



**FACULTY OF ELECTRICAL ENGINEERING  
UNIVERSITY OF BANJA LUKA**

# **ELECTRONICS**

**VOLUME 21, NUMBER 1, JUNE 2017**

# FACULTY OF ELECTRICAL ENGINEERING UNIVERSITY OF BANJA LUKA

Address: Patre 5, 78000 Banja Luka, Bosnia and Herzegovina

Phone: +387 51 211824

Fax: +387 51 211408

Web: [www.etf.unibl.org](http://www.etf.unibl.org)

## ELECTRONICS

Web: [www.electronics.etfbl.net](http://www.electronics.etfbl.net)

E-mail: [els-journal@etf.unibl.org](mailto:els-journal@etf.unibl.org)

### *Editor-in-Chief:*

Branko L. Dokić, Ph. D.

Faculty of Electrical Engineering, University of Banja Luka, Bosnia and Herzegovina

E-mail: [branko.dokic@etf.unibl.org](mailto:branko.dokic@etf.unibl.org)

### *Co Editor-In-Chief:*

Prof. Tatjana Pešić-Brdjanin, University of Banja Luka, Bosnia and Herzegovina

E-mail: [tatjana.pesic-brdjanin@etf.unibl.org](mailto:tatjana.pesic-brdjanin@etf.unibl.org)

### *International Editorial Board:*

- Prof. Goce Arsov, St. Cyril and Methodius University, Macedonia
- Prof. Zdenka Babić, University of Banja Luka, Bosnia and Herzegovina
- Prof. Petar Biljanović, University of Zagreb, Croatia
- Prof. Milorad Božić, University of Banja Luka, Bosnia and Herzegovina
- Prof. Octavio Nieto-Taladriz Garcia, Polytechnic University of Madrid, Spain
- Dr Zoran Jakšić, IHTM, Serbia
- Prof. Vladimir Katić, University of Novi Sad, Serbia
- Prof. Tom J. Kazmierski, University of Southampton, United Kingdom
- Prof. Vančo Litovski, University of Niš, Serbia
- Dr Duško Lukač, University of Applied Sciences, Germany
- Prof. Danilo Mandić, Imperial College, London, United Kingdom
- Prof. Bratislav Milovanović, University of Niš, Serbia
- Prof. Vojin Oklobdžija, University of Texas at Austin, USA
- Prof. Predrag Pejović, University of Belgrade, Serbia
- Prof. Ninoslav Stojadinović, University of Niš, Serbia
- Prof. Robert Šobot, Western University, Canada
- Prof. Slobodan Vukosavić, University of Belgrade, Serbia
- Prof. Volker Zerbe, University of Applied Sciences of Erfurt, Germany

### *Secretary:*

Aleksandar Pajkanović, M.Sc.

Drago Čavka

Svjetlana Kovačević

Jovica Bulović

### *Publisher:*

Faculty of Electrical Engineering, University of Banja Luka, Bosnia and Herzegovina

Number of printed copies: 100





# LUT Based Generalized Parallel Counters for State-of-art FPGAs

Burhan Khurshid

**Abstract-** Generalized Parallel Counters (GPCs) are frequently used in constructing high speed compressor trees. Previous work has focused on achieving efficient mapping of GPCs on FPGAs by using a combination of general Look-up table (LUT) fabric and specialized fast carry chains. The resulting structures are purely combinational and cannot be efficiently pipelined to achieve the potential FPGA performance. In this paper, we take an alternate approach and try to eliminate the fast carry chain from the GPC structure. We present a heuristic that maps GPCs on FPGAs using only general LUT fabric. The resultant GPCs are then easily re-timed by placing registers at the fan-out nodes of each LUT. We have used our heuristic on various GPCs reported in prior work. Our heuristic successfully eliminates the carry chain from the GPC structure with the same LUT count in most of the cases. Experimental results using Xilinx Kintex-7 FPGAs show a considerable reduction in critical path and dynamic power dissipation with same area utilization in most of the cases.

**Index Terms -** Look-up table, Compressor trees, Technology mapping, Retiming

Original Research Paper

DOI: 10.7251/ELS1721003K

## I. INTRODUCTION

Multi-operand addition is an important operation in many arithmetic circuits. It is frequently used in many applications like filtering [1], motion estimation [2], array multiplication [3, 4, 5, 6, 7] etc. Compressor trees form the basic elements in multi-operand additions. Compressor trees based on carry save adders (CSA) typically provide higher speeds due to the avoidance of long carry chains. Wallace [3] and Dadda [7] trees are CSA based compressor trees which are frequently used in application specific integrated circuit (ASIC) design. However, the introduction of fast carry chains in FPGAs has made ripple carry addition faster than the carry save addition. Evidently CSA based compressor trees are not well suited for implementation involving FPGAs [8].

Prior work on compressor tree synthesis using FPGAs has used GPCs as basic constituent element. It has been demonstrated that the usage of GPCs can lead to a considerable reduction in the critical path delay with

comparable resource utilization [8, 9, 10, 11, 12, 13, 14]. Initial attempts in this regard were made by Parandeh-Afshar et al. [8, 9, 10, 11]. In [9] they claim to report the first method that synthesizes compressor trees on FPGAs. The proposed heuristic constructs compressor trees from a library of GPCs that can be efficiently implemented on FPGAs. Their latter work [11] focuses on further reducing the combinational delay and any increase in area by formulating the mapping of GPCs as an integer linear programming (ILP) problem. They reported an average reduction in delay by 32% and area by 3% when compared to an adder tree. In [10] they focus on reducing the combinational delay by using embedded fast carry chains. This concept was further extended in [8] and a delay reduction of 33% and 45% was achieved in Xilinx Virtex-5 and Altera Stratix-III FPGAs respectively.

Matsunaga et al. [12, 14] also formulated the mapping of GPCs as an ILP with speed and power as optimization goals. Their results show a 28% reduction in GPC count when compared to [9]. A reduction in GPC count results in reduction of compression stages thereby reducing the delay and power consumption.

Recent attempts from Kumm and Zipf [15, 16] focus on exploiting the low-level structure of Xilinx FPGAs to develop novel GPCs with high compression ratios and efficient resource utilization. Both general purpose LUT fabric and specialized carry chains have been used for synthesizing resource-efficient delay-optimal GPCs.

All the above mentioned approaches (except [9]) focus on exploiting the fast carry chain embedded in modern FPGAs. The idea is to use the fast carry chain to connect the adjacent logic cells and by pass the programmable routing network to reduce delay [10]. In this paper, however, we try to avoid the usage of embedded carry chains and propose a heuristic that tries to implement GPCs using only the general LUT fabric. The heuristic tries to minimize the number of LUTs in a GPC. The area-optimized GPCs are then easily retimed by inserting registers that are available in each logic cell. Thus instead of using an LUT-carry chain combination we use an LUT-register combination to map the GPCs. The motivation for our approach is backed by following reasons:

- i. GPCs based on LUTs and carry chains are purely combinational in nature. FPGAs are synchronous devices and it is better to adhere to synchronous practices while using them as implementation platforms. Our approach provides this synchronous description by including registers in the synthesis process.

Manuscript received 15 June 2015. Received in revised form 5 May 2017. Accepted for publication 2 June 2017.

Burhan Khurshid is with the Department of Computer Science and Engineering, National Institute of Technology Srinagar, Jammu and Kashmir, India, 190006 (phone: +91-9797875163; e-mail: [burhan\\_07phd12@nitsri.net](mailto:burhan_07phd12@nitsri.net)).

ii. Usually specialized FPGA resources are fixed in position. Routing data to and from the fixed blocks sometimes creates problems in the placement and routing (PAR) phase of the FPGA design flow. Thus instead of using fixed specialized resources it is desirable to use general LUT resources as their placement can be altered during PAR.

iii. Finally, retiming GPC structures by placing registers at the input of nodes with large capacitances reduces the switching activities at these nodes [17]. This results in reduced dynamic power dissipation.

The rest of the paper is organized as follows. Section II presents the basic preliminaries about the GPCs and the terminology used in this paper. Section III discusses the heuristic that is used to synthesize different GPCs. Synthesis and implementation is carried out in section IV. Conclusions are drawn in section V and references are listed at the end.

## II. PRELIMINARIES AND TERMINOLOGY

A compressor tree is a circuit that takes  $k$ ,  $n$ -bit unsigned operands:  $A_{k-1}, A_{k-2} \dots A_1, A_0$  and generates two output values, Sum ( $S$ ) and Carry ( $C$ ), such that:

$$\sum_{i=0}^{k-1} A_i = S + C \quad (1)$$

A generalized parallel counter computes the sum of bits having different weights. A GPC is traditionally represented as a tuple  $(K_{i-1}, K_{i-2} \dots K_1, K_0; n)$ , where  $K_i$  denotes the number of input bits of weight  $i$ , and  $n$  is the number of output bits. The upper limit on the value of GPC is given by:

$$M = K_0 2^0 + K_1 2^1 + \dots + K_{i-1} 2^{i-1} \quad (2)$$

$$M = \sum_{i=0}^{K-1} K_i 2^i \quad (3)$$

$$n = \lceil \log_2(M + 1) \rceil \quad (4)$$

As an example, a  $(1, 4, 1, 5; 5)$  GPC has five input bits of weight 0; one input bit of weight 1; four input bits of weight 2 and one input bit of weight 3. The upper limit on the output value is 31 and five output bits are required to represent the output.

Logic synthesis is concerned with hardware realization of a desired functionality with minimum possible cost. The *cost* of a circuit is a measure of its speed, resource utilization, power consumption or any combination of these. A *Boolean network* is a directed acyclic graph (DAG) that represents a combinational function. Logic gates, primary inputs (PIs) and primary outputs (POs) within this network are represented by *nodes*. Each node implements a *local function*. A *global function* is implemented by connecting the logic implemented by individual nodes. The transformation of a Boolean network into targeted logic elements gives the *circuit-netlist*. For FPGAs the targeted element is a  $k$ -LUT.

A *cone* of node  $v$ ,  $C_v$ , is a sub-network that includes the node  $v$  and some of its non-PI predecessor nodes. Any node  $u$  within this cone has a path to the *root* node  $v$ ,  $u \rightarrow v$ , which lies

entirely in  $C_v$ . The *level* of the node  $v$  is the length of the longest path from any PI node to  $v$ . Network *depth* is defined as the largest level of a node in the network. The critical path delay and area of a circuit is measured by the depth and number of LUTs respectively. A node may have zero or more predecessor nodes known as *fan-in* nodes. Similarly a node may drive zero or more successor nodes known as *fan-out* nodes. A network is said to be  $k$  bounded if the fan-in of every node does not exceed  $k$ .

## III. GPC MAPPING HEURISTIC

This section describes the heuristic for efficiently mapping the GPCs onto LUTs. The primary goal of the heuristic is to eliminate the fast carry chain and map the GPCs onto minimum possible LUTs. Eliminating the carry chain makes the GPCs feasible to pipelining. The resulting structures are easily pipelined by placing the registers along the feed-forward cut-sets. We explain the different steps involved in the heuristic by considering the mapping of GPC  $(1, 4, 1, 5; 5)$ . Conventional implementation requires four LUTs and a CARRY4 primitive, with a total delay of  $T_L + 4T_{CC}$ , where  $T_L$  is the delay associated with a single LUT and  $T_{CC}$  is the single carry delay. Figure 1 shows the Boolean network for  $(1, 4, 1, 5; 5)$  GPC. The network has eleven inputs and five outputs. All the primary inputs, primary outputs and intermediate signals have been labeled.

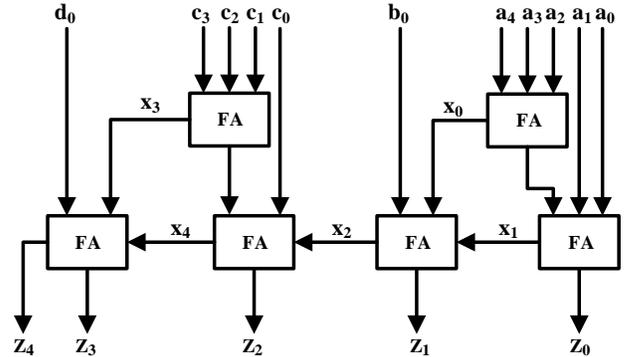


Fig. 1. Boolean network for  $(1, 4, 1, 5; 5)$  GPC

**Construction:** The first step constructs multiple networks from the original network. This is done by traversing the parent network and dividing it at the output nodes. Thus Boolean networks corresponding to each output node are constructed in this step. For the parent network of figure 1 there are five output nodes resulting in five different Boolean networks. The individual networks are named as per their outputs  $Z_0, Z_1, Z_2, Z_3$  and  $Z_4$ . This is shown in figure 2.

**Recognition and Prioritization:** After the individual networks have been obtained, the heuristic searches for redundant nodes in each of the networks. Redundant nodes are the nodes which exist in more than one network. These are shown as shaded portions in figure 2. The network for redundant nodes is then drawn separately as shown in figure 3. Each redundant network is assigned a priority based on the number of appearances in the original networks of figure 2. For example, the network in figure 3(a) is assigned a priority

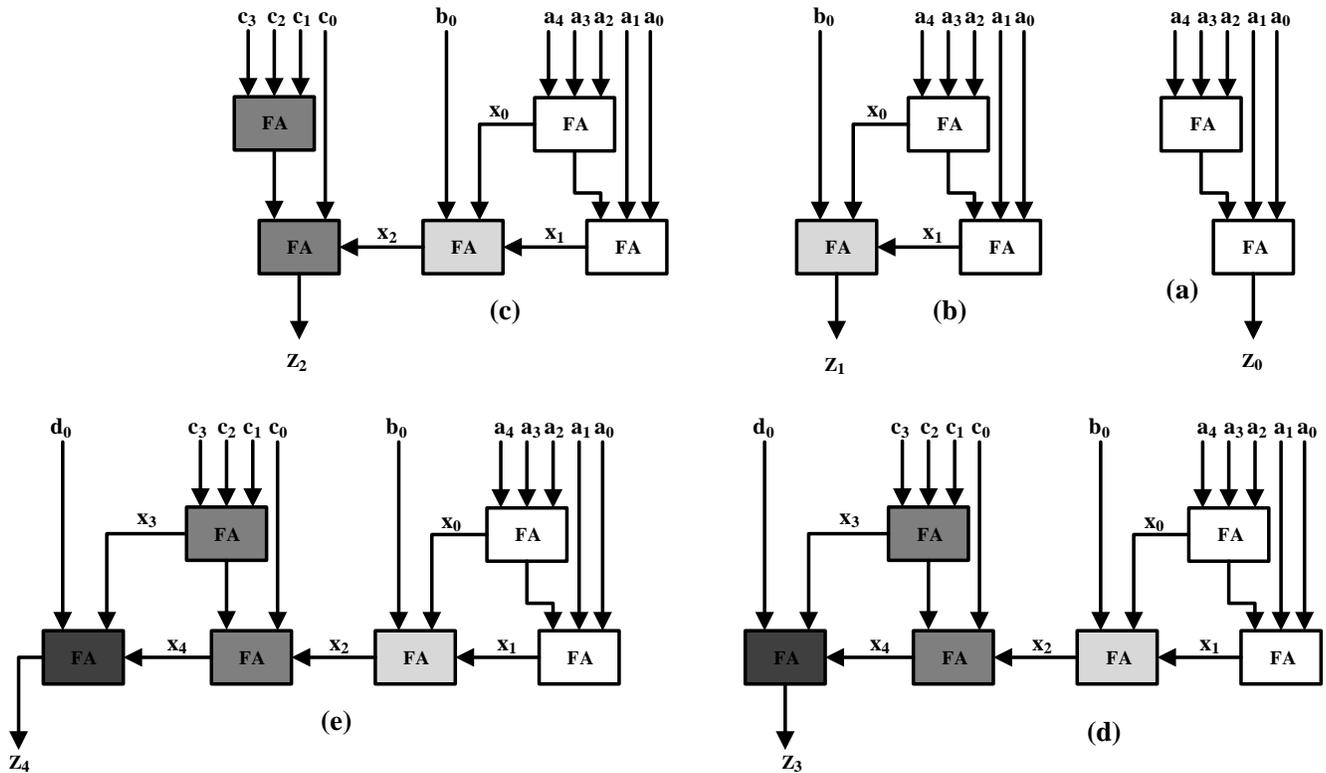


Fig. 2. Boolean networks for individual outputs. Redundant nodes are shown in same shades.

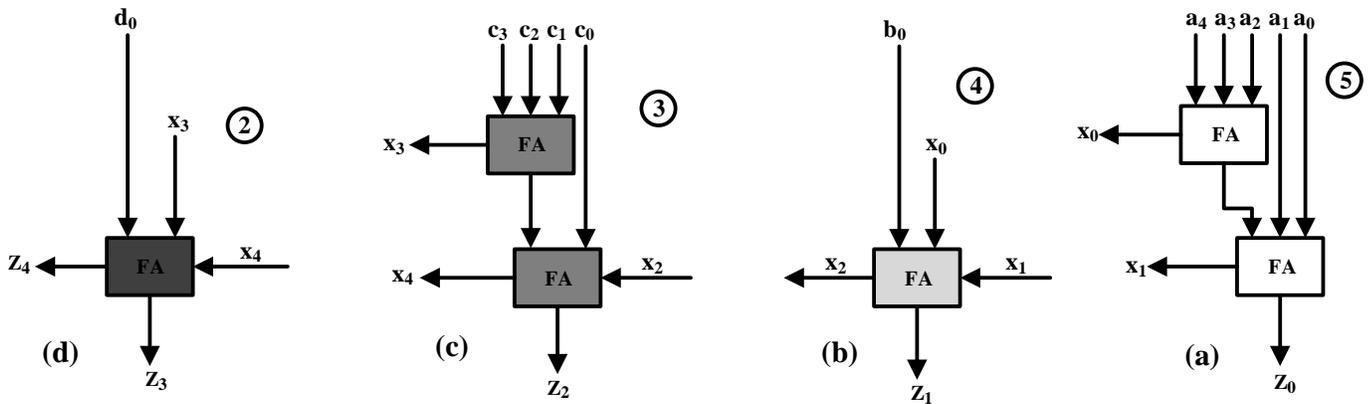


Fig. 3. Boolean networks for redundant nodes. The number in the circle represents the priority of each network.

of 5 because it appears in five different networks. Similarly 3(b) is assigned a priority of 4 because it appears in four different networks and so on. Note that the entire parent network can be constructed by interconnecting these redundant networks.

