the algorithms, we used HLS pragma Pipeline II=1 along with loop unrolling with factors 2 and 4. Pipeline II (Initiation Interval) corresponds to the number of clock cycles between the start of two consecutive iterations. Loop unrolling is used to duplicate the logic so that the total number of iterations reduces, which overall increases the speed of execution. These primitives consumed 2.86% BRAM, DSP48E 29.1%, FF 9.34%, and LUT 23.78% more FPGA resources and improved the performance by 16.9X times with reference to a sequential approach. The total time PS consumes to implement the same algorithm is much higher than PL, which is 824.7 sec.

$$FOM = [(LUT + FF) * 8] + [(BRAM + DSP) * 16] \quad (13)$$

The power requirements to implement the algorithm and Figure of Merit (FOM), which measures the algorithm's actual performance, is a calculator based upon (13) is shown in Table II. FOM provides a quantitative measure of the resources used by the design in terms of Lookup Table (LUT), Flip flops, Block Random Access Memory (BRAM), and Digital Signal Processing (DSP) slices. Table III compares the existing and proposed methods regarding the hardware used and the

TABLE I
RMS ERROR OF CURVE FIT OPTIMIZATION RESULTS

Averaged scans	NLLSF (RMS error)	without optimization	with optimization	Pearson Correlation Coefficient
8	1.79e-04	2.90e-04	2.31e-04	0.9934
16	1.88e-04	12.0e-04	3.61e-04	0.9444
32	0.90e-04	5.56e-04	1.04e-04	0.9989
64	0.96e-04	2.99e04	1.46e-04	0.9978
128	0.60e-04	1.19e-04	0.84e-04	0.9992
256	1.06e-04	2.28e-04	1.50e-04	0.9981
512	0.93e-04	2.42e-04	1.50e-04	0.9982
1024	0.85e-04	4.88e-04	1.28e-04	0.9987