**Covering and Re-structuring:** Next the heuristic tries to optimally map these redundant networks onto LUTs. Mapping is done as per the priority, as it results in the maximum logic density. For example the network in figure 3(a) has a priority of 5 and, if mapped optimally will result in an improved logic density in all the networks it is a part of. In this paper, we have targeted FPGAs with 6-input LUTs as basic logic elements. Thus the mapping should ensure a proper utilization of this

basic element. For efficient mapping each network in figure 3 is divided into sub-networks. This is again done by traversing through the network and dividing it at output nodes. Thus the network of figure 3(a) is divided into three sub-networks corresponding to outputs  $X_0$ ,  $X_1$  and  $Z_0$ . Similarly networks in 3(b), 3(c) and 3(d) are divided into different sub-network as per their fan-out. This is shown in figure 4. A straight forward approach to mapping would be to assign the logic implemented by each sub-network to a separate LUT. This, however, leads to under utilization of the resources. For efficient mapping, therefore, the entire assembly of sub-networks is re-structured. This requires transferring some sub-networks from their original networks to sub-networks that

belong to different networks. For example sub-network  $X_0$  that originally belonged to 4(a) is now transferred to 4(b) and included with sub-networks  $X_2$  and  $Z_1$ . This re-structuring of sub-networks ensures a proper utilization of the LUT fabric. Note that the 6-input LUTs in Xilinx FPGAs can implement a

single 6-input function or two 5-input functions with shared inputs. The re-structured sub-networks are shown in figure 5. The re-structured sub-networks are then efficiently mapped onto 6-input LUTs by directly mapping their functionalities onto these target elements.

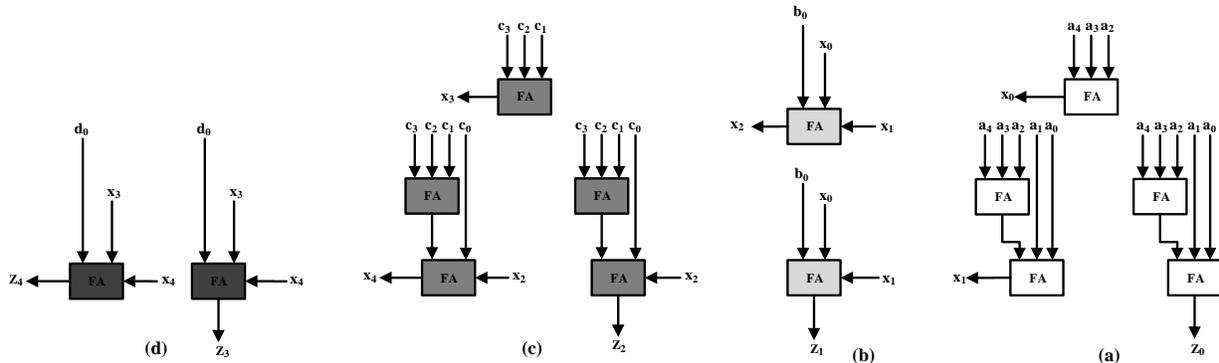


Fig. 4. Sub-networks for different networks

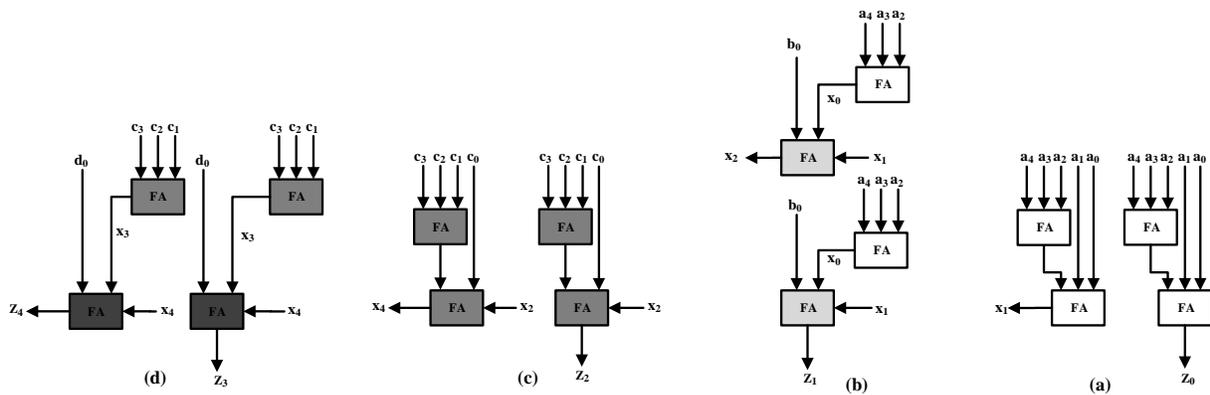


Fig. 5. Re-structuring of networks for efficient utilization of LUTs

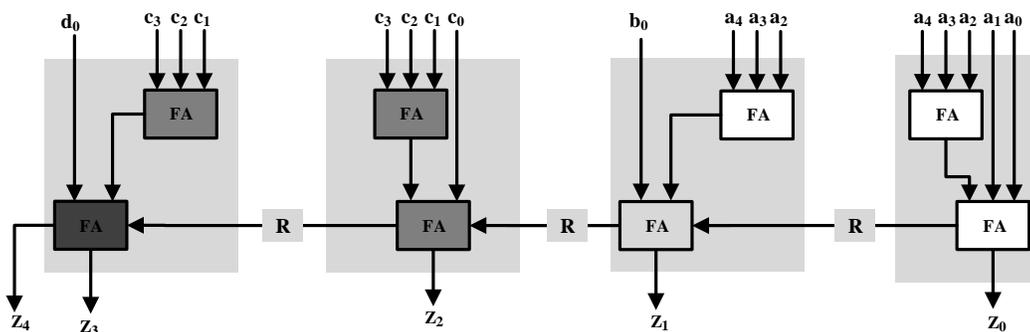


Fig. 6. Re-timed optimal circuit for (1, 4, 1, 5; 5) GPC

**Re-construction and Re-timing:** The parent network is then constructed by connecting the mapped networks from step III. The overall structure is a simple feed-forward structure having a unidirectional dataflow. This feed-forward nature lends itself for efficient pipelining by simply placing the registers along the feed-forward cut-sets. The final mapped and re-timed structure is shown in figure 6.

The circuit implementation of figure 6 requires four LUTs and three registers and has a critical path that includes only the delay of a single LUT ( $T_L$ ). The carry chain has been eliminated and there is no increase in the delay associated with the GPC. Different GPCs proposed in prior work were

implemented using this heuristic. The carry chain was successfully eliminated in all of the GPCs with no extra hardware cost, except in few cases where the column length of the GPCs exceeded five. The circuits for different GPCs are shown in figures 7, 8, 9 and 10. A theoretical evaluation of different GPCs is listed in table 1. With respect to table 1 it should be noted that previous implementations using carry chains consider only LUTs as the hardware resource. However, for each bit in a carry chain there is a carry multiplexer (MUXCY) and a dedicated XOR gate for adding/subtracting the operands with a selected carry bit. Thus an increase in LUT count that is observed in some GPCs using

our heuristic may be compensated by the elimination of the resources included in the carry chain.

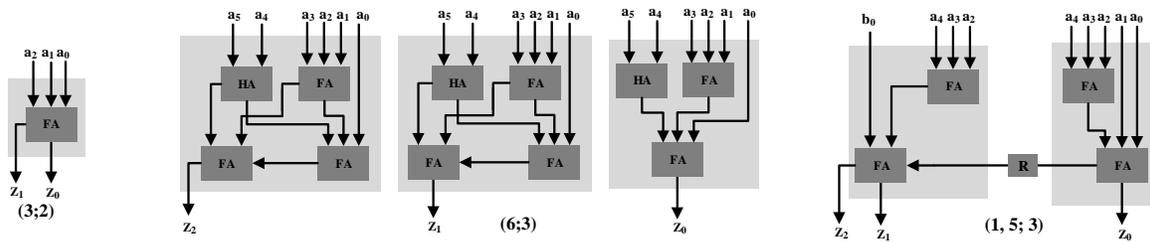


Fig. 7. LUT based GPCs from [9]

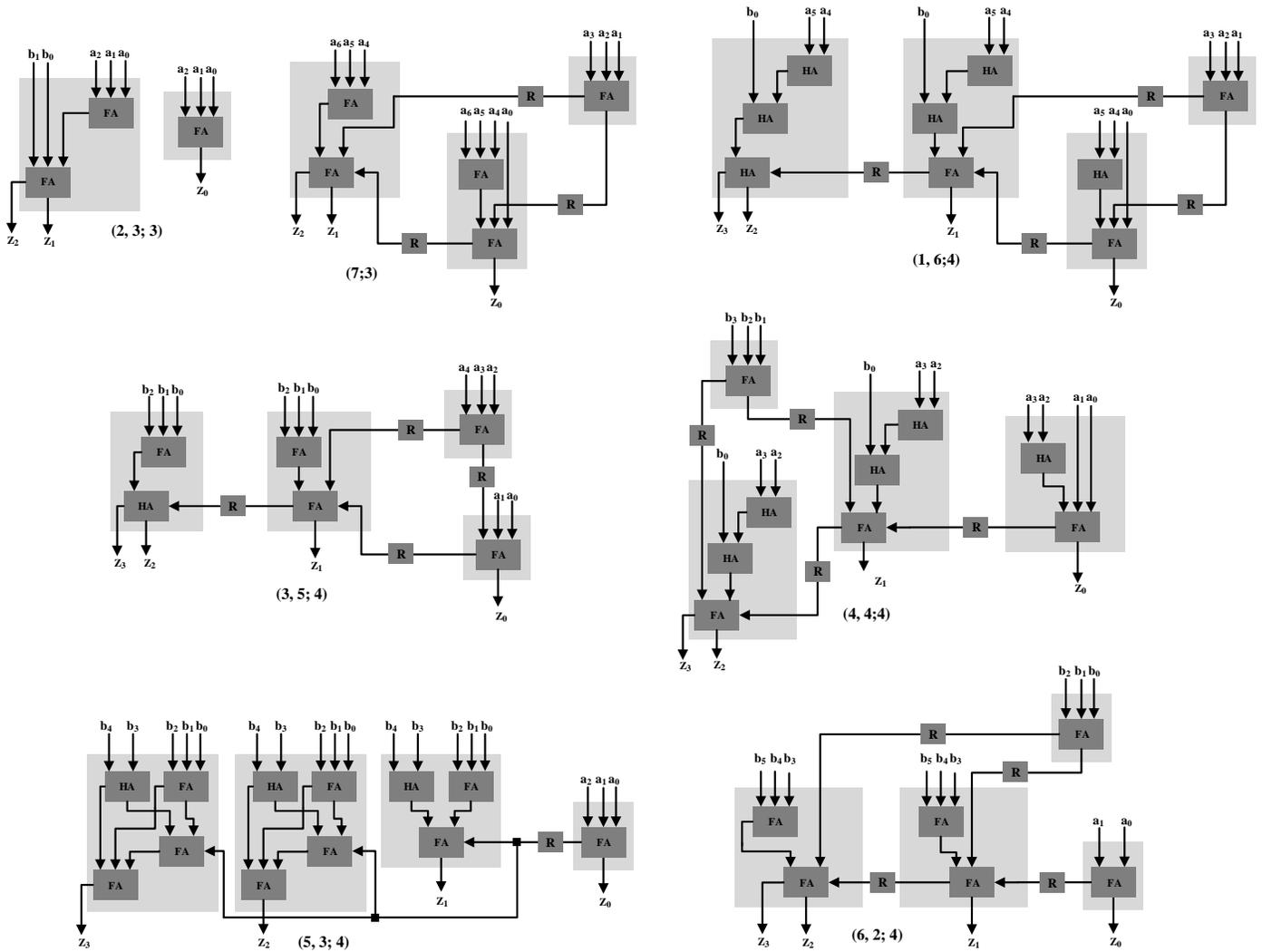


Fig. 8. LUT based GPCs from [8]

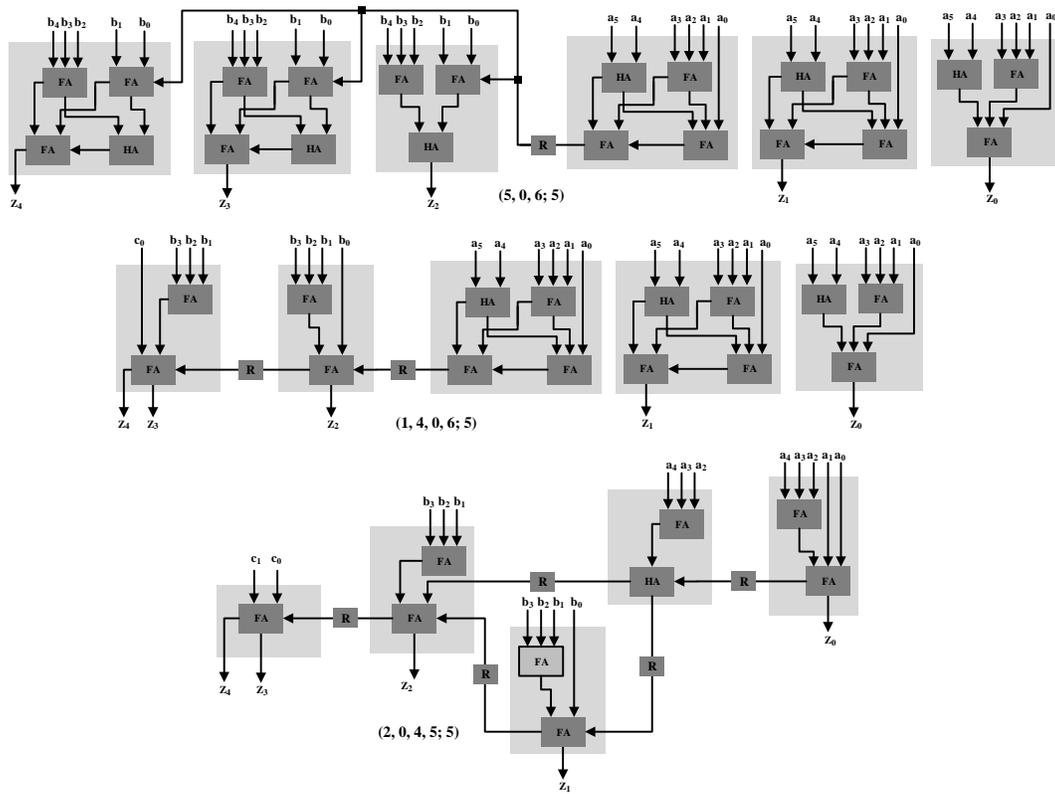


Fig.9. LUT based GPCs from [15]

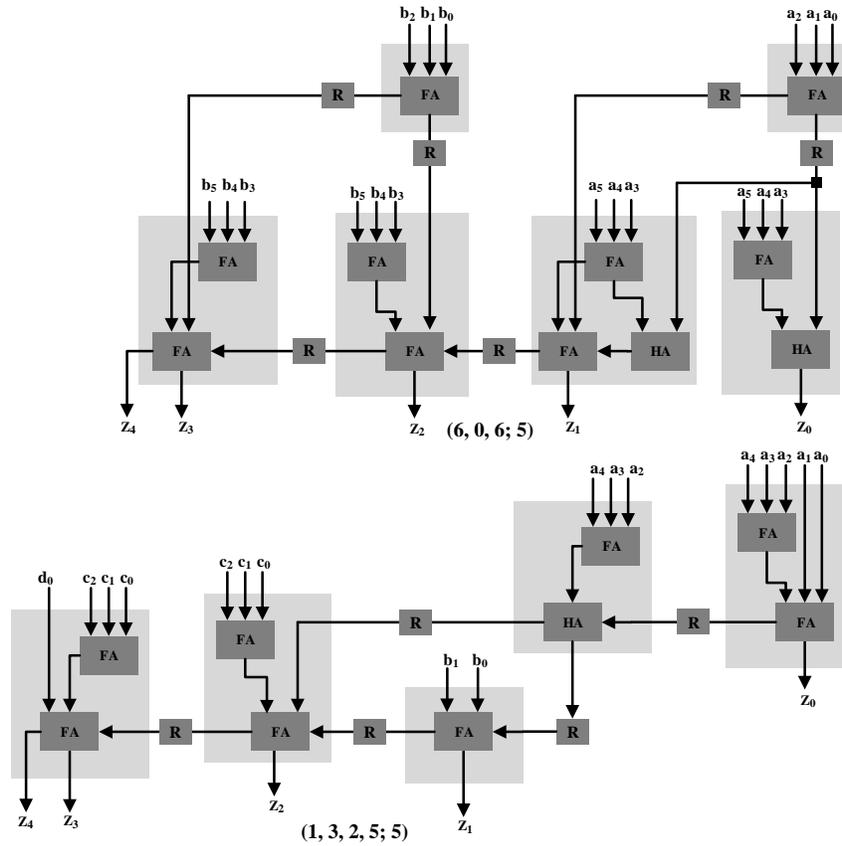


Fig. 10. LUT based GPCs from [16]

TABLE I  
COMPARISON OF DIFFERENT GPCs

GPCs	Previous Mappings		Mappings based on proposed heuristic	
	LUTs	Delay	LUTs	Delay
<b>GPCs from [9]</b>				
(3;2)	1	$T_L^1$	1	$T_L$
(6;3)	3	$T_L$	3	$T_L$
(1,5;3)	3	$T_L$	2	$T_L$
<b>GPCs from [8]</b>				
(6;3)	4	$2T_L+T_R^2+4T_{CC}^3$	3	$T_L$
(1,5;3)	3	$T_L+3T_{CC}$	2	$T_L$
(2,3;3)	3	$T_L+3T_{CC}$	2	$T_L$
(7;3)	4	$2T_L+T_R+4T_{CC}$	3	$T_L$
(1,6;4)	4	$2T_L+T_R+4T_{CC}$	4	$T_L$
(3,5;4)	4	$2T_L+T_R+4T_{CC}$	4	$T_L$
(4,4;4)	4	$2T_L+T_R+4T_{CC}$	4	$T_L$
(5,3;4)	4	$2T_L+T_R+4T_{CC}$	4	$T_L$
(6,2;4)	4	$2T_L+T_R+4T_{CC}$	4	$T_L$
<b>GPCs from [15]</b>				
(6;3)	3	$2T_L+T_R+3T_{CC}$	3	$T_L$
(1,5;3)	2	$T_L+2T_{CC}$	2	$T_L$
(2,3;3)	2	$T_L+2T_{CC}$	2	$T_L$
(7;3)	3	$2T_L+T_R+3T_{CC}$	3	$T_L$
(5,3;4)	3	$2T_L+T_R+3T_{CC}$	4	$T_L$
(6,2;4)	3	$2T_L+T_R+3T_{CC}$	4	$T_L$
(5,0,6;5)	4	$T_L+4T_{CC}$	6	$T_L$
(1,4,1,5;5)	4	$T_L+4T_{CC}$	4	$T_L$
(1,4,0,6;5)	4	$T_L+4T_{CC}$	5	$T_L$
(2,0,4,5;5)	4	$2T_L+T_R+4T_{CC}$	5	$T_L$
<b>GPCs from [16]</b>				
(6,0,6;5)	4	$T_L+4T_{CC}$	6	$T_L$
(1,3,2,5;5)	4	$T_L$	5	$T_L$

<sup>1</sup>delay associated with LUT.