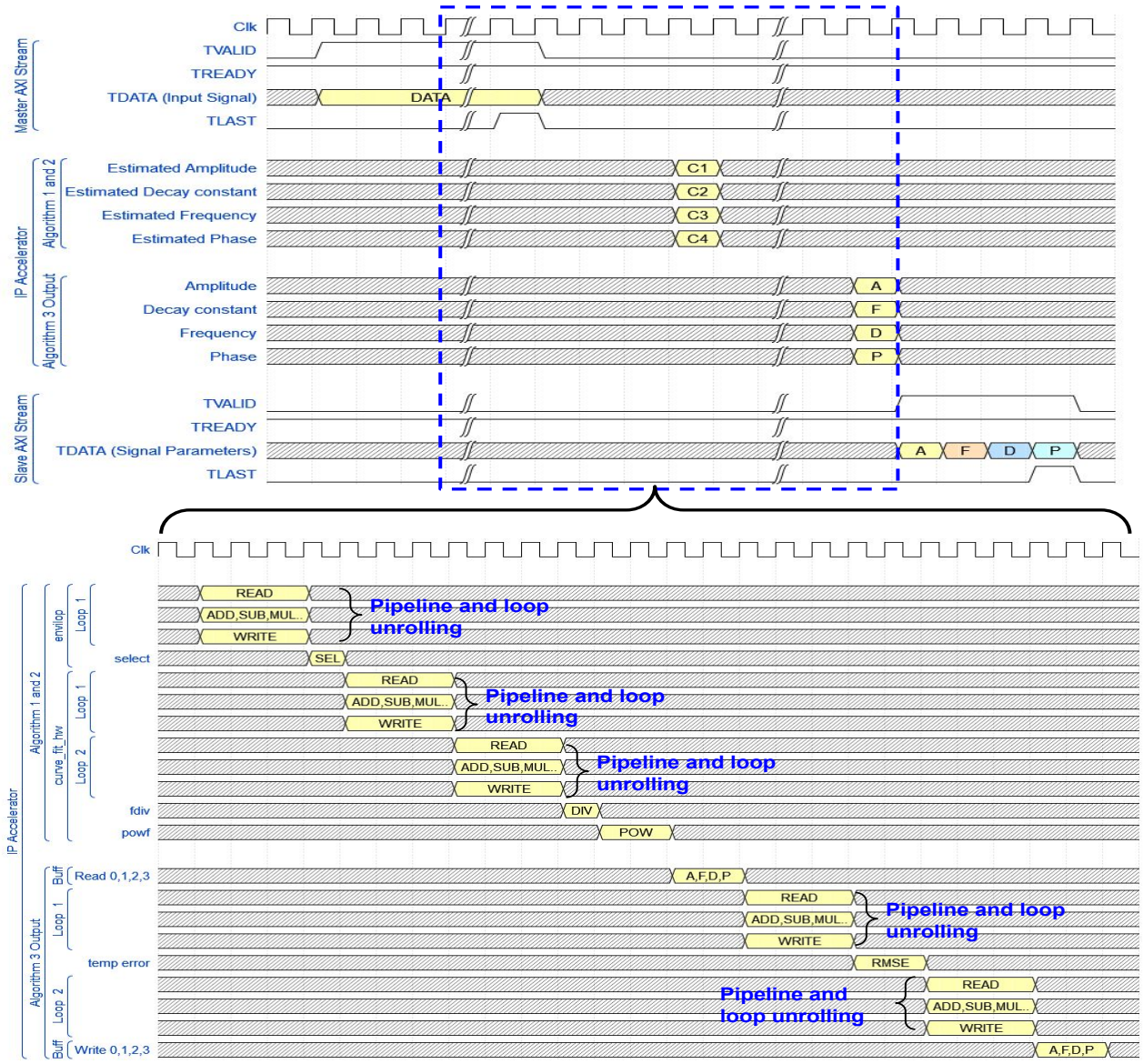

Fig. 4. Implementation timing diagram.

TABLE II
RESOURCE UTILIZATION OF THE PROPOSED CURVE FITTING METHOD ON PYNQ-Z2 FPGA

Averaged scans	Floating point on PL (Sequential)	Floating point on PL (Pipeline)	Processing system
BRAM_18K (%)	42.14	45	-
DSP48E (%)	45.45	74.55	-
FF (%)	18.46	27.80	-
LUT (%)	32.95	56.73	-
FOM*	299796	481720	-
Static Power (Watts)	0.15	0.15	-
Dynamic Power (Watts)	1.69	1.89	-
Execution time (Sec)	2.03	0.12	824.77

*Figure Of Merit (FOM) refer to (13).

time consumed to implement the proposed design. The Non-Linear Least-Squares (NLLSF) method consumes 60 us to perform the cure fitting, and the signal length is unknown; however, as discussed in the introduction, the gradient base NLLSF is not a data-driven approach to perform curve fitting since it requires optimal initialisation guess of parameters, which is not ideal for all the applications. On the other hand, the proposed method provides a data-driven and resource-inexpensive approach that fits any application.

VI. Conclusion

Overall, this work introduces an optimal solution to extract the signal parameters of the exponential decay signal. The residual error between the actual and estimated signal is in the order of 10^{-4} to 10^{-5}, which is similar to the standard nonlinear least-square fit approach. With the help of this approach, we