<sup>2</sup>delay associated with routing.

<sup>3</sup>delay associated with carry chain

#### IV. SYNTHESIS, IMPLEMENTATION AND RESULTS

Synthesis and implementation is done using xc7k70t-2fbg676 device from Xilinx Kintex-7 family. The parameters considered are resources utilized, critical path delay and dynamic power dissipation. Constraints relating to synthesis and implementation are duly provided and a complete timing closure is ensured in each case. Synthesis and implementation is carried out in Xilinx Vivado 2016.3 [18] with speed as the optimization goal. Power analysis is done using the Xpower analyzer tool. For power analysis switching activity is captured in the value change dump (VCD) file by applying test vectors and checking for correct output. Similar test benches have been used to ensure a fair comparison. Table 2

provides a comparison of different performance metrics for different GPCs.

From table 2 it is observed that the GPC mappings based on the proposed heuristic show an average increase in speed by almost 65% and an average reduction in dynamic power dissipation by 10%. The carry chain is eliminated in each GPC with an overhead of pipelining registers and LUTs (in few cases). Each slice in Kintex-7 supports four registers which normally remain unutilized. Our experimentation with different arithmetic circuits on Kintex-7 devices reveal that each carry chain utilizes resources that are equivalent to 1 to 1.5 6-input LUTs. Thus any increase in LUT count is justified by the elimination of carry chain.

TABLE II  
PERFORMANCE COMPARISON OF DIFFERENT GPCs ON XC7K70T-2FBG676

GPCs	Previous Mappings			Mappings based on proposed heuristic		
	LUTs	Critical path(nS)	Power (mW)	LUTs	Critical path(nS)	Power(mW)
<b>GPCs from [9]</b>						
(3;2)	2	0.318	0.0287	1	0.091	0.0213
(6;3)	3	0.642	0.0300	3	0.641	0.0240
(1,5;3)	3	0.642	0.0301	2	0.108	0.0214
<b>GPCs from [8]</b>						
(6;3)	4	0.872	0.0321	3	0.641	0.0300
(1,5;3)	3	0.772	0.0301	2	0.108	0.0300
(2,3;3)	3	0.602	0.0301	2	0.318	0.0300
(7;3)	4	0.877	0.0321	3	0.108	0.0300
(1,6;4)	4	1.101	0.0321	4	0.108	0.0300
(3,5;4)	4	1.002	0.0321	4	0.108	0.0300
(4,4;4)	4	0.988	0.0321	4	0.108	0.0300
(5,3;4)	4	0.887	0.0321	4	0.602	0.0300
(6,2;4)	4	1.007	0.0321	4	0.108	0.0300
<b>GPCs from [15]</b>						
(6;3)	3	0.802	0.0311	3	0.641	0.0300
(1,5;3)	2	0.796	0.0300	2	0.108	0.0300
(2,3;3)	2	0.598	0.0300	2	0.318	0.0300
(7;3)	3	0.885	0.0311	3	0.108	0.0300
(5,3;4)	3	0.839	0.0311	4	0.602	0.0300
(6,2;4)	3	0.991	0.0311	4	0.108	0.0300
(5,0,6;5)	4	0.989	0.0500	6	0.641	0.0426
(1,4,1,5;5)	4	0.839	0.0500	4	0.091	0.0371
(1,4,0,6;5)	4	0.989	0.0500	5	0.641	0.0377
(2,0,4,5;5)	4	1.101	0.0500	5	0.091	0.0377
<b>GPCs from [16]</b>						
(6,0,6;5)	4	1.006	0.0500	6	0.091	0.0452
(1,3,2,5;5)	4	0.989	0.0500	5	0.091	0.0377

## V. CONCLUSIONS

In this paper we took an alternate approach to GPC synthesis on FPGAs. Unlike prior work on GPC synthesis that used a combination of LUTs and carry chains, we used a combination of LUTs and registers and eliminated the carry chain completely from the GPC structure. Our approach works in two steps: first a heuristic is used to eliminate the carry chain and map the GPC logic efficiently onto the underlying LUT fabric. The mapped GPC is then retimed by placing the registers along the feed-forward cut-sets. Retiming breaks the critical path resulting in higher operating frequencies. Our implementation targeting Xilinx FPGAs show an increase in speed and reduction in power dissipation for almost same resources utilized.

## REFERENCES

- [1] S. Mirzaei, A. Hosangadi, and R. Kastner, "High speed FIR filter implementation using add and shift method," International Conference on Computer Design, San Jose, CA, USA, Oct. 1-4, 2006.
- [2] C.Y. Chen, S.Y. Chien, Y.W. Huang, T.C. Chen, T.C. Wang, and L.G. Chen, "Analysis and architecture design of variable block-size motion estimation for H.264/AVC," IEEE Transactions on Circuits and Systems-I, Vol. 53, No. 2, pp. 578-593, Feb. 2006.
- [3] L. Dadda, "Some schemes for parallel multipliers," Alta Frequenza, Vol. 34, pp. 349-356, May, 1965.
- [4] O. Kwon, K. Nowka, and Jr. Swartzlander, "A 16-bit by 16-bit MAC design using fast 5:3 compressor cells," Journal of VLSI Signal Processing, Vol. 31, No. 2, pp. 77-89, June, 2002.
- [5] H. Mora Mora, J. Mora Pascual, J. L. Sánchez Romero, and F. Pujol López, "Partial production reduction based on lookup tables," International Conference on VLSI Design, Hyderabad, India, pp. 399-404, January 3-7, 2006.
- [6] J. Poldre and K. Tammema, "Reconfigurable multiplier for Virtex FPGA family," International Workshop on Field-Programmable Logic and Applications, Glasgow, UK, pp. 359-364, Aug. 30 – Sept. 1, 1999.
- [7] C. S. Wallace, "A suggestion for a fast multiplier," IEEE Transactions on Electronic Computers, Vol. 13, pp. 14-17, Feb., 1964.
- [8] H. Parandeh-Afshar, A. Neogy, P. Brisk and P. lenne, "Compressor Tree Synthesis on Commercial High-Performance FPGAs," ACM Transactions on Reconfigurable Technology and Systems, Vol. 4, No. 4, Article 39, December 2011.
- [9] H. Parandeh-Afshar, P. Brisk, and P. lenne, "Efficient Synthesis of Compressor Trees on FPGAs," in Asia and South Pacific Design Automation Conference (ASPDAC). IEEE, 2008, pp. 138-143.

- [10] H. Parandeh-Afshar, P. Brisk, and P. Ienne, "Exploiting Fast Carry-Chains of FPGAs for Designing Compressor Trees," Proceedings of the 19<sup>th</sup> International Conference on Field Programmable Logic and Applications. 242-249.
- [11] H. Parandeh-Afshar, P. Brisk, and P. Ienne, "Improving Synthesis of Compressor Trees on FPGAs via Integer Linear Programming," in Design, Automation and Test in Europe (DATE). IEEE, 2008, pp. 1256–1261.
- [12] T. Matsunaga, S. Kimura, and Y. Matsunaga, "Power and Delay Aware Synthesis of Multi-Operand Adders Targeting LUT-Based FPGAs," International Symposium on Low Power Electronics and Design (ISLPED), pp. 217–222, 2011.
- [13] T. Matsunaga, S. Kimura, and Y. Matsunaga, "Multi-Operand Adder Synthesis Targeting FPGAs," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E94-A, no. 12, pp. 2579–2586, Dec. 2011.
- [14] T. Matsunaga, S. Kimura, and Y. Matsunaga, "An Exact Approach for GPC-Based Compressor Tree Synthesis," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E96-A, no. 12, pp. 2553–2560, Dec. 2013.
- [15] M. Kumm and P. Zipf, "Efficient High Speed Compression Trees on Xilinx FPGAs," in Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen, 2014.
- [16] M. Kumm and P. Zipf, "Pipelined Compressor Tree Optimization using Integer Linear Programming," in 24<sup>th</sup> International Conference on Field Programmable Logic and Applications, 2014.
- [17] K. K. Parhi, "VLSI Digital Signal Processing Systems Design and Implementation," Wiley, 1999.
- [18] <http://www.xilinx.com>

# Efficient Computerized-Tomography Reconstruction Using Low-Cost FPGA-DSP Chip

Bassam A. Abo-Elftooh, Mohamed H. El-Mahlawy and Mahmoud E. A. Gadallah

**Abstract**—In this paper, filtered back-projection algorithm is optimally implemented using low-cost Spartan 3A-DSP 3400 chip. The optimization enables parallel implementation. The combination of the pixel parallelism and projection parallelism is presented to significantly reduce the total reconstruction time to produce the image. The applied data is presented in fixed point format to achieve efficient implementation with maximum speed. The selection of data bus-width is optimized with very little error and good visual quality required for medical images. Before implementation, the computer tomography (CT) reconstruction simulator is developed to provide a testing reference for the hardware implementation. Using the combination of the pixel parallelism and projection parallelism, the presented hardware design achieves image reconstruction of a 512-by-512 pixel image from 1024 projections in 134.8 ms using 50 MHz clock cycles. It achieves the reduction of the required number of clock cycles to form an image from projections by 60 % comparing to the state of the art of the reconstruction time using field programmable gate array (FPGA) design.

**Index Terms**— CT image reconstruction using FPGA, Filtered Back-Projection (FBP), Pixel parallelism, Projection parallelism.

*Original Research Paper*  
DOI: 10.7251/ELSI721012A

## I. INTRODUCTION

Computerized tomography (CT) is a medical image processing application, marked by computationally intensive algorithms. It is the process of generating a cross sectional image of an object from a series of projections collected around the object [1-2]. The reconstruction process uses these projections to calculate the average x-ray attenuation coefficient in cross-sections of a scanned slice. The most common approach for the CT image reconstruction is filtered back-projection (FBP). The FBP comes in parallel-beam and fan-beam variations [2]. Parallel-beam back-

projection is focused on this paper, but the presented design in this paper can be extended to the fan-beam back-projection with modifications. The FBP includes digital signal processing (DSP) functions and requires accurate and fast processing of large data, which forces CT equipment manufacturers to strike a balance between image resolution, image generation time, and system cost when designing a new scanner.

Hardware implementation of FBP accelerates this processing. Field programmable gate array (FPGA) vendors introduce economical devices with large capacity and DSP capabilities. The FBP can benefit from the features of the FPGA. Furthermore, the fine-grained parallelism inherent to FPGAs is suitable for a highly parallelizable process like FBP [3-4].

Higher image resolution provides more diagnostic details to the radiologist, but it takes longer time to generate the images. It requires more processing power that significantly adds to the cost. Cost-effective acceleration of image reconstruction would allow manufacturers to generate higher resolution of the CT images while maintaining the required balance in their system design. In this paper, it is required to implement the parallel-beam FBP algorithm used in the CT on a single FPGA chip based on the mentioned features. Before hardware implementation, a software simulator is built to analyze and get accurate validation of each reconstruction process, used as a reference to validate each stage in the hardware design. The generated image from hardware is imported to the software simulator for the comparison with the generated image from the software. In addition, the simulator assists the hardware implementation to be optimized and minimized the hardware overhead.

Several previous works were attempted in the area of hardware implementation of the parallel-beam FBP algorithm. The authors in [5] demonstrated the fixed point architecture of the sinogram data quantized with 12-bit, and it does not be optimized for medical applications. The author in [6] showed the main divisions of CT scanner and suggests the areas that can be implemented using the Application Specific Integrated Circuits (ASICs) and the FPGA. The authors in [7] presented an ASIC hardware implementation of Radon transform, and the multiprocessing of the parallel-beam back-projection. The authors in [8-9] presented a proposal to accelerate the FBP but they do not show the hardware implementation and do not consider the quality of the medical image. The authors in [10]

Manuscript received 31 July 2015. Received in revised form 30 May 2017.  
Accepted for publication 28 June 2017.

Bassam A. Abo-Elftooh is with Electronics and Communication Engineering department, Faculty of Engineering Sciences and Arts, Misr International University (MIU), Cairo, Egypt (corresponding author: +201141978844; e-mail: engineerbassam@gmail.com).

Mohamed H. El-Mahlawy is with Electrical Engineering department, faculty of Engineering and Technology, Future University in Egypt (FUE), Cairo, Egypt (email: mohamed.elmahlawy@fue.edu.eg)

Mahmoud E. A. Gadallah is with the Modern University, Maadi, Cairo, Egypt (e-mail: mgadallah1956@gmail.com).

used fixed point implementation and maximized the parallelism but they deal with Joint Photographic Experts Group (JPEG) image that is not used in medical applications. The authors in [11] implemented the parallel-beam FBP algorithm based on projection parallelism. They achieved image reconstruction of a 512-by-512 pixel image from 1024 projections in 250 ms using expensive Virtex FPGA chip with 65 MHz. The main objective of this paper is to build hardware for fast reconstruction of high image quality using economical chip. Therefore, the combination of the pixel parallelism and projection parallelism is utilized using low-cost Spartan 3A-DSP 3400 chip with 50 MHz clock cycles. It achieves image reconstruction of a 512-by-512 pixel image from 1024 projections in 134.8 ms using 50 MHz clock cycles. The presented FPGA hardware implementation of the parallel-beam FBP algorithm, based on the pixel parallelism and projection parallelism, is the most efficient in the speed and the hardware cost with respect to all previous published works.

This paper is organized as follows: section II presents the concept of the FBP algorithm. Developing of the CT image reconstruction simulator will be presented in section III. Developing of the non-parallel hardware implementation of the FBP algorithm will be presented in section IV. The parallel hardware implementation of the FBP algorithm will be presented in section V. Finally, the conclusion and the future work will be discussed in the last section.

## II. FILTERED BACK-PROJECTION ALGORITHM

Radon found the way for reconstruction of the image from projections, which is only one part of the CT system [12]. A practical method was made by Hounsfield [13], and Cormack independently invented the same process [14]. The basis of tomographic imaging is described in other papers [15-16]. In the BP, the measurements at each angle (projection) are smeared back along the same line ( $\theta$ ), as shown in Fig. 1. It is known the point of density is somewhere along that line. Therefore, a crude reconstruction results when the measured value is assigned along the entire line. Adding up the values for all  $\theta$  will yield a picture of the object. When only six projections are used, a star-shaped pattern emanating from the dense points emerges as shown in Fig. 2(a). When enough projections are used, the blurring is apparent as shown in Fig. 2(b).

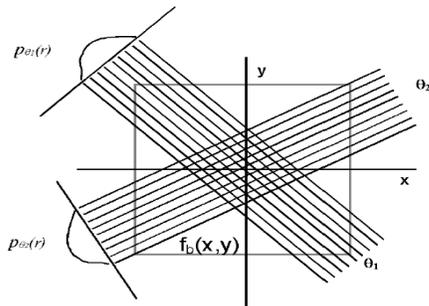


Fig. 1. Two projections are back-projected and added together.

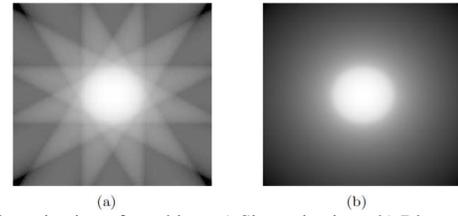


Fig. 2. Back-projecting of an object. a) Six projections. b) Blurred object.

The BP for a single projection of unknown density is [2]:

$$imm_{\theta}(x, y) = \int P_{\theta}(r) \delta(x \cos \theta + y \sin \theta - r) dr \quad (1)$$

where,  $imm_{\theta}(x, y)$  is the back-projected density due to the projection  $P_{\theta}(r)$ . The  $\delta()$  function is the Dirac delta function. Sum over all angles to obtain a summation image;  $img_b(x, y)$ .

$$img_b(x, y) = \int_0^{\pi} imm_{\theta}(x, y) d\theta \quad (2)$$

$$img_b(x, y) = img(x, y) ** \frac{1}{r} \quad (3)$$

where  $img(x, y)$  is the actual image, and  $**$  is a 2-D convolution. The actual image is blurred by  $1/r$ . In frequency space, it means that the Fourier amplitudes have been multiplied by  $1/\omega$ . This blurring must be removed (filtered). To fix that, it is required to pre-weight the Fourier Transform (FT) of each 1-D projection with  $|\omega|$  prior to the BP, known as FBP and is still the standard technique in commercial scanners. This method is efficient because it only involves a 1-D FT which is used in practice. Alternatively, one may perform a space-domain convolution [17-18].