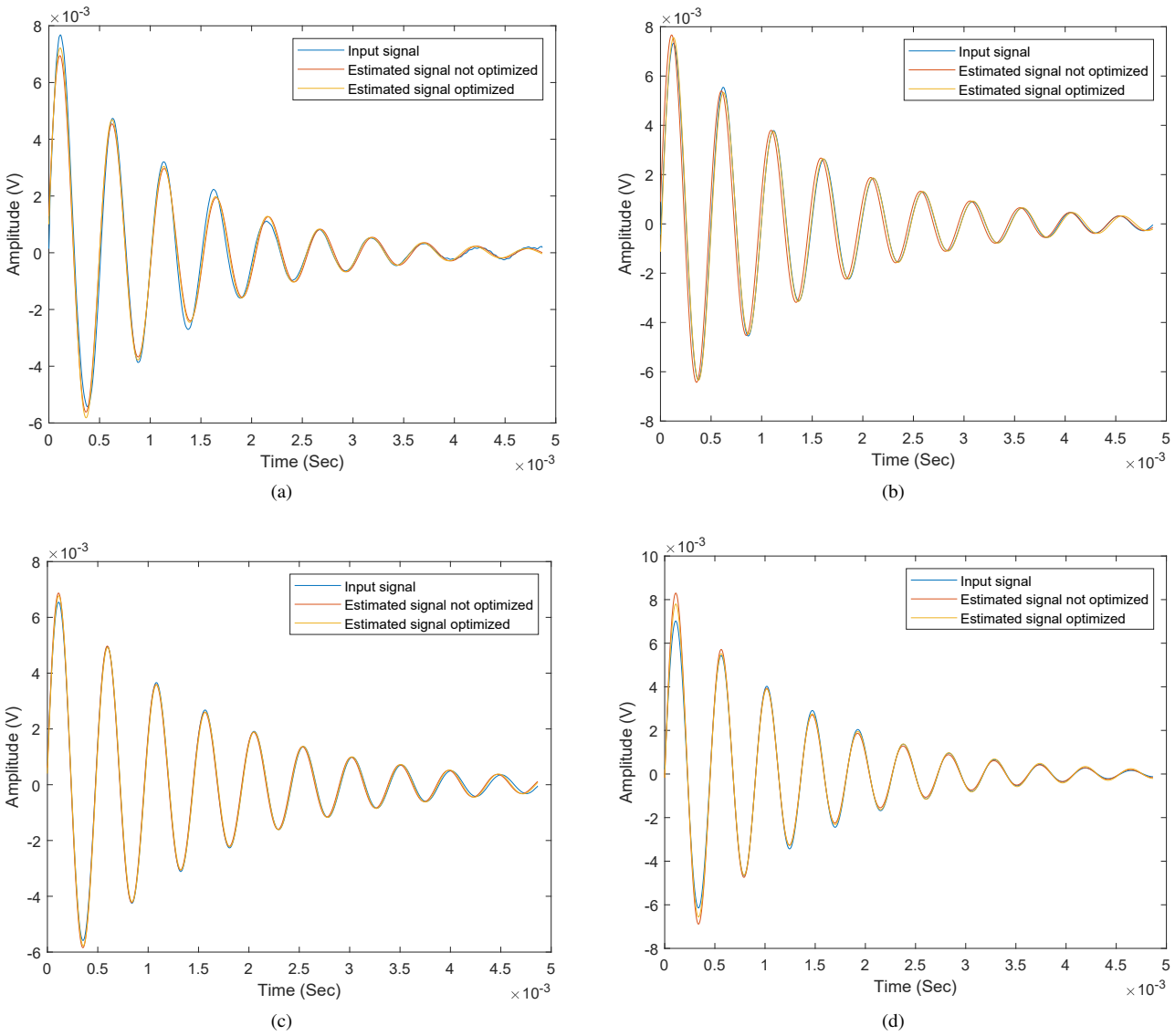

Fig. 5. Proposed curve fit approach on different FID signals (a) 8 $NaNO_2$ FID scans averaged data; (b) 32 $NaNO_2$ FID scans averaged data; (c) 128 $NaNO_2$ FID scans averaged data; (d) 512 $NaNO_2$ FID scans averaged data.

can calculate all four essential parameters of a large signal under 0.12 sec. The proposed method overcomes multiple bottlenecks in implementing parameter estimation algorithms on FPGA in terms of resources; apart from that, it is a data-driven approach where the initial guess is not required.

TABLE III
COMPARISON BETWEEN EXISTING AND PROPOSED METHOD

Work	Technology	Signal length (samples)	Execution time (sec)
[5] Non-Linear Least-Squares (NLLSF)	Virtex-5(1Mhz)	-	60us
[3] parameter estimation using Auto-encoding	GPU	1000	-
[Proposed work] SSA on NQR signal	PYNQ-Z2 (100MHz)	4868	0.12

REFERENCES

[1] M. Osborne, "Nonlinear least squares: The Levenberg algorithm revisited," *J. Australian Math. Soc.*, vol. 19, pp. 343-357, 1976

[2] L.H. Iselin, "An analytical method for estimating the 14n nuclear quadrupole resonance parameters of organic compounds with complex free induction decays for radiation effects studies," Tech. Rep., Florida Univ., TN, US, 1992

[3] J.C. Visschers, D. Budker, and L. Bougas, "Rapid parameter estimation of discrete decaying signals using autoencoder networks," *Mach. Learn.: Sci. Technol.*, vol. 2, no. 4, pp. 045024 (1-16), Sep. 2021

[4] G. Bostrom, D. Atkinson, and A. Rice, "The discrete Fourier transform algorithm for determining decay constants—Implementation using a field programmable gate array," *Rev. Sci. Instrum.*, vol. 86, no. 4, pp. 043106, Apr. 2015.

[5] A. Abba, F. Caponio, A. Geraci and G. Ripamonti, "Implementation of high efficiency non-linear least-squares in FPGA devices for digital

spectroscopy," in *Proc. IEEE Nucl. Sci. Symp. Med. Imaging Conf.*, pp. 1371-1376, 2010.

[6] K.T. Varma, R. Peesapati, C.V.R. Rao, A. K. Rajarajan, M. Dixit, and G. Joshi, "A Novel SSA-NLLSF Approach for Denoising NQR Signals," *Appl. Magn. Reson.*, vol. 52, no. 11, pp. 1601–1613, Aug. 2021

[7] K. Wang, J. Chiasson, M. Bodson and L. M. Tolbert, "A nonlinear least-squares approach for identification of the induction motor parameters," in *Proc. IEEE Conf. Decision Control (CDC).*, pp. 3856-3861 Vol.4, 2004,

[8] W.M. Kolb, "*Curve Fitting for Programmable Calculators,*" 3rd ed. Bowie, Maryland: Syntec Inc, 1984.

[9] C. Ridders, "Three-point iterations derived from exponential curve fitting," *IEEE Trans. Circuits Syst.*, vol. 26, no. 8, pp. 669–670, Aug. 1979.

[10] A.T. Conn, K. Scheinberg, and L. Vicente, "Introduction to Derivative-Free Optimization," SIAM. 2009.

[11] K.V.S. Kumar, V.R. Kopparthi, S.L. Sabat, K.T. Varma, and R. Peesapati, "System on chip implementation of floating point matrix inversion using modified Gram-Schmidt based QR decomposition on PYNQ FPGA," in *Proc.IEEE Int. Symposium on Smart Electronic Systems (iSES).*, pp. 84-88, 2021.