Due to its shape in the frequency domain,  $|\omega|$  is what is called a ramp function. Such a filter in practice is impossible to build. It is infinite in length and it has the drawback of amplifying high frequency noise. A variety of different filters may be used instead. The basic and widely used filter is the Ram-Lak filter [19], but it still amplifies high frequency noise. Superior results are obtained by multiplying  $|\omega|$  filter by a smoothing window that attenuates the higher frequencies which is mostly represents observation noise. The commonly used windows are presented in Fig. 3.

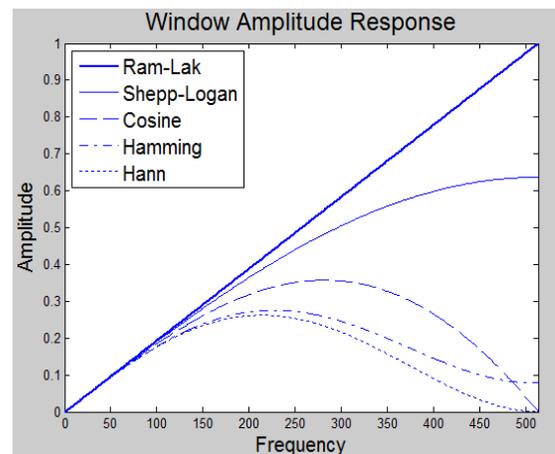


Fig. 3. Different smoothing windows

The projection data is measured into the polar coordinates while it is back-projected into the Cartesian coordinates of the final image so interpolation algorithm is required [20]. Various methods of interpolation algorithm are specified (nearest neighbor, linear or cubic). The interpolation can be performed either on the filtered projection samples (ray-driven) or on the reconstructed images (pixel-driven) [21]. At the first type, the filtered sample is traced along the x-ray path at a fixed increment. At each stop, the intensity of the sample is distributed to the neighboring pixels. Alternatively, we can start at an image pixel location and calculate the intensity contribution from the projection samples that intersect the ray pass through the center of this pixel. The pixel-driven takes place in the projection space (1-D) which makes it preferred over a ray-driven occurred in the image space (2-D) to save the intensive computation.

### III. DEVELOPMENT OF THE CT IMAGE RECONSTRUCTION SIMULATOR

The presented CT software simulator (CTSIM) in this paper is developed to simulate the CT image reconstruction process and to provide the reference for the validation of each stage in the hardware implementation [22]. The main components of the CTSIM are the object, the projections and the image reconstruction process [23]. Fig. 4 shows the block diagram of the CTSIM. The projections are generated from radon transform of Matlab Shepp-Logan phantom [24], test image, or text file including the projections (Sinogram). The object attenuation coefficients are represented by gray level intensities of the image. The first step of the image reconstruction algorithm is the filtration, done in either time or frequency domain as shown in Fig. 5, based on window selection and frequency band scaling. After that, the BP process is performed.

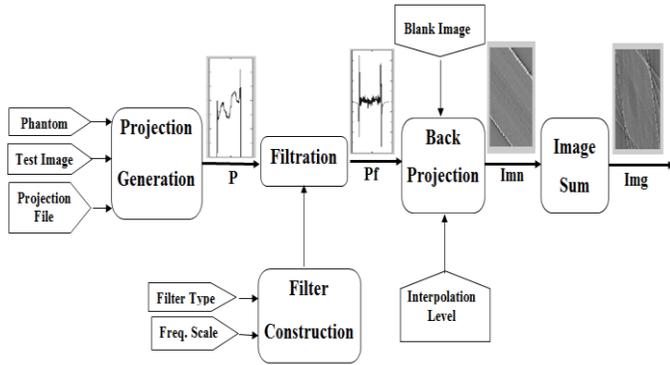


Fig. 4. General block diagram of the CTSIM.

In the BP process, the output image size is estimated from the projection length. The projection length is equal to the image diagonal. An empty image with this size is formed. This image is a 2-D matrix  $T$  that includes indices. Smearing the projection through this matrix is done by replacing each index by the gray level of the projection element with the same index, as shown in Fig.6. The BP is done with the selected interpolation algorithm.

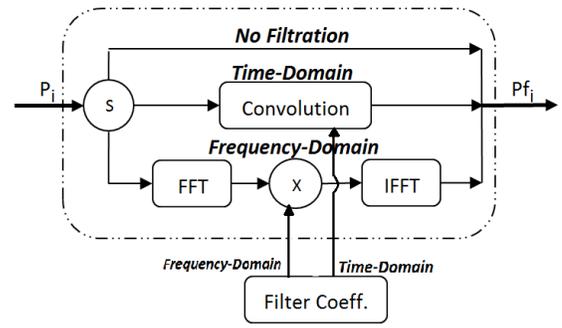


Fig. 5. The block diagram of the filtration process.

Matrix  $T$  rotates according to its corresponding angle through the equation:

$$T_{\theta}(x, y) = X \cos(\theta) + Y \sin(\theta) \quad (4)$$

This matrix is divided into an integer part ( $a_{\theta}$ ) and a fraction part ( $T_{\theta} - a_{\theta}$ ). The integer part is used for addressing, while the fraction part is used as the interpolation factor of the BP process. After the BP process of each projection is finished, the corresponding image is:

$$\begin{aligned} \text{imn}_{\theta}(x, y) = & (1 - (T_{\theta}(x, y) - a_{\theta}(x, y))) * P_{\theta}(a_{\theta}) \\ & + (T_{\theta}(x, y) - a_{\theta}(x, y)) * P_{\theta}(a_{\theta} + 1) \end{aligned} \quad (5)$$

Each new image is added to the previous one to produce the final reconstructed image:

$$\text{img}(x, y) = \sum_{\theta=0}^{179} \text{imn}_{\theta}(x, y) \quad (6)$$

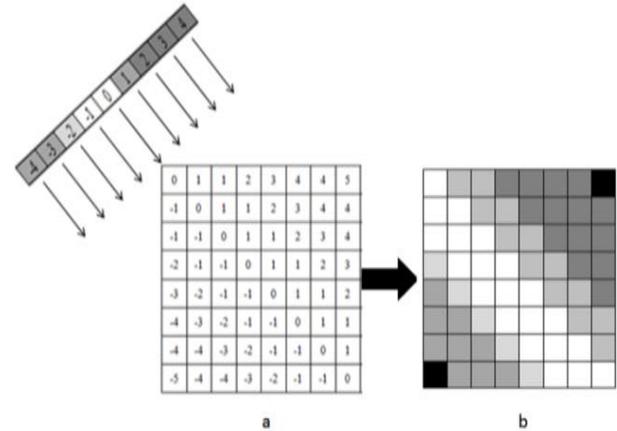


Fig. 6. Back-projection process at angle 45°.

During image reconstruction process, the program records the behavior of each projection separately through each stage to enable the tracing and analysis of the reconstruction process through different reconstruction stages. The impact of different windows on the reconstructed image is studied. For example, an oval phantom is reconstructed with a rectangular and a sinc function as shown in Fig. 7(a). The reconstructed images and the difference between them are shown in Fig. 7(b). Sinc function shows a significant noise reduction

plus a slight reduction in spatial resolution [21]. The CTSIM monitors the filtration and reconstruction time.

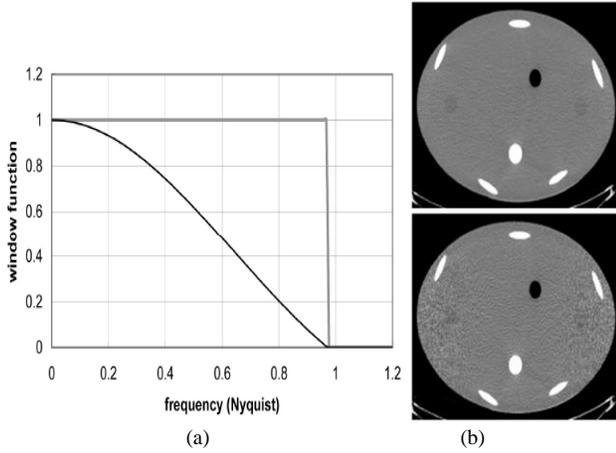


Fig. 7. (a) Different windows. (b) Upper image from Sinc window and the lower image from rectangular window.

Some techniques are used to evaluate the accuracy of the image reconstruction process using the CTSIM. The error is measured using the normalized mean absolute distance measure (ABS) and the worst case distance (WORST). The quality of reconstructed image can be evaluated using the peak signal to noise ratio (PSNR), mean structural similarity index (MSSIM) [25]. The PSNR and MSSIM tests use the Matlab Shepp-Logan phantom as a reference image. The projections are generated from this reference image and these projections are used to get the reconstructed image. They are calculated according to the following equations.

i- ABS: The normalized mean absolute error measurement.

$$ABS = \frac{\sum_{i=1}^n \sum_{j=1}^m |p_{i,j} - r_{i,j}|}{\sum_{i=1}^n \sum_{j=1}^m |p_{i,j}|} \quad (7)$$

where,  $p$  denotes the reference image and  $r$  denotes the reconstructed image. Each of the images has a size of  $m \times n$ .

ii- WORST: The worst case error over a  $2 \times 2$  pixel area.

$$WORST = \max_{1 \leq k \leq [n/2], \text{ and } 1 \leq l \leq [m/2]} (|P_{k,l} - R_{k,l}|) \quad (8)$$

where,

$$P_{k,l} = \frac{1}{4} (p_{2k,2l} + p_{2k+1,2l} + p_{2k,2l+1} + p_{2k+1,2l+1}) \quad (9)$$

$$R_{k,l} = \frac{1}{4} (r_{2k,2l} + r_{2k+1,2l} + r_{2k,2l+1} + r_{2k+1,2l+1}) \quad (10)$$

In equation 8, the term  $[n/2]$  and term  $[m/2]$  denote the largest integers less than  $n/2$  and  $m/2$ , respectively.

iii- PSNR: the peak signal to noise ratio in decibels between two images. First calculates the mean-squared error:

$$MSE = \frac{\sum_{M,N} [p_{m,n} - r_{m,n}]^2}{M * N} \quad (11)$$

$M$  and  $N$  are the number of rows and columns of input image, respectively, then computes the PSNR:

$$PSNR = 10 \log_{10} \left( \frac{F^2}{MSE} \right) \quad (12)$$

where,  $F$  is the maximum fluctuation in the input image (255 for 8-bit).

iv- MSSIM: The mean structural similarity index between two images. If one of the images being regarded as perfect quality (phantom or test image), then MSSIM can be considered as the quality measure of the other image (reconstructed) [25]. When the projection source selection is a phantom or test image, we have a reference image that provides us the opportunity to test the output image compared to the input one.

In addition, the CTSIM helps to trade-off the interpolation algorithm which affects the smoothness at the cost of image reconstruction time as shown in Fig. 8. Nearest neighbor interpolation is the fastest method. However, it provides the worst smoothness. Linear interpolation slightly requires more execution time but its results are continuous and smooth. Cubic results are close to those of linear interpolation but, it requires extremely more execution time. Image quality is shown in Fig. 9.

Finally, another analysis is available; sinogram analysis. If a certain detector doesn't work probably, zero corresponding projection element value or a problem occurs at a certain angle. It is no projection element values at this angle during projections acquisition process. This will be detected by scanning all sinogram elements. This analysis guarantees complete data acquisition process.

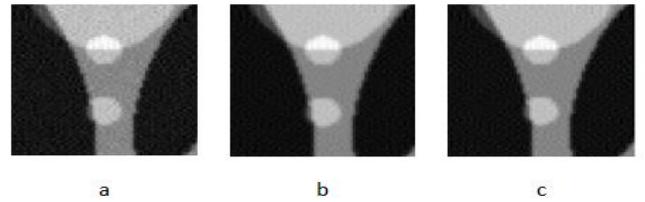


Fig. 8. a) Nearest neighbor, b) linear and c) cubic interpolation algorithm.

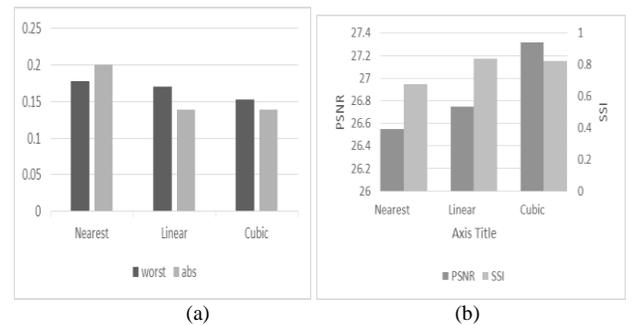


Fig. 9. (a) Worst case and absolute error. (b) PSNR and MSSIM.

#### IV. DEVELOPMENT OF NON-PARALLEL HARDWARE IMPLEMENTATION OF FBP

The main goal of this section is to reconstruct 512-by-512-pixel image from sinogram of 1024 projections (each projection has 1024 samples). The presented non-parallel hardware implementation is enhanced by applying the data in fixed point format. The block and the schematic diagram of the proposed FPGA implementation are shown in Fig. 10 and Fig. 11, respectively. It consists of the FBP main block, the projection buffering memory and the image summation memory. These two memories are off-chip to provide adequate space for the implemented FBP algorithm.

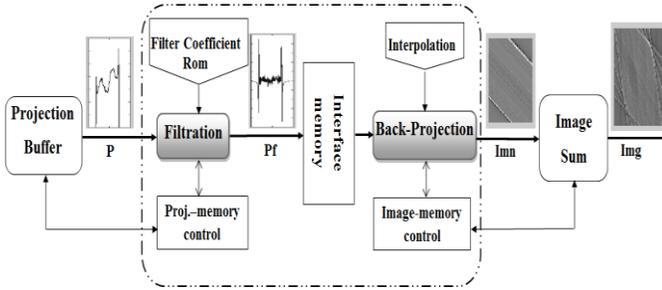


Fig. 10. The main block diagram of the presented implemented algorithm.

The main block has two controllers; the first one controls the projection acquisition process while the other controls the output image summation. FBP main block consists of three blocks; filtration, back-projection and ping-pong dual-port memory. After a projection is acquired and filtered, it is back-projected.

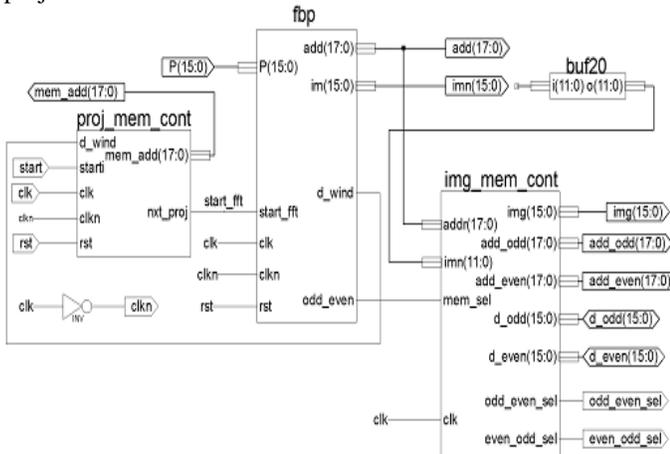


Fig. 11. Schematic diagram of the top level implemented algorithm.

##### A. Projection acquisition

The projection is acquired through projection memory controller, shown in Fig. 10 and Fig. 11, from the *projection buffering memory*. The controller organizes handshaking between the projection buffering memory and the FBP module, through two control signals (*start\_filt*, *d\_wind*). The first control signal tells the filtration module that there is a projection ready to be filtered. The filtration module in turn responds with control signal *d\_wind* which enables projection acquisition process from the memory, shown in Fig. 12. Using

the CTSIM, presented in section III, it is found that the suitable format to represent the projection (sinogram data) is (1, 8, 7) for sign, integer, and fraction, respectively. The sinogram has 1024 projections (each projection has 1024 samples). The required memory capacity is 1 Mega words. The word of the projection sample is 16 bits.

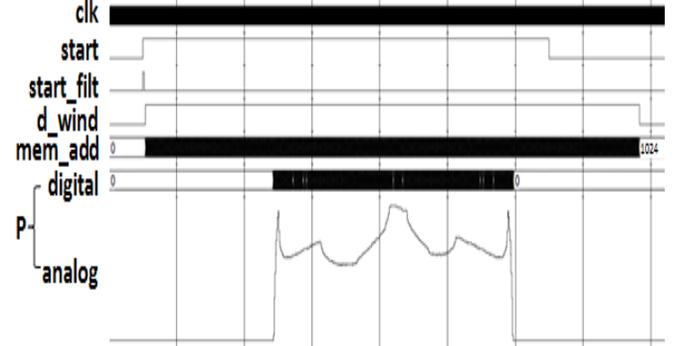


Fig. 12. Control signals and projection extraction.

##### B. Filtered back projection (FBP)

The FBP subsystem receives the acquired projection,  $P(15:0)$ , from the external *projection buffering memory* and produces a corresponding image,  $imn(15:0)$ . It is divided into three parts, shown in Fig. 13 and explained as follows.

1. *Filtration*: The filtration is implemented in the frequency domain with direct mapping from the software of the simulation (Fast Fourier Transform (FFT), filter coefficient multiplication, and then Inverse Fast Fourier Transform (IFFT)). Fig. 14 shows the timing diagram of the filtration process. Most signals in Fig. 14 are internal signals of the filtration core in the schematic diagram of Fig. 13. The FFT is implemented with FFT intellectual property (IP) core [26] that is used in the pipelining architecture to provide faster transform but with maximum hardware utilization. After the assertion of the control signal *start\_filt*, the FFT core starts and produces required indices,  $xn\_index$ , corresponding to the order of the input samples which are used to address the projection buffering memory.