[12] C. Monea, "Improved Signal Models for Nuclear Quadrupole Resonance Spectroscopy," in *Proc. Int. Conf. Electronics, Computers Artificial Intell. (ECAI).*, pp. 1-4, 2020.

Preparation of Papers for Electronics (September 2011)

First A. Author, Second B. Author, and Third C. Author

***Abstract*—These instructions give you guidelines for preparing papers for ELECTRONICS journal. Use this document as a template if you are using Microsoft *Word* 6.0 or later. Otherwise, use this document as an instruction set. The electronic file of your paper will be formatted further. Define all symbols used in the abstract. Do not cite references in the abstract. Do not delete the blank line immediately above the abstract; it sets the footnote at the bottom of this column.**

***Index Terms*—About four key words or phrases in alphabetical order, separated by commas.**

Paper Classification
DOI: 10.7251/ELSxxxxxxxx

I. Introduction

THIS document is a template for Microsoft *Word* versions 6.0 or later.

When you open the file, select "Page Layout" from the "View" menu in the menu bar (View | Page Layout), which allows you to see the footnotes. Then, type over sections of file or cut and paste from another document and use markup styles. The pull-down style menu is at the left of the Formatting Toolbar at the top of your *Word* window (for example, the style at this point in the document is "Text"). Highlight a section that you want to designate with a certain style, then select the appropriate name on the style menu. The style will adjust your fonts and line spacing. **Do not change the font sizes or line spacing to squeeze more text into a limited number of pages.** Use italics for emphasis; do not underline. The length of the manuscript is limited to the maximum of 15 pages.

To insert images in *Word,* position the cursor at the insertion point and either use Insert | Picture | From File or copy the image to the Windows clipboard and then Edit | Paste Special | Picture (with "float over text" unchecked).

We will do the final formatting of your paper.

II. Procedure for Paper Submission

A. Review Stage

The manuscripts are to be submitted using the Electronics Journal online submission system – accessible from Journal's homepage. Prepare it in two-column format as shown in this template. Place all figures and tables at the end of the paper (after the references) on separate page(s). Figures and tables must have the same caption names as referenced in the text. Only PDF format of the manuscript is allowed at the review stage. Please, check if all fonts are embedded and subset and that the quality of diagrams, illustrations, and graphics is satisfactory. Failing to provide above listed requirements is a valid reason for rejection.

B. Final Stage

When you submit your final version (after your paper has been accepted), prepare it in two-column format, including figures and tables in accordance with this template. Pack all of your files (manuscript source file in *Word*, figures, and manuscript PDF form) within one archive file (you may use any of the available file compression tools: *WinZip*, *WinRAR*, *7-Zip*, etc.). Do not forget to provide the manuscript converted in PDF format that will be used as a reference for final formatting of your paper. Figures should be named as referenced in the manuscript (e.g. *fig1.eps*, *fig2.tif*, etc.)

C. Figures and Tables

Format and save your graphic images using a suitable graphics processing program and adjusts the resolution settings. We accept images in the following formats: PS, EPS, TIFF, GIF, and PNG. Additionally, it is allowed to use images generated by using one of the following software tools: Microsoft Word, Microsoft PowerPoint, or Microsoft Excel. The resolution of a RGB color file should be 400 dpi. Please note that JPG and other lossy-compressed image formats are not allowed. Use available software tools to convert these images to appropriate format.

Image quality is very important to how yours graphics will reproduce. Even though we can accept graphics in many formats, we cannot improve your graphics if they are poor quality when we receive them. If your graphic looks low in

Manuscript received 15 September 2011 (write the date when you have first sent the manuscript for review). Received in revised form 20 October 2011 (write the date when you have sent the manuscript in its revised form if revisions required for your paper after review).

(Place here any sponsor and financial support acknowledgments).

F. A. Author is with the Faculty of Electrical Engineering, University of Banja Luka, Banja Luka, Bosnia and Herzegovina (corresponding author to provide phone: +387-51-222-333; fax: +387-51-111-222; e-mail: author@et-fbl.net).

S. B. Author was with Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia. He is now with the Institute "Mihailo Pupin", Belgrade, Serbia (e-mail: author@pupin.rs).

T. C. Author is with the School of Electrical Engineering, University of Belgrade, Belgrade, Serbia, on leave from the Faculty of Electronic Engineering, University of Niš, Niš, Serbia (e-mail: author@elfak.ni.ac.rs).