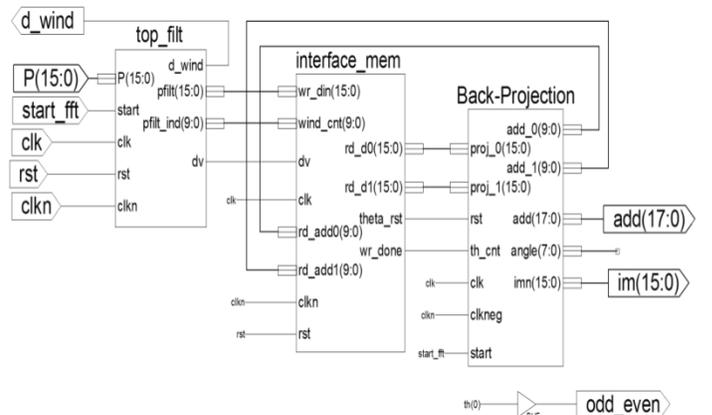


Fig. 13. Schematic diagram of the FBP.

The projection samples are applied to the real input of the core ( $xn\_re$ ), while the imaginary input ( $xn\_im$ ) is grounded. The core produces FFT real and imaginary outputs  $xk\_re$  and

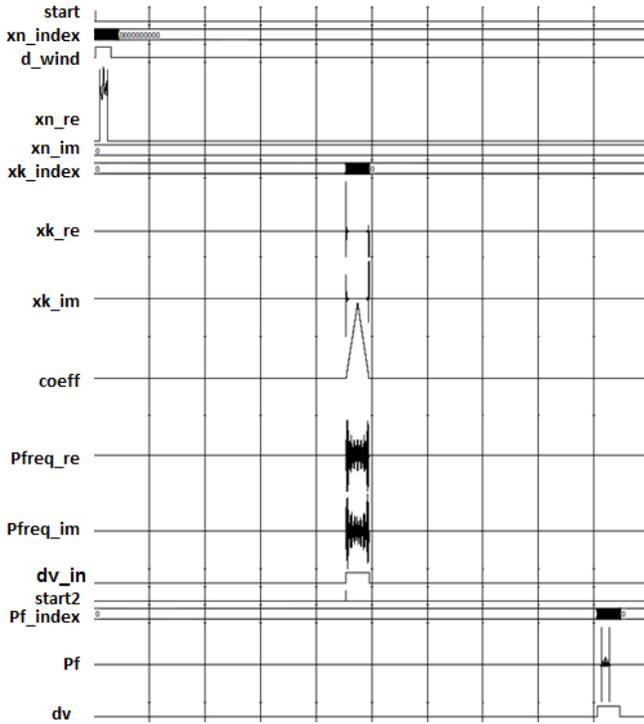


Fig. 14. Timing diagram of the filtration process.

$xk\_im$  with its corresponding indices,  $xk\_index$ . These indices address the ROM containing the filter coefficients to synchronize the output samples with its corresponding coefficients. The filter coefficients are exported from the CTSIM, presented in section III, to a COE file and loaded to the ROM during the FPGA chip download. The FFT real and imaginary outputs are then multiplied by the filter coefficients. The input sample (1, 8, 7) format is expanded through the FFT core to be (1, 19, 7) output sample. The coefficients format is (0, 0, 9) and therefore the multiplication output is (1, 19, 16). This output could be rounded from 36 bit to 16 bit with format (1, 8, 7). The multiplier is implemented with three stages of pipelining that minimizes the hardware utilization and, in the same time, provides the IFFT core the desired three clock cycles latency [26].

The multiplier outputs are applied to the IFFT, implemented with pipelining architecture. The output of the IFFT has (1, 19, 7) format which in turn rounded to 16 bit with (1, 14, 1) format. The output of the IFFT has to be divided by the number of samples ( $N = 1024$ ) that is done by shifting the data 10 bits to the right. So the filtered projection has (1, 4, 11) format. All the rounding schemes depend on the analysis done by the CTSIM for the maximum and minimum possible values in each stage. The IFFT produces three outputs; the filtered projection,  $Pf(15:0)$ , output indices,  $Pf\_ind(9:0)$ , to address the memory to which  $Pf$  be written and control signal  $dv$  which tells the memory that the  $Pf$  data is valid.

Using 50 MHz clock cycles, the latency of the FFT core for 1024 input samples is  $63.66 \mu\text{s}$  (3183 clock cycles), so the total filtration time equals to FFT latency plus multiplication time, and IFFT latency. But the multiplication is executed simultaneously with the FFT outputs with three clock cycles

latency ( $3 \times 20\text{ns} = 0.06 \mu\text{s}$ ). Therefore, the filtration process takes  $127.38 \mu\text{s}$  ( $63.66 \mu\text{s} + 0.06 \mu\text{s} + 63.66 \mu\text{s}$ ), based on 6369 clock cycles.

2. *Ping-Pong Dual-Port Memory (PPDM)*: The PPDM is the interface memory, illustrated in Fig. 13, between the filtration module and the back-projection module. This memory includes two internal RAMs;  $M_1$  and  $M_2$ . It is designed to let the current filtered projection  $Pf_i$  be written to  $M_1$  while the previous projection  $Pf_{i-1}$  is back-projected from  $M_2$ . Therefore, the filtration does not need to wait until the BP finishes, which enhances the speed of the design implementation. After that projection  $Pf_i$  is read from  $M_1$  while projection  $Pf_{i+1}$  is written to  $M_2$ . Therefore, the memory PPDM is called ping-pong memory. When one of the internal RAM is in the read mode, the other will be in the write mode. In the write mode, the RAM is write-enabled and addressed with  $Pf\_ind(9:0)$ . In the read mode and from equation (5), the BP reads two successive samples simultaneously, so the RAM needs two data outputs read from two addresses. The PPDM enables the BP to simultaneously read two successive samples. The address inputs of the PPDM,  $rd\_add0(9:0)$  and  $rd\_add1(9:0)$ , are fed from the address outputs  $add\_0(9:0)$  and  $add\_1(9:0)$  of the BP module, respectively. The data outputs from the PPDM,  $rd\_d0(15:0)$ , and  $rd\_d1(15:0)$ , simultaneously feed the BP module.

3. *Back-projection*: Direct implementation of equation (5) consumes two multipliers. This equation can be modified to:

$$imm_\theta = P_\theta(a_\theta) + (T_\theta - a_\theta)(P_\theta(a_\theta + 1) - P_\theta(a_\theta)) \quad (13)$$

This modification reduces the resources to one multiplier only. The hardware implementation of this process is done using three blocks as shown in Fig. 15. The first block ( $T\_gen$ ) that generates matrix  $T_\theta$  with its integer part  $tint(10:0)$  and fraction part  $tfr(13:0)$ , the second block ( $Proj\_address$ ) uses the integer part for addressing of the PPDM containing  $P_{fil}$ , and the third block ( $BP\_interp$ ) uses the fraction part as the BP interpolation factor ( $T_\theta - a_\theta$ ).

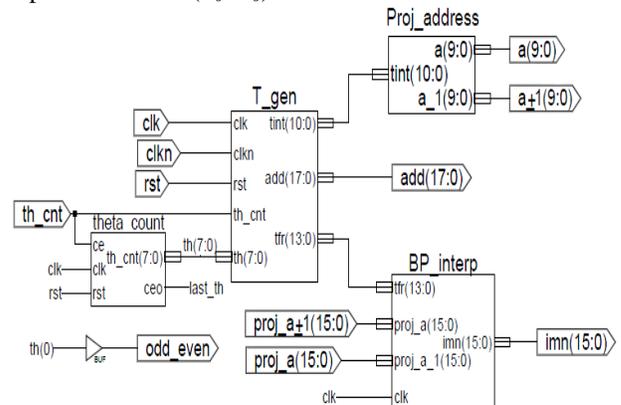


Fig. 15. Schematic diagram of the back-projection.

Matrix generator ( $T\_gen$  shown in Fig. 15) generates matrix  $T_\theta$  by direct hardware realization of equation (4). Two counters (X and Y) generate the Cartesian coordinates starting from the top left pixel of the image in a zigzag direction.

Counter X starts from -256 to 255, while counter Y starts from 255 to -256. Counter Y is enabled when counter X ends to 255, to go to the next row and counter X starts from -256 again. Counter X and counter Y format is 9 integer bits. The cosine and sine values of the projection angles are stored in two ROMs in a (1, 1, 14) format. The outputs of two counters (X and Y) and two ROMs (COS & SIN) are asserted at the rising edge, while the multiplication is done at the falling edge to emphasize enough setup and hold time for the multiplier. The resultant  $T_\theta$  is 25 bits; with fixed point format (1, 10, 14). From equation (13),  $T_\theta$  will be used as two separate parts; the integer part ( $tint$ ) and the fraction part ( $tfr$ ). An 18-bit counter is enabled for each  $T_\theta$  to generate the address ( $add(17:0)$ ) corresponding to each pixel from 0 to  $512^2$ . 0 is the address of the pixel that has coordinates (-256, 255) which is the first pixel in the zigzag, while address  $512^2$  corresponds to last pixel with coordinates (255, -256).

The block *Proj\_address*, shown in Fig. 15, uses the integer part  $a_\theta$  to address its corresponding projection samples in the *PPDM*. The integer part of matrix  $T_\theta$  ( $tint$ ) is supposed to be a matrix of positive values resembling addresses but actually it has negative values, so the center of  $tint$  has to be shifted by adding a positive value to all of  $tint$  elements. This value equals to the minimum possible negative value plus one (not to have zeros), which is the left bottom corner of  $T_{45} = -256 \times \cos(45^\circ) - 256 \times \sin(45^\circ) \approx -363$ . The added value is  $363 + 1 = 364$ , i.e.  $a_\theta = tint + 364$ . The addressing part of the BP, *Proj\_address*, generates two successive addresses ( $a_\theta, a_\theta + 1$ ) to the *PPDM* to read two successive samples ( $proj(a_\theta), proj(a_\theta+1)$ ). These samples have to be interpolated, discussed in section III.

The BP with linear interpolation (*BP\_interp*) is implemented using equation (13). Two inputs projection samples, read from the *PPDM*, have (1, 4, 11) format. These two samples are subtracted, and the output of the subtractor has the same format. The interpolation factor ( $tfr$ ) format is (0, 0, 14). This factor is multiplied by the output of the subtractor. The multiplication output format is (1, 4, 25), rounded to have the same format of the projection sample  $proj(a_\theta)$ ; (1, 4, 11). Then, this output is added to the projection sample. Each output of the adder represents an image pixel with (1, 4, 11) format that forms the desired back-projected image,  $imn$ . Image  $imn$  is generated pixel by pixel with its corresponding address. To form an image with 512-by-512 pixel, it takes  $512^2$  clock cycles. That image takes 5.243 ms using 50 MHz clock cycles.

### C. Image Memory Accumulator

The new back-projected image is added to the previous ones through image accumulator memory. The accumulator consists of two RAMs. The first image is written to a RAM (odd RAM). Then the second image is added to the image stored in the odd RAM and the result is written to the other RAM (even RAM). The third image is added to the image stored in the even RAM and the result is written to the odd RAM and so on. In general, the new image is added to the sum of the previous images stored in a RAM, and the result is

written to the other RAM under the control of the image memory controller, *img\_mem\_cont*, shown in Fig. 11. Therefore, when a RAM is in read mode, the other is in write mode as shown in Fig. 16.

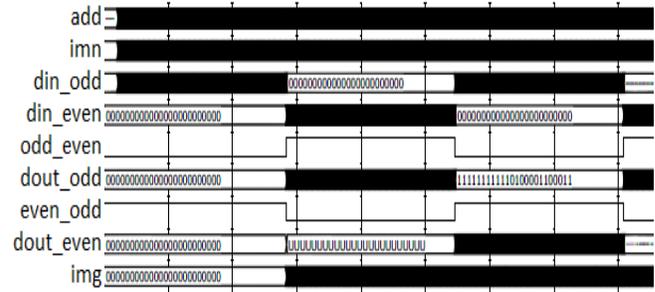


Fig. 16. Timing diagram of first Image accumulation process.

When a new odd image,  $imn$ , is generated from the BP, the signal,  $odd\_even$ , sets to low, and the data is read from the even RAM ( $dout\_even$ ) which is the sum of the previous images. The odd  $imn$  is added to this sum and the new sum is written to the input of odd RAM ( $din\_odd$ ). Next, when an even image is generated, the control selections,  $odd\_even$  and  $even\_odd$ , are inverted and so on. To avoid the overflow due to the sum accumulation, the least significant four bits of pixel value of the image  $imn$  are neglected. So the accumulated image format is (1, 8, 7). The effect of neglecting these bits will be studied in the next subsection by comparing the output image from the hardware (with four bits reduction) to the corresponding image from the CTSIM (without four bits reduction). The accumulation is simultaneously done with the BP process.

### D. Results

After the accumulation of all images, the last accumulated image is exported quantized to its 16-bit fixed point format (1, 8, 7). The data are arranged as 1-D from address 0 to  $512^2$ , so it is rearranged from 1-D series of data into a 2-D pixel by pixel matrix using the zigzag scheme. The output image is shown in Fig. 17.

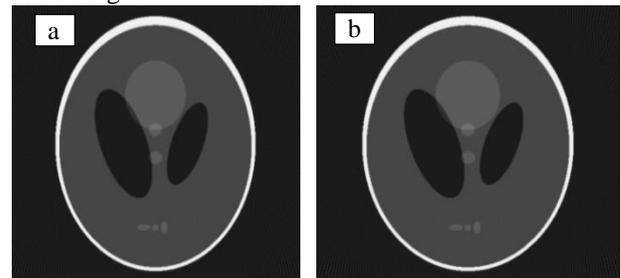


Fig. 17. Output image from (a) Software based on the CTSIM (b) Hardware implementation (16-bit).

Finally, the image generated by hardware implementation is compared to the image generated by the CTSIM. This comparison is evaluated based on the accuracy of the image quality. The image quality is tested in two ways. The first way is the subjective test that visually compares between the reconstructed images, based on the CTSIM and the hardware implementation, shown in Fig. 17. The second way is the objective test that contains the PSNR, the MSSIM

(Percentage), the ABS and the WORST, shown in Fig. 18. As mentioned before, Matlab Shepp-Logan phantom is used as a reference image. For more information of the test image used, how it was obtained and its projection data was obtained, refer to [22].

It is obvious that the subjective test and the objective test are almost the same that indicates that the design is functionally successful.

The BP process takes  $512^2$  clock cycles. So, there is enough time to acquire and filter the next projection. The acquisition

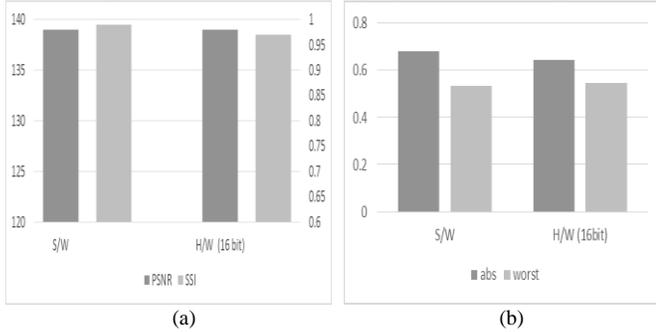


Fig. 18. Software based CTSIM vs. hardware implementation (16-bit): (a) Image quality b) Errors measurement.

and filtration of the next projection is synchronized to let the BP of this projection start just after the BP of the current one is completed. That saves time and let the overall time limited to the BP time of all projections plus the filtration time of the first projection only. The summation of the back-projected images is executed with the BP pixel by pixel. The exemplar timing diagram of the 1<sup>st</sup>, 2<sup>nd</sup>, intermediate and last projection processing is shown in Fig. 19. The time required to produce the final image, *img*, is the required time to perform the BP process multiplied by the number of projections ( $5.243 \text{ ms} \times 1024$ ) plus the first projection filtration time ( $127.38 \mu\text{s}$ ). The total reconstruction time of the non-parallel hardware implementation is 5.37s.

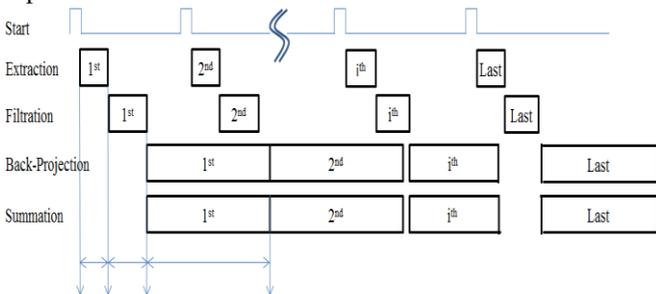


Fig. 19. Timing diagram of the implemented algorithm.

Now, the non-parallel hardware design presented in this section is ready to the timing simulation. The presented design is synthesized, mapped, placed and routed on the targeted chip Spartan 3A-DSP 3400. Device utilization is shown in Table I.

### E. Optimization

In this section, the optimum hardware utilization will be considered without affecting the reconstruction time and the image quality. It is done based on hardware reduction of the FFT IP core, the reduction of the data-bus width, and the efficient utilization of the chip supplied benchmarks, DSP and

RAM Blocks. The time reduction through the parallelism of the image reconstruction process will be presented in the next section.

TABLE I  
DEVICE UTILIZATION SUMMARY ON XC3SD3400A

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	25,448	47,744	53%
Number of 4 input LUTs	38,857	47,744	81%
Number of occupied Slices	22,890	23,872	95%
Total Number of 4 input LUTs	39,956	47,744	83%
Number of bonded IOBs	144	469	30%
Number of RAMB16BWERs	11	126	8%
Number of DSP48As	0	126	0%

1. *FFT IP Core Implementation Architecture*: However, the filtration, the BP, and the image accumulation are done simultaneously, it is noticed that the filtration time is much smaller than the time required for the BP or the image accumulation. Therefore, the filtration can be performed slower to get the benefit of the hardware reduction without affecting the overall speed or image quality. Radix-2 lite FFT core is implemented with much smaller resources at the expense of additional latency [26]. This latency will not affect the BP time. The FFT latency is  $246.44 \mu\text{s}$  ( $12320$  clock cycles), so the filtration time (FFT latency + Multiplication time + IFFT latency) will be  $246.4 + .06 + 246.4 = 492.86 \mu\text{s}$  ( $24643$  clock cycles). The BP time is  $5.243 \text{ ms}$ , so the overall speed is not affected while we gain a considerable reduction in the utilized resources of the target chip. The device utilization is shown in Table II.