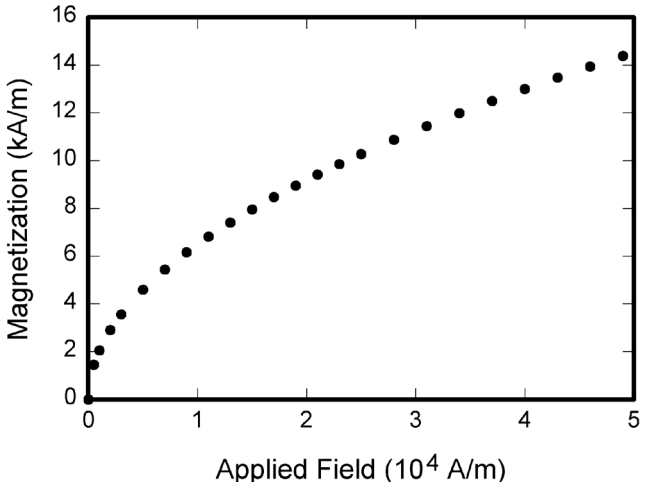

Fig. 1. Magnetization as a function of applied field. Note that "Fig." is abbreviated. There is a period after the figure number, followed by two spaces. It is good practice to explain the significance of the figure in the caption.

quality on your printer or monitor, please keep in mind that cannot improve the quality after submission.

If you are importing your graphics into this Word template, please use the following steps:

Under the option EDIT select PASTE SPECIAL. A dialog box will open, select paste picture, then click OK. Your figure should now be in the Word Document.

If you are preparing images in TIFF, EPS, or PS format, note the following. High-contrast line figures and tables should be prepared with 600 dpi resolution and saved with no compression, 1 bit per pixel (monochrome).

Photographs and grayscale figures should be prepared with 300 dpi resolution and saved with no compression, 8 bits per pixel (grayscale).

Most charts graphs and tables are one column wide (3 1/2 inches or 21 picas) or two-column width (7 1/16 inches, 43 picas wide). We recommend that you avoid sizing figures less than one column wide, as extreme enlargements may distort your images and result in poor reproduction. Therefore, it is better if the image is slightly larger, as a minor reduction in size should not have an adverse affect the quality of the image.

III. Math

If you are using *Word,* use either the Microsoft Equation Editor or the *MathType* add-on (http://www.mathtype.com) for equations in your paper (Insert | Object | Create New | Microsoft Equation *or* MathType Equation). "Float over text" should *not* be selected.

IV. Units

Use either SI (MKS) or CGS as primary units. (SI units are strongly encouraged.) English units may be used as secondary units (in parentheses). **This applies to papers in data storage.** For example, write "15 Gb/cm^2 (100 Gb/in^2)." An exception is when English units are used as identifiers in trade, such as "3½-in disk drive." Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity in an equation.

TABLE I
Units for Magnetic Properties

Symbol	Quantity	Conversion from Gaussian and CGS EMU to SI [a]
Φ	magnetic flux	1 Mx $\rightarrow 10^{-8}$ Wb = 10^{-8} V·s
B	magnetic flux density, magnetic induction	1 G $\rightarrow 10^{-4}$ T = 10^{-4} Wb/m^2
H	magnetic field strength	1 Oe $\rightarrow 10^3/(4\pi)$ A/m
m	magnetic moment	1 erg/G = 1 emu $\rightarrow 10^{-3}$ A·m^2 = 10^{-3} J/T
M	magnetization	1 erg/(G·cm^3) = 1 emu/cm^3 $\rightarrow 10^3$ A/m
$4\pi M$	magnetization	1 G $\rightarrow 10^3/(4\pi)$ A/m
σ	specific magnetization	1 erg/(G·g) = 1 emu/g $\rightarrow$ 1 A·m^2/kg
j	magnetic dipole moment	1 erg/G = 1 emu $\rightarrow 4\pi \times 10^{-10}$ Wb·m
J	magnetic polarization	1 erg/(G·cm^3) = 1 emu/cm^3 $\rightarrow 4\pi \times 10^{-4}$ T
χ, κ	susceptibility	1 $\rightarrow 4\pi$
χ_ρ	mass susceptibility	1 cm^3/g $\rightarrow 4\pi \times 10^{-3}$ m^3/kg
μ	permeability	1 $\rightarrow 4\pi \times 10^{-7}$ H/m = $4\pi \times 10^{-7}$ Wb/(A·m)
μ_r	relative permeability	$\mu \rightarrow \mu_r$
w, W	energy density	1 erg/cm^3 $\rightarrow 10^{-1}$ J/m^3
N, D	demagnetizing factor	1 $\rightarrow 1/(4\pi)$

Vertical lines are optional in tables. Statements that serve as captions for the entire table do not need footnote letters.

[a]Gaussian units are the same as cgs emu for magnetostatics; Mx = maxwell, G = gauss, Oe = oersted; Wb = weber, V = volt, s = second, T = tesla, m = meter, A = ampere, J = joule, kg = kilogram, H = henry.

The SI unit for magnetic field strength H is A/m. However, if you wish to use units of T, either refer to magnetic flux density B or magnetic field strength symbolized as $\mu_0 H$. Use the center dot to separate compound units, e.g., "A·m^2."