TABLE II  
DEVICE UTILIZATION SUMMARY WITH FFT LITE ARCHITECTURE.

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	4,699	47,744	9%
Number of 4 input LUTs	11,653	47,744	24%
Number of occupied Slices	7,215	23,872	30%
Total Number of 4 input LUTs	11,820	47,744	24%
Number of bonded IOBs	144	469	30%
Number of RAMB16BWERs	11	126	8%
Number of DSP48As	0	126	0%

2. *Data-Bus Width*: Another issue that reduces the utilized hardware area but may affect the image quality is the width of the data bus. This width is reduced from 16 bits to 9 bits. Therefore, the effect of the bus-width reduction on the image quality needs to be studied. This effect is tested by comparing the reconstructed image before and after the bus-width reduction. The Subjective test is shown in Fig. 20, and the objective test is shown in Fig. 21. It is obvious that the image quality is greatly affected by the bus-width reduction from 16 to 9 bits. This effect is in both the subjective and the objective test.

To overcome this problem, we used the CTSIM to analyze the data values through all the reconstruction stages. It is found that the data is enlarged by the image accumulation process. A proposed solution is to use different data bus-width before and after the image accumulation. The 9-bit bus-width

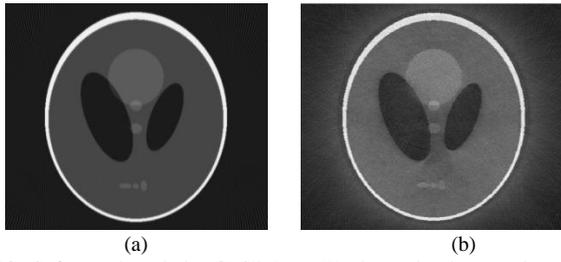


Fig. 20. Software based the CTSIM vs. Hardware implementation (9-bit): a) Original image b) Reconstructed.

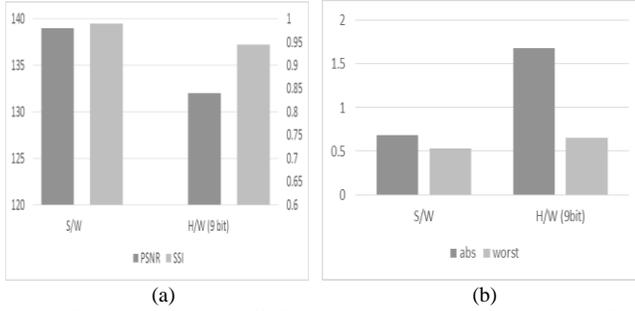


Fig. 21. Software based the CTSIM vs. Hardware implementation (9-bit): a) Image quality b) Error measurement.

represents the data before the image accumulation while, after the accumulation, the bus-width is increased by 3 bits. This 9/12 design is subjectively and objectively tested shown in Fig. 22 and Fig 23, respectively.

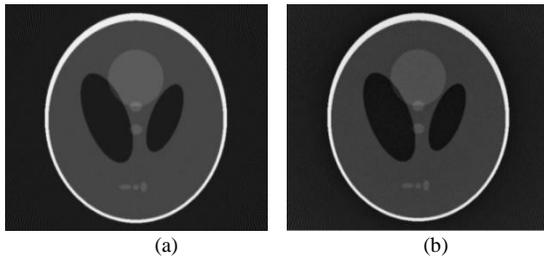


Fig. 22. Software based the CTSIM vs. Hardware implementation (9/12 bit): a) Original image b) Reconstructed.

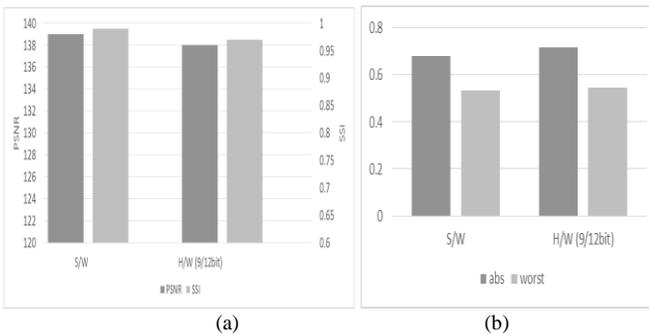


Fig. 23. Software based the CTSIM vs. Hardware implementation (9/12 bit): a) Image quality b) Error measurement.

This analysis illustrates that the image quality is not affected by this reduction. It is obvious that the reconstruction quality in the subjective test and the objective test are very close. At the same time, the utilized hardware area is significantly reduced. The device utilization summary is shown in Table III.

TABLE III  
DEVICE UTILIZATION SUMMARY OF 9/12 BUS-WIDTH DESIGN.

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	3,102	47,744	6%
Number of 4 input LUTs	7,163	47,744	15%
Number of occupied Slices	4,716	23,872	19%
Total Number of 4 input LUTs	7,401	47,744	15%
Number of bonded IOBs	156	469	33%
Number of RAMB16BWERs	11	126	8%
Number of DSP48As	0	126	0%

3. *The Exploitation of supplied benchmarks:* Another optimization issue is the optimum usage of the supplied benchmarks like DSP and RAM blocks. When implementing the FFT core, the complex multipliers and the butterfly arithmetic of the core can be implemented using DSP blocks [26] and the buffering memories can be implemented using the RAM blocks. Also, other multipliers and memories in the design can be implemented using these benchmarks that provides the advantage of saving more area, shown in Fig. 24, and maximize the utilization of the already supplied benchmarks without affecting the overall speed or image quality. The device utilization is shown in Table IV.

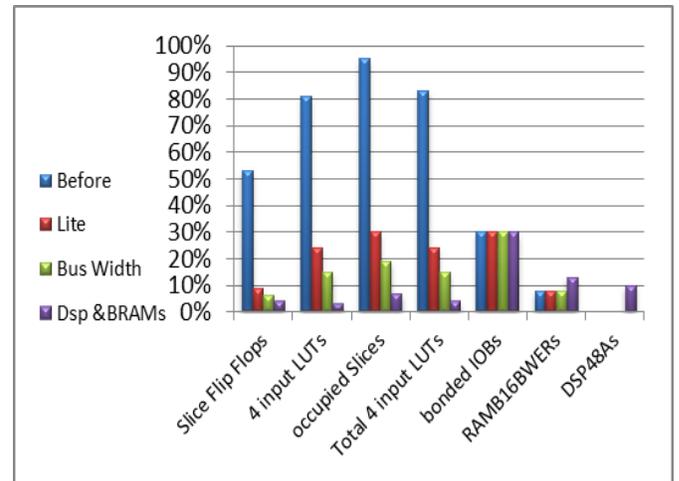


Fig. 24. Different fields of optimization to reduce the hardware utilization.

TABLE IV  
DEVICE UTILIZATION SUMMARY OF THE ALREADY SUPPLIED BENCHMARKS.

Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	1,910	47,744	4%
Number of 4 input LUTs	1,738	47,744	3%
Number of occupied Slices	1,814	23,872	7%
Total Number of 4 input LUTs	1,914	47,744	4%
Number of bonded IOBs	156	469	33%
Number of DSP48As	17	126	13%
Number of RAMB16BWERs	13	126	10%

## V. PARALLEL HARDWARE IMPLEMENTATION OF FBP

FPGA devices provide fine-grained parallelism. Therefore, the time reduction from parallel processing of the FBP algorithm can be achieved. Due to the small hardware utilization achieved, the FBP algorithm core can be redesigned to achieve the desired parallelism and fit to the same chip

(Spartan 3A-DSP 3400). It can be implemented in two schemes of parallelism; *pixel parallelism*, and *projection parallelism*.

In the *pixel parallelism*, the image is divided into multiple disjoint segments. Each segment will be reconstructed through BP block, image accumulation controller and image accumulation RAM. Each BP block will generate its own sub-matrix of  $T_\theta$  corresponding to its segment. These segments are arranged together to form the resultant image. Due to the small hardware overhead of the BP block, the total utilized area will not be greatly affected, while the BP time will be reduced by a factor equal to the number of the segments. The limitations of this parallelism are the increased number of input/output pins and the minimum time of BP which should be more than the filtration time based on the lite FFT architecture.

In the *projection parallelism*, the BP images from different projections are independent so projections can be divided into multiple groups. Each group of projections is processed separately while all groups are simultaneously processed through multiple FBP blocks. The reconstructed images from these blocks are added together to form the final reconstructed image. The total reconstruction time will be reduced by a factor equal to the number of groups. This parallelism is limited by the size of the chip as the design is almost repeated with each projection group, while the problems of limited input/output pins and minimum BP time do not exist.

From the above discussion, it is noticed that the pixel parallelism is more efficient in the hardware utilization than the projection parallelism as long as the design is away from its limitations. Therefore, the pixel parallelism is first designed to get the maximum available parallelism. After that the projection parallelism is designed. In this approach, the limitations of two schemes are avoided and their advantages are exploited. The 512-by-512 image is divided into eight disjoint segments with size of 256-by-128 pixels. The first and last value of the eight X and Y counters generating matrix  $T_\theta$  can be easily achieved. In Fig. 25, eight BP blocks, image sum controllers and image sum memory are used in parallel to produce eight reconstructed image segments. These segments are arranged to form the final reconstructed image. The device utilization summary of this parallelism is shown in Table V.

TABLE V  
DEVICE UTILIZATION OF 8 SEGMENTS OF PIXEL PARALLELISM ON XC3SD3400A.

Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	2,351	47,744	4%
Number of 4 input LUTs	2,571	47,744	5%
Number of occupied Slices	2,398	23,872	10%
Total Number of 4 input LUTs	2,871	47,744	6%
Number of bonded IOBs	447	469	95%
Number of BUFGMUXs	3	24	12%
Number of DSP48As	40	126	31%
Number of RAMB16BWERs	27	126	21%

Although there is enough area to implement more than eight segments, the limited number of input/output pins of the chip Spartan 3A-DSP 3400 and the projection filtration time (492.86  $\mu$ s) limit the pixel parallelism to only eight BP blocks. This pixel parallelism reduces the BP time to be 655.36  $\mu$ s (256 x 128 clock cycles) instead of 5.24 ms (512<sup>2</sup> clock cycles). Finally, the BP time based on the pixel parallelism is reduced by the factor of 8.

The parallelism will be continued by the projection parallelism. Input projections (1024 projections) are divided into five groups; each group contains 205 projections. Five groups are separately and simultaneously processed through five FBP blocks. Each FBP block is 8-segments of the pixel parallelism. The block diagram of this scheme is shown in Fig. 26. The total reconstruction time of the projection parallelism has the time of the BP ( $t_{BP}$ ) multiplied by the number of projections ( $N_{proj}$ ) plus the time of the first projection filtration ( $t_{filt}$ ). Therefore, the presented projection parallelism in this section reduces the total reconstruction time to be 205  $t_{BP} + t_{filt}$  instead of the 1024  $t_{BP} + t_{filt}$ . It is reduced by factor of 5 in the projection parallelism beside to the reduction of  $t_{BP}$  by factor of 8 in the pixel parallelism. The total reconstruction time ( $t_R$ ) is calculated according to equation (14).

$$t_R = \left( \frac{t_{BP}}{8} \times \frac{N_{proj}}{5} \right) + t_{filt} \quad (14)$$

By substitution in equation (14) with the values of  $t_{BP}$ ,  $N_{proj}$ , and  $t_{filt}$ , it implies the following calculations.

$$\begin{aligned} t_R &= ((5243 / 8) \times (1024/5)) + 492.94 \\ &= (655.36 * 205) + 492.94 = 134.8 \text{ ms.} \end{aligned}$$

The  $t_R$  is reduced by the factor of 40. Before the grouping of input projections, the 8-segment pixel parallelism is optimized to restrict the hardware utilization into less than 20% of the chip to permit the implementation of 5-group projection parallelism. The device utilization summary is shown in Table VI.

TABLE VI  
DEVICE UTILIZATION OF 8-SEGMENT PIXEL PARALLELISM AND 5-GROUP PROJECTION PARALLELISM.

Logic Utilization	Used	Available	Utilization
Total Number Slice Registers	11,801	47,744	24%
Number of 4 input LUTs	16,791	47,744	35%
Number of occupied Slices	13,217	23,872	55%
Total Number of 4 input LUTs	18,397	47,744	38%
Number of bonded IOBs	469	469	100%
Number of BUFGMUXs	3	24	12%
Number of DSP48As	120	126	95%
Number of RAMB16BWERs	126	126	100%



65MHz. The required number of clock cycles ( $N_{CLK}$ ) in the design presented in [11] is  $16.25 \times 10^6$  clock cycles ( $N_{CLK} = t_R/t_{CLK}$ , where  $t_{CLK}$  is the period of the chip clock). In this paper, the combination of the pixel parallelism and projection parallelism is utilized using low-cost Spartan 3A-DSP 3400 chip with 50MHz clock cycles. It achieves image reconstruction of a 512-by-512 pixel image from 1024 projections in 134.8 ms. The  $N_{CLK}$  in the design presented in this paper is  $6.47 \times 10^6$  clock cycles. Therefore, it achieves the reduction of the required number of clock cycles to form an image from projections by 60 % comparing to the state of the art in [11]. The presented FPGA hardware implementation of the parallel-beam FBP algorithm, based on the pixel parallelism and projection parallelism, is achieved the superiority in terms of the speed and the hardware cost with respect to all previous published works.

## VI. CONCLUSION

In this paper, the FPGA hardware implementation of the parallel-beam FBP algorithm, based on the pixel parallelism and projection parallelism, was presented. First, the non-parallel hardware was implemented in three main blocks; filtration, BP and dual-port memory. The filtration block receives one by one of the 1024 projections and exports the filtered projection to the dual-port memory from which the BP block read it. The images are externally accumulated to reconstruct the final a 512-by-512 pixel image. The BP and the filtration processes are overlapped to shrink the total reconstruction time to almost the BP time of all projections only.

In addition, the optimization between the speed and the hardware utilization without affecting the image quality was presented. The reduction of the hardware utilization of the implemented algorithm on the target chip can be achieved through the hardware reduction of the FFT IP core, the efficient utilization of the DSP and RAMs blocks of the target chip, and the reduction of the data-bus width. By suitable rounding at each stage, the bus-width is minimized from 16 to 9/12 bit that significantly reduced the hardware utilization. The reduction of the reconstruction time can be achieved through the pixel and projection parallelism. Eight segments of the pixel parallelism are implemented that achieved reconstruction time reduction by factor of 8. Although there is enough area to implement more than eight segments, the limited number of input/output pins of the target chip and the projection filtration time limits the parallelism to only eight BP blocks. In addition, the parallelism is extended by extra five groups of the projection parallelism which in turn reduce the reconstruction time by factor of 5. The combination of the two schemes of parallelism reduces the reconstruction time by factor of 40. It achieves image reconstruction of a 512-by-512 pixel image from 1024 projections in 134.8 ms using low-cost Spartan 3A-DSP 3400 chip with 50MHz clock cycles. The number of clock cycles in the presented design to form a 512-by-512 pixel image from 1024 projections is  $6.47 \times 10^6$  clock cycles. It reduces the number of clock cycles by 60 % comparing to the state of the art in [11].

Before hardware implementation, a software simulator CTSIM is built to analyze and get accurate validation of each reconstruction process, used as a reference to validate each stage in the hardware design. The image generated from hardware implementation is imported to the CTSIM for the comparison with the image generated from the software. This comparison is tested either in subjective manner or in objective manner. In addition, the CTSIM assists the hardware implementation to be optimized and minimized the hardware utilization.

The presented design can be easily expanded to accommodate more projections with higher densities and to produce larger images. Due to re-configurability of the FPGA, the design can be easily edited to accommodate the fan-beam FBP. That can be done in two approaches; the first is to reformat the fan-beam projections into a set of parallel-beam projections. This reformat is done using an extra module called the rebinning module; the other approach is the use of the modified back-projection algorithm. Where some steps of the FBP is modified to reconstruct the image from the fan-beam projections.