V. Helpful Hints

A. Figures and Tables

Because we will do the final formatting of your paper, you do not need to position figures and tables at the top and bottom of each column. In fact, all figures, figure captions, and tables can be at the end of the paper. Large figures and tables may span both columns. Place figure captions below the figures; place table titles above the tables. If your figure has two parts, include the labels "(a)" and "(b)" as part of the artwork. Please verify that the figures and tables you mention in the text actually exist. **Please do not include captions as part of the figures. Do not put captions in "text boxes" linked to the figures. Do not put borders around the outside of your figures.** Use the abbreviation "Fig." even at the beginning of a sentence. Do not abbreviate "Table." Tables are numbered with Roman numerals.

Color printing of figures is not available Do not use color unless it is necessary for the proper interpretation of your figures.

Figure axis labels are often a source of confusion. Use words rather than symbols. As an example, write the quantity "Magnetization," or "Magnetization *M*," not just "*M*." Put units in parentheses. Do not label axes only with units. As in Fig. 1, for example, write "Magnetization (A/m)" or "Magnetization ($A \cdot m^{-1}$)," not just "A/m." Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)," not "Temperature/K."

Multipliers can be especially confusing. Write "Magnetization (kA/m)" or "Magnetization (10^3 A/m)." Do not write "Magnetization (A/m) x 1000" because the reader would not know whether the top axis label in Fig. 1 meant 16000 A/m or 0.016 A/m. Figure labels should be legible, approximately 8 to 12 point type.

B. References

Number citations consecutively in square brackets [1]. The sentence punctuation follows the brackets [2]. Multiple references [2], [3] are each numbered with separate brackets [1]–[3]. When citing a section in a book, please give the relevant page numbers [2]. In sentences, refer simply to the reference number, as in [3]. Do not use "Ref. [3]" or "reference [3]" except at the beginning of a sentence: "Reference [3] shows ... ." Please do not use automatic endnotes in *Word*, rather, type the reference list at the end of the paper using the "References" style.

Number footnotes separately in superscripts (Insert | Footnote).[1] Place the actual footnote at the bottom of the column in which it is cited; do not put footnotes in the reference list (endnotes). Use letters for table footnotes (see Table I).

Please note that the references at the end of this document are in the preferred referencing style. Give all authors' names; do not use "*et al*." unless there are six authors or more. Use a space after authors' initials. Papers that have not been published should be cited as "unpublished" [4]. Papers that have been accepted for publication, but not yet specified for an issue should be cited as "to be published" [5]. Papers that have been submitted for publication should be cited as "submitted for publication" [6]. Please give affiliations and addresses for private communications [7].

Capitalize only the first word in a paper title, except for proper nouns and element symbols. For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [8]. All references **must be** written in Roman alphabet.

C. Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have already been defined in the abstract. Abbreviations such as IEEE, SI, ac, and dc do not have to be defined. Abbreviations that incorporate periods should not have spaces: write "C.N.R.S.," not "C. N. R. S." Do not use abbreviations in the title unless they are unavoidable (for example, "IEEE" in the title of this article).

D. Equations

Number equations consecutively with equation numbers in parentheses flush with the right margin, as in (1). First use the equation editor to create the equation. Then select the "Equation" markup style. Press the tab key and write the equation number in parentheses. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Use parentheses to avoid ambiguities in denominators. Punctuate equations when they are part of a sentence, as in

$$\int_0^{r_2} F(r,\varphi)\,\mathbf{d}\,d\varphi = [\sigma r_2/(2\mu_0)] \cdot \int_0^{\infty} \exp(-\lambda\,|z_j - z_i|)\,\lambda^{-1} J_1(\lambda r_2)\,J_0(\lambda r_i)\,d\lambda\,. \tag{1}$$

Be sure that the symbols in your equation have been defined before the equation appears or immediately following. Italicize symbols (*T* might refer to temperature, but T is the unit tesla). Refer to "(1)," not "Eq. (1)" or "equation (1)," except at the beginning of a sentence: "Equation (1) is ... ."

E. Other Recommendations

Use one space after periods and colons. Hyphenate complex modifiers: "zero-field-cooled magnetization." Avoid dangling participles, such as, "Using (1), the potential was calculated." [It is not clear who or what used (1).] Write instead, "The potential was calculated by using (1)," or "Using (1), we calculated the potential."

Use a zero before decimal points: "0.25," not ".25." Use "cm^3," not "cc." Indicate sample dimensions as "0.1 cm x 0.2 cm," not "0.1 x 0.2 cm^2." The abbreviation for "seconds" is "s," not "sec." Do not mix complete spellings and abbreviations of units: use "Wb/m^2" or "webers per square meter," not "webers/m^2." When expressing a range of values, write "7 to 9" or "7-9," not "7~9."

A parenthetical statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.) In American English, periods and commas are within quotation marks, like "this period." Other punctuation is "outside"! Avoid contractions; for example, write "do not" instead of "don't." The serial comma is preferred: "A, B, and C" instead of "A, B and C."

If you wish, you may write in the first person singular or plural and use the active voice ("I observed that ..." or "We observed that ..." instead of "It was observed that ..."). Remember to check spelling. If your native language is not English, please get a native English-speaking colleague to carefully proofread your paper.