## REFERENCES

- [1] G. T. Herman, *Image Reconstruction from Projections: Fundamentals of Computerized Tomography*, Academic Press, 1980.
- [2] Kak, A.C., and Slaney, M., *Principles of Computerized Tomographic Imaging*, IEEE Press, 1988.
- [3] D. U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, 3<sup>rd</sup> Edition, Florida: Springer, 2007.
- [4] J. Xu, N. Subramanian, A. Alessio, and S. Hauck, "Impulse C vs. VHDL for Accelerating Tomographic Reconstruction," *18<sup>th</sup> IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*, pp. 171-174, 2-4 May 2010.
- [5] Agi, I., Hurst, P.J., and Current, K.W., "A VLSI architecture for high-speed image reconstruction: considerations for a fixed-point architecture," *Proceedings of SPIE, Parallel Architectures for Image Processing*, vol. 1246, pp. 11-24, 1990.
- [6] Wu, M. A., "ASIC Applications in Computed Tomography Systems," *Proceedings of 4<sup>th</sup> Annual IEEE International ASIC Conference and Exhibit*, Rochester, NY, pp. P1-3/1-4, USA, 1991.
- [7] Agi, I., Hurst, P.J., and Current, K.W., "An image processing IC for back-projection and spatial histogramming in a pipelined array," *IEEE Journal of Solid-state Circuits*, vol. 28, no. 3, pp. 210-221, March 1993.
- [8] Stephen G. Azevedo, Brian K. Cabral and J. Foran, "Tomographic image reconstruction and rendering with texture-mapping hardware," *SPIE Proceedings of Mathematical Methods in Medical Imaging III*, vol. 2299, pp. 280-289, July 1994.
- [9] Chen, Chung-Ming, Cho, Zang-Hee, and Wang, Cheng-Yi, "A Fast Implementation of the Incremental Back-projection Algorithms for Parallel Beam Geometries," *IEEE Transactions on Nuclear Science*, vol. 43, no. 6, pp. 3328-3334, Dec. 1996.
- [10] Bins, J., Draper, B., Bohm, W., and Najjar, W., "Precision vs. Error in JPEG Compression," *SPIE proceeding of Parallel and Distributed Methods for Image Processing III*, Denver CO, pp.76-87, Oct. 1999.
- [11] M. Leeser, S. Coric, E. Miller, H. Yu, and M. Trepanier, "Parallel-Beam Back-projection: an FPGA Implementation Optimized for Medical Imaging," *Journal of VLSI signal processing*, vol. 39, no. 3, pp. 295-311, March 2005.
- [12] Johann Radon, "Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten", *Berichte über die Verhandlungen der Königlich-Sächsischen Akademie der Wissenschaften zu Leipzig, Mathematisch-Physische Klasse* [Reports on the proceedings of the Royal Saxonian Academy of Sciences at Leipzig, mathematical and physical section], Leipzig: Teubner, vol. 69, pp. 262-277, 1917; Translation: Radon, J.; Parks, P.C. (translator) (1986), "On the

- determination of functions from their integral values along certain manifolds", *IEEE Transactions on Medical Imaging*, vol. 5, no. 4, pp. 170-176, 1986.
- [13] N. Hounsfield, "Computerized transverse axial scanning (tomography)," *Description of system, British Journal of Radiology*, vol. 46, pp. 1016-1022, 1973.
- [14] A. M. Cormack, "Reconstruction of densities from their projections, with applications in radiological physics," *Physics in Medicine and Biology*, vol. 18, no. 2, pp. 195-207, March 1973.
- [15] Gabor T. Herman, *Fundamental of Computerized Tomography – Image reconstruction from projections*, Second edition, Springer, 2009.
- [16] Edwin L. Dove, *Notes on Computerized Tomography*, Univ. of Iowa, 2001.
- [17] V. Thakur, A. K. Verma, P. Jena and G. S. Prasad, "Design and implementation of FPGA based Digital Pulse Compression via fast convolution using FFT-OS method," *International Conference on Microwave, Optical and Communication Engineering (ICMOCE)*, Bhubaneswar, pp. 455-458, 2015.
- [18] Thingom and P. Khundrakpam, "FPGA implementation of FIR filter using RADIX-2r," *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, pp. 1524-1528, 2016.
- [19] J. C. Russ, *The Image Processing Handbook*, 5<sup>th</sup> edition, North Carolina, CRC Press, 2007.
- [20] R. W. S. a. L. R. Rabiner, "A digital signal processing approach to interpolation," *Proceedings of IEEE*, vol. 61, no. 6, pp. 692-702, June 1973.
- [21] J. Hsieh, *Computed tomography: principles, design, artifacts, and recent advances*, Washington USA, SPIE and John Wiley & Sons, Inc., 2009.
- [22] Bassam Abd-Elwahab, "CT Image Reconstruction using FPGA", M.Sc. thesis, Military Technical College, Cairo, Egypt, 2012.
- [23] M. Kevin M. Rosenberg, *CT Sim 3.5 User Manual*, New Mexico, 2002.
- [24] L.A. Shepp, B.F. Logan, "The Fourier Reconstruction of a Head Section," *IEEE Transactions on Nuclear Science*, vol. 21, no. 3, pp. 21-43, June 1974.
- [25] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli, "Image quality assessment: From error measurement to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, April 2004.
- [26] LogiCORE IP, FFT v7.1, DS260 Product Specification. [Performance]. Xilinx, April 19, 2010.

# CFOA-Based Fractional Order $PI^{\lambda}D^{\delta}$ Controller

Tada Comedang and Pattana Intani

**Abstract**— Conventional Current Feedback Operational Amplifier (CFOA) is not current controllable or not electronically controllable. It is thus of interest to add a current mirror into the CFOA in order to make it current controllable. This modification can be achieved by using Diamond Transistor (DT) instead of going through complicated IC fabrication process. This work applies the modified CFOA in fractional-order proportional integral derivative ( $PI^{\lambda}D^{\delta}$ ) controller. Both simulation and experimental results confirm that the modified CFOA is electronically controllable.

**Index Terms**— CFOA, Fractional-order proportional integral derivative.

Original Research Paper  
DOI: 10.7251/ELS1721025C

## I. INTRODUCTION

SEVERAL active elements for analogue signal processing have recently been proposed. Some applications of these components have been given in the literature, for example, differentially buffered transconductance amplifiers (DBTAs) [1], current differencing transconductance amplifiers (CDTAs) [2], current follower transconductance amplifiers (CFTAs) [3], current conveyor transconductance amplifiers (CCTAs) [4-8], differential difference current conveyor (DDCC) [9-10] and others. Unfortunately, for the most part, development of the applications will be done via a simulation program with a transistor model of the active components of some bipolar or CMOS technology where practical usability is questionable. Attributable, experimental verification via their on-chip fabrication is expensive and time-consuming [11].

Among the mentioned active elements, the current feedback operational amplifier (CFOA) is an interesting active component, especially suitable for a class of analogue signal processing [12-14]. This device can operate in both current and voltage modes, provides flexibility and enables a variety of circuit designs. In addition, it can offer advantageous

features, such as high slew rate, freedom from parasitic capacitances, wide bandwidth and simple implementation [15-18].

Nowadays, the CFOA can be found commercially, for example, the AD844 from Analog Devices Inc. [13]. However, the CFOA cannot be controlled by the electronic controllability of the hysteresis of the output signal. The electronic control method has become more popular more than those by passive elements (i.e., resistors and capacitors) due to how it can easily be adapted to automatic or microcontroller-based controls. Diamond transistors (DT) are readily available commercially (commercially marked OPA 860) [19] and belong to the group of well-known commercial products commonly used for wide-bandwidth systems, including high performance video, RF and IF circuitry. It includes a wideband, bipolar operational transconductance amplifier (OTA) and voltage buffer amplifier. The transconductance of the OPA860 can be adjusted with an electronic control, allowing bandwidth, quiescent current, and gain trade-offs to be optimized. Used as a basic building block, the OPA860 simplifies the design of AGC amplifiers, LED driver circuits for fibre optic transmission, integrators for fast pulses, and fast control loop amplifiers and control amplifiers for capacitive sensors and active filters. Concrete experiments have led to the observation that the OPA860 is a useful element for analogue signal processing in the frequency range of units and tens of wide bandwidth.

Proportional integral derivative (PID) control has been adopted in many engineering applications [20-22]. Recently, several literature reviews have studied mechanical systems described by fractional-order state equations [23-25], i.e., equations involving so-called fractional derivatives and integrals [26-28]. A fractional-order  $PI^{\lambda}D^{\delta}$  controller, first proposed by Podlubny, is a generalization of a PID controller, involving an integrator of order and a differentiation of order [29]. Expanding derivatives and integrals to fractional orders can adjust a control system's frequency response directly and continuously. This great flexibility makes it possible to design more robust control systems [30]. Several methods have been reported for fractional-order  $PI^{\lambda}D^{\delta}$  design. At the present time, there are numerous methods for the approximation of fractional derivatives and integrals, and fractional calculus can be easily used in a wide variety of applications (e.g. control theory, new fractional controllers, system models, electrical circuits theory, fractances, capacitor theory, etc.) [31-49].

Existing evidence has confirmed that the best fractional-

Manuscript received 15 December 2015. Received in revised form September 14, 2016. Accepted for publication December 28, 2016.

T. Comedang is with the Faculty of Electrical Engineering, Pathumwan Institute of Technology, Bangkok, Thailand (phone: +668-24-520-041; e-mail: tada-comedang@hotmail.com).

P. Intani is with the Faculty of Electrical Engineering, Pathumwan Institute of Technology, Bangkok, Thailand (pattana@pit.ac.th).

order controller can outperform the best integer-order controller. It has also been answered in the literature why fractional-order control should be considered even when integer (high)-order control works comparatively well [50-51]. Fractional-order  $PI^{\lambda}D^{\delta}$  controller tuning has reached a matured state of practical use. Because (integer-order) PID control dominates the industry, we believe that fractional-order  $PI^{\lambda}D^{\delta}$  will gain increasing impact and wide acceptance. Furthermore, we also believe that, based on some real world examples, fractional-order control is ubiquitous when the dynamic system is of a distributed parameter nature [52].

In this paper, the conception of this active element for experimental purposes is built from commercially available devices. There is the diamond transistor and a wideband voltage buffer designed (OPA860) for positive and negative  $W$  terminals. Note that the manufactured CFOA [53] does not provide this feature. It is necessary to have two positive current outputs for the proposed fractional-order  $PI^{\lambda}D^{\delta}$ . The CFOA is applied for fractional-order  $PI^{\lambda}D^{\delta}$  design. To match the criteria of industrial applications, the best fractional-order controller can outperform the best integer-order controller. Similarly, a circuit exhibiting fractional-order behaviour is called a fractance. The design of fractances can be performed easily using any of the rational approximations or a truncated continued fraction expansion (CFE), which also gives a rational approximation. Generally speaking, there are three basic fractance devices. The most popular is a domino ladder circuit network. Very often used is a tree structure of electrical elements [54]. Therefore, fractional-order  $PI^{\lambda}D^{\delta}$  causes the result  $0 < \lambda, \delta \leq 1$ , and the fractional order is approximated using a fractance circuit.

## II. CFOA ELEMENT BASED ON OPA860

In Fig. 1, the schematic symbol of the CFOA, which was given in the paper [53], has been shown with an additional output terminal of  $W^-$ , which is the negative of the  $W$  terminal. The proposed behavioural model of the CFOA is given in Fig. 1, where  $Y$  and  $X$  are input terminals and  $Z$ ,  $W$ , and  $W^-$  are output terminals. Its definition is shown in the matrix as follows (1):

$$\begin{bmatrix} V_X \\ V_Y \\ I_Z \\ V_{W^+} \\ V_{W^-} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1/g_{m1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & g_{m2} & 0 & 0 \end{bmatrix} \begin{bmatrix} I_X \\ V_Y \\ V_Z \\ I_{W^+} \\ I_{W^-} \end{bmatrix}, \quad (1)$$

where  $g_m$  represents the transconductance gain of the CFOA. For a CFOA implemented with OPA860,  $g_m$  can be

$$g_{m1} = \frac{1}{R_X + (1/I_{Q1})}, \quad (2)$$

$$g_{m2} = \frac{R_L}{R_E + (1/I_{Q2})}. \quad (3)$$

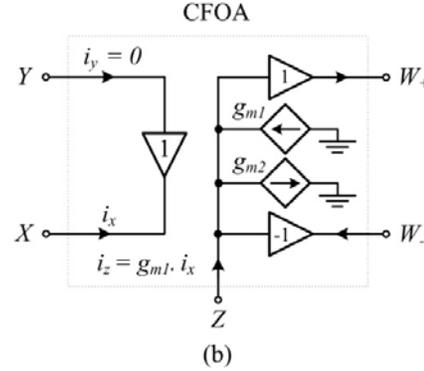
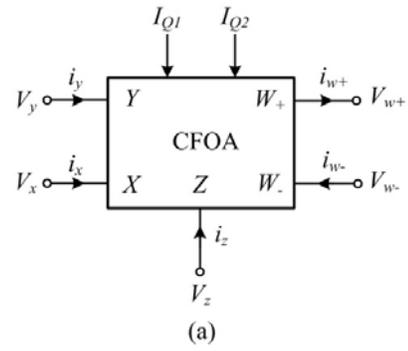


Fig. 1. CFOA elements' circuit symbol. (a) schematic symbol (b) equivalent circuit.

From the advantage already mentioned in the introduction, CFOA can be constructed from commercially available devices, as is shown in Fig. 2. It consists of the diamond transistors (OPA 860), where  $g_m$  represents the amplifier gain of the CFOA for a negative  $W$  terminal ( $W^-$ ) implemented with diamond transistors. In this paper, the idea to develop the features of the CFOA increased by the using the Buffer of OPA860, which can be useful in the work for the proposed FO-PID voltage mode.

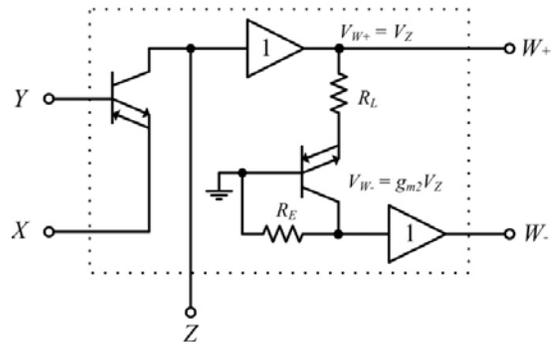


Fig. 2. The CFOA using commercially available devices numbers OPA 860.

The principle negative  $W$  terminal of CFOA. Fig. 3 through Fig. 8 show the main features obtained by simulations of the CFOA element shown in Fig. 2. The result presented in Fig. 3 is that the voltage transfer between the  $Y$  and  $X$  ports of the proposed CFOA is approximately 185 MHz, and current transfers between the  $X$  and  $Z$  ports being approximately 81 MHz is shown in Fig. 4. The simulations results lead to the

observation that the CFOA is a useful element for the frequency range of units and tens of megahertz (MHz). Later, Fig. 5 through Fig. 8 show the DC transfer characteristic between the  $Z$  and  $W^+$  ports. It also shows the DC transfer characteristic between the  $Z$  and  $W^-$  ports. This transconductance gain can be controlled by the bias current  $I_{Q1}$  and adjust the gain adjustable resistor ( $R_L$ ) from  $100\ \Omega$  to  $500\ \Omega$ , and  $I_{Q2} = 11.2\ \text{mV}$ . The results presented show high linearity input voltage ( $Z$ ) from  $-4$  to  $4$  volts.

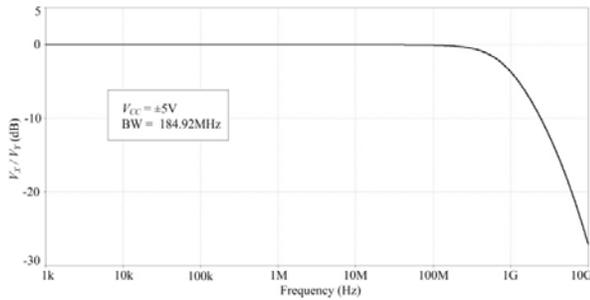


Fig. 3. Voltage transfer between the  $Y$  and  $X$  ports of CFOA.

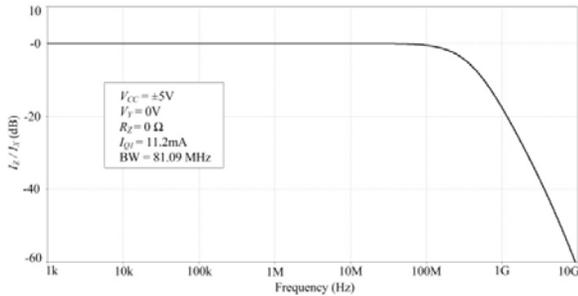


Fig. 4. Current transfers between the  $X$  and  $Z$  ports of CFOA.

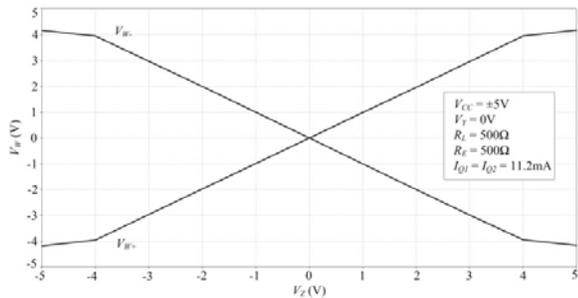


Fig. 5. DC transfer characteristic between the  $Z$  and  $W^+$  ports.

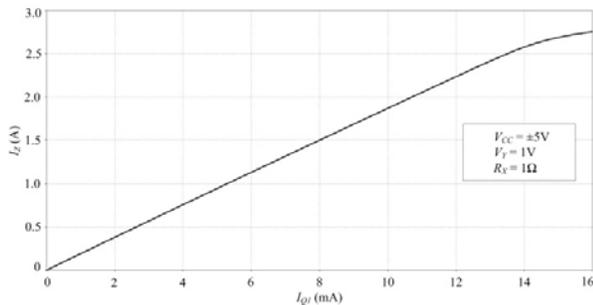


Fig. 6. DC transfer characteristic of the  $Z$  port while adjusting the transconductance gain of  $I_{Q1}$ .

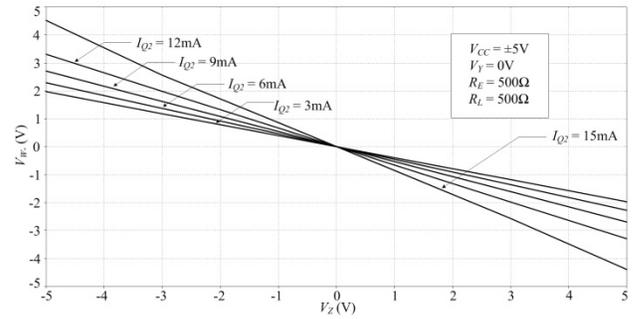


Fig. 7. DC transfer characteristic between the  $Z$  and  $W^-$  ports while adjusting the transconductance gain of  $I_{Q2}$ .