[1] It is recommended that footnotes be avoided (except for the unnumbered footnote with the receipt date and authors' affiliations on the first page). Instead, try to integrate the footnote information into the text.

VI. Some Common Mistakes

The word "data" is plural, not singular. The subscript for the permeability of vacuum μ_0 is zero, not a lowercase letter "o." The term for residual magnetization is "remanence"; the adjective is "remanent"; do not write "remnance" or "remnant." Use the word "micrometer" instead of "micron." A graph within a graph is an "inset," not an "insert." The word "alternatively" is preferred to the word "alternately" (unless you really mean something that alternates). Use the word "whereas" instead of "while" (unless you are referring to simultaneous events). Do not use the word "essentially" to mean "approximately" or "effectively." Do not use the word "issue" as a euphemism for "problem." When compositions are not specified, separate chemical symbols by en-dashes; for example, "NiMn" indicates the intermetallic compound $Ni_{0.5}Mn_{0.5}$ whereas "Ni–Mn" indicates an alloy of some composition Ni_xMn_{1-x}.

Be aware of the different meanings of the homophones "affect" (usually a verb) and "effect" (usually a noun), "complement" and "compliment," "discreet" and "discrete," "principal" (e.g., "principal investigator") and "principle" (e.g., "principle of measurement"). Do not confuse "imply" and "infer."

Prefixes such as "non," "sub," "micro," "multi," and "ultra" are not independent words; they should be joined to the words they modify, usually without a hyphen. There is no period after the "et" in the Latin abbreviation "*et al.*" (it is also italicized). The abbreviation "i.e.," means "that is," and the abbreviation "e.g.," means "for example" (these abbreviations are not italicized).

An excellent style manual and source of information for science writers is [9].

VII. Editorial Policy

Each manuscript submitted is subjected to the following review procedure:

- It is reviewed by the editor for general suitability for this publication
- If it is judged suitable, two reviewers are selected and a single-blinded review process takes place
- Based on the recommendations of the reviewers, the editor then decides whether the particular paper should be accepted as is, revised or rejected.

Do not submit a paper you have submitted or published elsewhere. Do not publish "preliminary" data or results. The submitting author is responsible for obtaining agreement of all coauthors and any consent required from sponsors before submitting a paper. It is the obligation of the authors to cite relevant prior work.

Every paper submitted to "Electronics" journal are single-blind reviewed. For conference-related papers, the decision to accept or reject a paper is made by the conference editors and publications committee; the recommendations of the referees are advisory only. Undecipherable English is a valid reason for rejection.

VIII. Publication Principles

The contents of "Electronics" are peer-reviewed and archival. The "Electronics" publishes scholarly articles of archival value as well as tutorial expositions and critical reviews of classical subjects and topics of current interest.

Authors should consider the following points:

1) Technical papers submitted for publication must advance the state of knowledge and must cite relevant prior work.
2) The length of a submitted paper should be commensurate with the importance, or appropriate to the complexity, of the work. For example, an obvious extension of previously published work might not be appropriate for publication or might be adequately treated in just a few pages.
3) Authors must convince both peer reviewers and the editors of the scientific and technical merit of a paper; the standards of proof are higher when extraordinary or unexpected results are reported.
4) Because replication is required for scientific progress, papers submitted for publication must provide sufficient information to allow readers to perform similar experiments or calculations and use the reported results. Although not everything need be disclosed, a paper must contain new, useable, and fully described information. For example, a specimen's chemical composition need not be reported if the main purpose of a paper is to introduce a new measurement technique. Authors should expect to be challenged by reviewers if the results are not supported by adequate data and critical details.
5) Papers that describe ongoing work or announce the latest technical achievement, which are suitable for presentation at a professional conference, may not be appropriate for publication in "Electronics".

IX. Conclusion

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

Appendix

Appendixes, if needed, appear before the acknowledgment.

Acknowledgment

The preferred spelling of the word "acknowledgment" in American English is without an "e" after the "g." Use the singular heading even if you have many acknowledgments. Avoid expressions such as "One of us (S.B.A.) would like to thank ... ." Instead, write "F. A. Author thanks ... ." **Sponsor and financial support acknowledgments are placed in the unnumbered footnote on the first page, not here.**

References

[1] G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.

[2] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[3] H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.

[4] B. Smith, "An approach to graphs of linear forms (Unpublished work style)," unpublished.

[5] E. H. Miller, "A note on reflector arrays (Periodical style—Accepted for publication)," *IEEE Trans. Antennas Propagat.*, to be published.

[6] J. Wang, "Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication)," *IEEE J. Quantum Electron.*, submitted for publication.

[7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[8] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces (Translation Journals style)," *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740–741 [*Dig. 9th Annu. Conf. Magnetics* Japan, 1982, p. 301].

[9] M. Young, *The Techincal Writers Handbook.* Mill Valley, CA: University Science, 1989.

[10] J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility (Periodical style)," *IEEE Trans. Electron Devices*, vol. ED-11, pp. 34–39, Jan. 1959.

[11] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 570–578, Jul. 1993.

[12] R. W. Lucky, "Automatic equalization for digital communication," *Bell Syst. Tech. J.*, vol. 44, no. 4, pp. 547–588, Apr. 1965.

[13] S. P. Bingulac, "On the compatibility of adaptive controllers (Published Conference Proceedings style)," in *Proc. 4th Annu. Allerton Conf. Circuits and Systems Theory*, New York, 1994, pp. 8–16.

[14] G. R. Faulhaber, "Design of service systems with priority reservation," in *Conf. Rec. 1995 IEEE Int. Conf. Communications,* pp. 3–8.

[15] W. D. Doyle, "Magnetization reversal in films with biaxial anisotropy," in *1987 Proc. INTERMAG Conf.*, pp. 2.2-1–2.2-6.

[16] G. W. Juette and L. E. Zeffanella, "Radio noise currents n short sections on bundle conductors (Presented Conference Paper style)," presented at the IEEE Summer power Meeting, Dallas, TX, Jun. 22–27, 1990, Paper 90 SM 690-0 PWRS.

[17] J. G. Kreifeldt, "An analysis of surface-detected EMG as an amplitude-modulated noise," presented at the 1989 Int. Conf. Medicine and Biological Engineering, Chicago, IL.

[18] J. Williams, "Narrow-band analyzer (Thesis or Dissertation style)," Ph.D. dissertation, Dept. Elect. Eng., Harvard Univ., Cambridge, MA, 1993.

[19] N. Kawasaki, "Parametric study of thermal and chemical nonequilibrium nozzle flow," M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.

[20] J. P. Wilkinson, "Nonlinear resonant circuit devices (Patent style)," U.S. Patent 3 624 12, July 16, 1990.

[21] IEEE Criteria for Class IE Electric Systems (Standards style), IEEE Standard 308, 1969.

[22] Letter Symbols for Quantities, ANSI Standard Y10.5-1968.

[23] R. E. Haskell and C. T. Case, "Transient signal propagation in lossless isotropic plasmas (Report style)," USAF Cambridge Res. Lab., Cambridge, MA Rep. ARCRL-66-234 (II), 1994, vol. 2.

[24] E. E. Reber, R. L. Michell, and C. J. Carter, "Oxygen absorption in the Earth's atmosphere," Aerospace Corp., Los Angeles, CA, Tech. Rep. TR-0200 (420-46)-3, Nov. 1988.

[25] (Handbook style) *Transmission Systems for Communications,* 3rd ed., Western Electric Co., Winston-Salem, NC, 1985, pp. 44–60.

[26] *Motorola Semiconductor Data Manual,* Motorola Semiconductor Products Inc., Phoenix, AZ, 1989.

[27] (Basic Book/Monograph Online Sources) J. K. Author. (year, month, day). *Title* (edition) [Type of medium]. Volume (issue). Available: http://www.(URL)

[28] J. Jones. (1991, May 10). Networks (2nd ed.) [Online]. Available: http://www.atm.com

[29] (Journal Online Sources style) K. Author. (year, month). Title. *Journal* [Type of medium]. Volume(issue), paging if given. Available: http://www.(URL)

[30] R. J. Vidmar. (1992, August). On the use of atmospheric plasmas as electromagnetic reflectors. *IEEE Trans. Plasma Sci.* [Online]. *21(3).* pp. 876–880. Available: http://www.halcyon.com/pub/journals/21ps03-vidmar

Information for Authors

Editorial objectives

In the journal ”Electronics”, the scientific papers from different fields of electronics and electrical engineering in the broadest sense are published. Main topics are electronics, automatics, telecommunications, computer techniques, power engineering, nuclear and medical electronics, analysis and synthesis of electronic circuits and systems, new technologies and materials in electronics etc. The main emphasis of papers should be on methods and new techniques, or the application of existing techniques in a novel way.

The reviewing process

Each manuscript submitted is subjected to the following review procedures:

- It is reviewed by the editor for general suitability for this publication;
- If it is judged suitable, two reviewers are selected and a double-blind review process takes place;
- Based on the recommendations of the reviewers, the editor then decides whether the particular paper should be accepted as it is, revised or rejected.

Submissions Process

The manuscripts are to be submitted using the Electronics Journal online submission system – accessible from Journal’s homepage.

Manuscripts have to be prepared in accordance with the instructions given in the template for paper preparation that can be found on the journal’s web page (www.els-journal.etf.unibl.org).

Authors should note that proofs are not supplied prior to publication and ensure that the paper submitted is complete and in its final form.

Copyright

Articles submitted to the journal should be original contributions and should not be under consideration for any other publication at the same time. Authors submitting articles for publication warrant that the work is not an infringement of any existing copyright and will indemnify the publisher against any breach of such warranty. For ease of dissemination and to ensure proper policing of use, papers and contributions become the legal copyright of the publisher unless otherwise agreed.