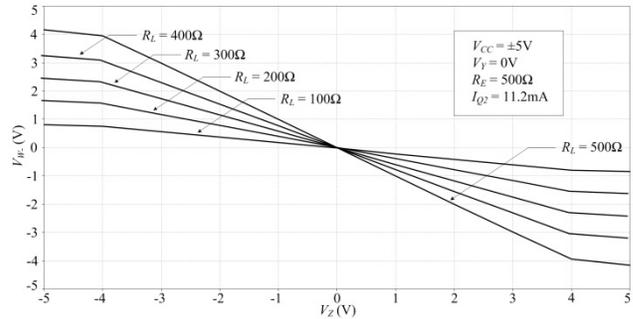


Fig. 8. DC transfer characteristic between the  $Z$  and  $W^-$  ports while adjusting the gain adjustable  $R_L$ .

Results suitable for applications in practice bases for proportional integral derivative (PID) were introduced in some recent works, replacing offers by simulation programs with transistor models of active components of some bipolar or CMOS technology. The above designs of topologies of CFOA for analogue signal processing useful wide band up the prototyping most applications of special active elements, which are not currently available on the chip.

### III. CFOA FRACTIONAL-ORDER PID

The design realisation and performance of the fractional-order  $PI^{\lambda}D^{\delta}$  controller have been presented. The fractional-order  $PI^{\lambda}D^{\delta}$  is constructed using two circuits exhibiting fractional order behaviour, called a fractance [54] of orders  $\lambda$  and  $\delta$  ( $0 < \lambda, \delta \leq 1$ ). The most popular is a domino ladder circuit network [55-56]. However, from the results of study in earlier research, resistors and capacitors are untraceable on the market [57-62]. Therefore, this research will focus the design in accordance with market aplikasi that actually works. In this section, a comparison of integer-order PID and fractional-order  $PI^{\lambda}D^{\delta}$  is made using PSPICE and by practical experimentation.

#### A. Analogue Realization: Fractance Circuits (domino ladder circuit network)

The design of fractances can be performed easily using the rational approximations, which also give a rational approximation. The relationship between the finite domino ladder network, shown in Fig. 9, and the continued fraction (4) provides an easy method for designing a circuit with a given impedance  $Z(s)$ .

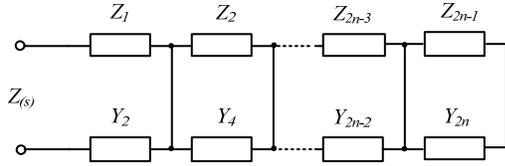


Fig. 9. Finite domino ladder network.

Let us consider the circuit depicted in Fig. 9, where  $Z_{2k-1}(s)$  and  $Y_{2k}(s)$ ,  $k = 1, \dots, n$ , are given impedances of the circuit elements. The resulting impedance  $Z(s)$  of the entire circuit can be found easily if we consider it in the right-to-left direction [31,37]:

$$Z(s) = Z_{1(s)} + \frac{1}{Y_{2(s)} + \frac{1}{\dots \frac{1}{Y_{2n-2(s)} + \frac{1}{Z_{2n-1(s)} + \frac{1}{Y_{2n(s)}}}}}} \quad (4)$$

The rational approximation of the fractional integrator/differentiator can be formally expressed as:

$$s^{\pm\alpha} \approx \left\{ \frac{P_p(s)}{Q_q(s)} \right\}_{p,q} = Z(s), \quad (5)$$

where  $p$  and  $q$  are the orders of the rational approximation and  $P$  and  $Q$  are polynomials of degree  $p$  and  $q$ , respectively.

The rational approximation of the fractional integrator/differentiator can be formally expressed as:

$$Z(s) = R_1 + \frac{1}{C_{1(s)}} \frac{1}{R_2} \frac{1}{C_{2(s)}} \frac{1}{R_3} \frac{1}{C_{3(s)}} \frac{1}{R_4} \frac{1}{C_{4(s)}} \frac{1}{R_5} \frac{1}{C_{5(s)}}, \quad (6)$$

If we consider that  $Z_{2k-1} \equiv R_{k-1}$  and  $Y_{2k} \equiv C_{k-1}$  for  $k = 1, \dots, n$  in Fig. 9, then the values of the resistors and capacitors of the network are chosen as  $R_1=51\Omega$ ,  $R_2=535\Omega$ ,  $R_3=5.6k\Omega$ ,  $R_4=13k\Omega$ ,  $R_5=820k\Omega$ ,  $C_1=470pF$ ,  $C_2=820pF$ ,  $C_3=3.9nF$  and  $C_4=820pF$ .

Therefore, the direct calculation of circuit elements was proposed. The impedance of the domino ladder circuit network (or transmission line) can be written as:

$$Z(s) = \frac{1}{(0.8 \times 10^{-9})s^{0.2}} \quad (7)$$

where  $0.8 \times 10^{-9}$  is independent of the angular frequency and  $\alpha = 0.2$ .

To demonstrate the performance of the proposed domino ladder circuit network, the measured prototype used is shown in Fig. 10.

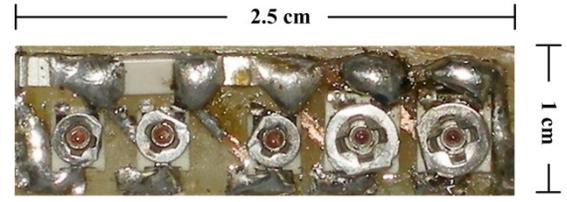


Fig. 10. The measured prototype domino ladder circuit network.

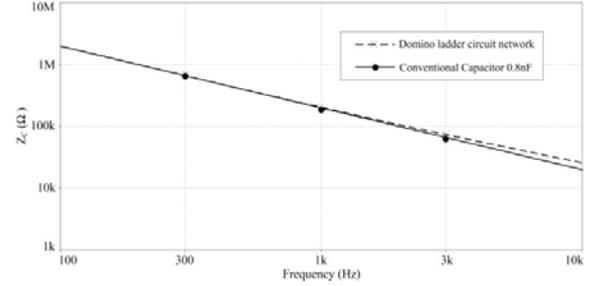


Fig. 11. Magnitude of the input impedance of the domino ladder circuit network compared to that of a conventional capacitor.

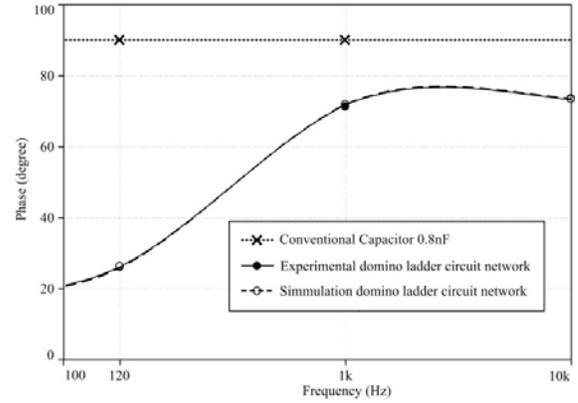


Fig. 12. Phase experiment of the frequency variations of the domino ladder circuit network compared to simulation.

From the simulation and experimental results in Fig. 11 and Fig. 12, the Magnitude and Phase of the domino ladder circuit network are compared to those of a conventional capacitor. The results confirm that the domino ladder circuit network characteristics presented work correctly from a theoretical perspective.

### B. Synthesis of proposed fractional-order $PI^{\alpha}D^{\beta}$ employing CFOA

A fractional-order  $PI^{\alpha}D^{\beta}$  controller is composed of proportional fractional-order integral and fractional-order derivative terms. The proposed fractional-order  $PI^{\alpha}D^{\beta}$  controller employs three CFOA, a grounded domino ladder circuit network ( $Z(s)$ ) and resistors as shown in Fig. 13. The transfer function of fractional-order  $PI^{\alpha}D^{\beta}$  can be written as:

$$T(s) = \frac{V_{out}(s)}{V_{in}(s)} = K_p + \frac{K_i}{s^{\alpha_1}} + s^{\alpha_2} K_d, \quad (8)$$

$$T(s) = \frac{s^{\alpha_1 + \alpha_2} K_d + s^{\alpha_1} K_p + K_i}{s^{\alpha_1}}. \quad (9)$$

Here, the proportional constant,

$$K_p = \frac{g_{m2}R_2R_8}{g_{m1}R_1R_5}, \quad (10)$$

the integral constant,

$$K_i = \frac{g_{m4}R_8}{g_{m3}R_3R_6Z_1s^{\alpha_1}}, \quad (11)$$

and the derivative constant,

$$K_d = \frac{g_{m6}R_4R_8Z_2s^{\alpha_2}}{g_{m5}R_7}, \quad (12)$$

The transfer function of the fractional-order  $PI^{\alpha}D^{\beta}$  controller is given by

$$T(s) = \frac{g_{m2}R_2R_8}{g_{m1}g_{m7}R_1R_5} + \frac{g_{m4}R_8}{g_{m3}g_{m7}R_3R_6Z_1s^{\alpha_1}} + \frac{g_{m6}R_4R_8Z_2s^{\alpha_2}}{g_{m5}g_{m7}R_7}, \quad (13)$$

where  $g_{m(\text{odd number})}$  is shown in (2) and  $g_{m(\text{even number})}$  is shown in (3), respectively.

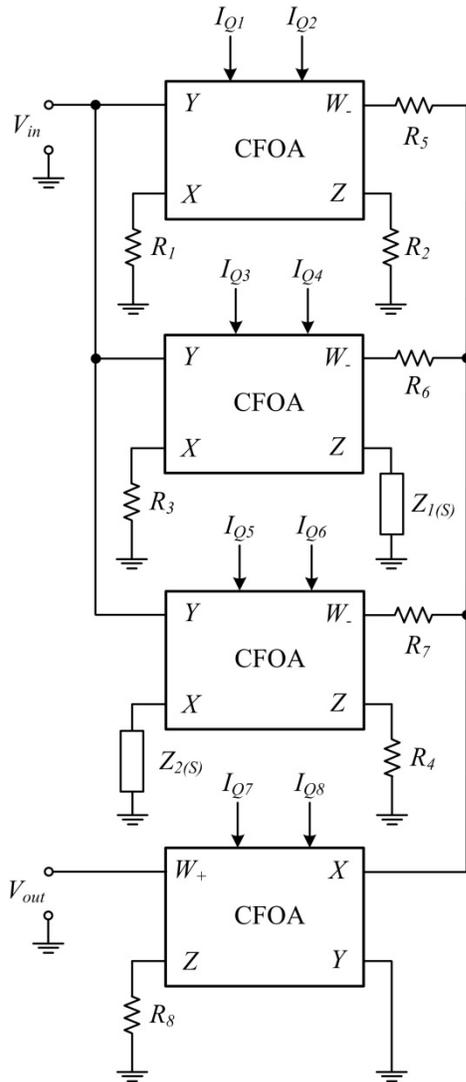


Fig. 13. Analogue fractional order  $PI^{\alpha}D^{\beta}$  controller.

From (13), the controller's parameters can be assigned to the required values by adjusting the corresponding resistor. Additionally, it can be seen that the fractional-order  $PI^{\alpha}D^{\beta}$  parameters ( $K_p$ ,  $K_i$  and  $K_d$ ) can be independently electronically controlled by the current transfers  $I_{Q1}$ - $I_{Q6}$ , respectively.

#### IV. SIMULATION AND EXPERIMENTAL RESULTS

In this section, a comparison of integer-order PID and fractional-order  $PI^{\alpha}D^{\beta}$  is made using the PSPICE simulation program and by practical experimentation. To prove the performances of the proposed controllers, the Diamond transistors (DT) employed in CFOA of the proposed circuit were simulated and experimented with by using the commercially available OPA 860. Fig. 2 depicts a schematic description of the internal construction of the CFOA.

Fig. 13 depicts an analogue implementation of a fractional-order  $PI^{\alpha}D^{\beta}$  controller. A fractional-order integrator is approximated by the domino ladder circuit network impedance  $Z_1(s)$ , and fractional-order differentiator is approximated by the impedance  $Z_2(s)$ , where orders of both approximations are  $\alpha = 0.2$ . In this case, if we use identical resistors (R-series) and identical capacitors (C-shunt) in the domino ladder circuit network, then the behaviour of the circuit will be that of a fractional-order integrator/differentiator. Realization and measurements of such types of controllers were done in this paper.

The transfer function of the fractional-order  $PI^{\alpha}D^{\beta}$  controller can be evaluated and rewritten as a fractional order:

$$T(s) = \frac{V_{out}(s)}{V_{in}(s)} = K_p + \frac{K_i}{s^{0.2}} + s^{0.2}K_d, \quad (14)$$

$$T(s) = \frac{s^{0.4}K_d + s^{0.2}K_p + K_i}{s^{0.2}}. \quad (15)$$

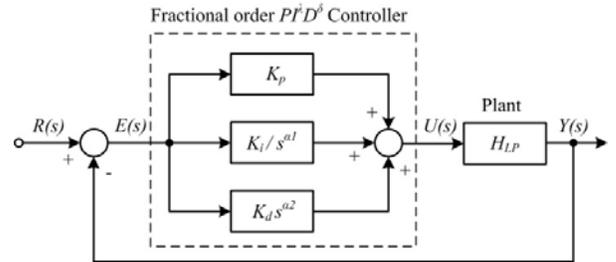


Fig. 14. Fractional-order  $PI^{\alpha}D^{\beta}$ -controlled System.

To validate the practical application of the proposed controller, a passive low-pass filter was used to realize a closed-loop control system, as depicted in Fig. 14. For the low-pass filter, the circuit shown in Fig. 15 with an addition output terminal is chosen. The transfer function of the 2<sup>nd</sup>-order Sallen-Key Low-pass filter can be written as:

$$H_{LP} = \frac{V_{out}}{V_{in}} = \frac{1}{s^2 + s\left(\frac{1}{R_2C_1} + \frac{1}{R_1C_1}\right) + \frac{1}{R_1R_2C_1C_2}}, \quad (16)$$

This Sallen-Key low-pass filter has the following transfer function with the values of  $R_1=30\text{ k}\Omega$ ,  $R_2=18\text{ k}\Omega$ ,  $C_1=10\text{ nF}$  and  $C_2=4.7\text{ nF}$ . Therefore, direct calculation of circuit elements was proposed. The Sallen-Key low-pass filter can be written as:

$$H_{LP} = \frac{V_{out}}{V_{in}} = \frac{39401103.23}{s^2 + s8888.89 + 39401103.23} \quad (17)$$

From (15), the  $R$  and  $C$  values for the Sallen-Key low-pass filter can be calculate at a given cut-off frequency ( $f_c$ ), quality factor ( $Q$ ) and Damping ratio ( $\zeta$ ) as follows:  $f_c = 1\text{ kHz}$ ,  $Q = 0.7$ , and  $\zeta = 0.7$ .

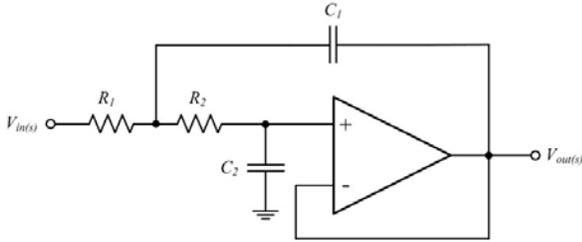


Fig. 15. Sallen-Key Active Filter.

#### A. Comparison in Simulation

To demonstrate the performance of the proposed fractional-order  $PI^{\delta}D^{\delta}$  controller, the PSPICE simulation program was used for the examinations. The CFOAs of the proposed fractional-order  $PI^{\delta}D^{\delta}$  controller were simulated by using the domino ladder circuit network from (7) to achieve the behaviour of the circuit as a fractional-order integrator/differentiator. For the proposed fractional-order  $PI^{\delta}D^{\delta}$  controller, PSPICE simulations are performed with  $R_1=R_5=R_6=R_7=R_8=100\ \Omega$ ,  $R_2=200\ \Omega$ ,  $R_3=200\text{ k}\Omega$ ,  $R_4=100\text{ k}\Omega$ ,  $Z_{1(s)}=Z_{2(s)}=1/[(0.8 \times 10^{-9})s^{0.2}]$ , and  $I_{Q1}=I_{Q2}=I_{Q3}=I_{Q4}=I_{Q5}=I_{Q6}=I_{Q7}=I_{Q8}=11.2\text{ mA}$  for the circuits depicted in Fig. 13. Therefore, the fractional-order  $PI^{\delta}D^{\delta}$  controller has the parameters that are calculated as  $K_p=1.06 \times 10^3$ ,  $K_i=662.50 \times 10^6$ , and  $K_d=42.24 \times 10^{-3}$ . These proposed circuits were biased with the symmetrical  $\pm 5\text{ V}$  supply voltages.

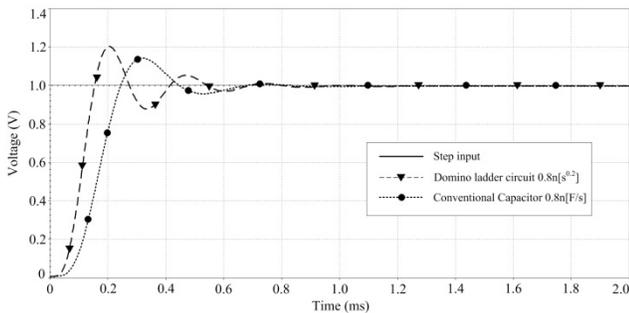


Fig. 16. Step response for PID and open-loop systems.

In Fig. 16, the fractional-order  $PI^{\delta}D^{\delta}$  controller is compared with integer-order PID. From Fig. 16, we can see that the overshoot of the unit step response with an open loop using the designed fractional-order  $PI^{\delta}D^{\delta}$  controller is much shorter than that using the designed integer-order PID controller.

A closed-loop control system depicted in Fig. 14 can be constructed with the fractional-order  $PI^{\delta}D^{\delta}$  controller in Fig. 13 and the Sallen-Key low-pass filter with the transfer function given in (16). The simulated response for this fractional-order  $PI^{\delta}D^{\delta}$  controller with unit step is given in Fig. 17. It is observed that the proposed fractional-order  $PI^{\delta}D^{\delta}$  controller system enters steady-state and follows the unit step input with a steady-state error compared with integer-order PID.

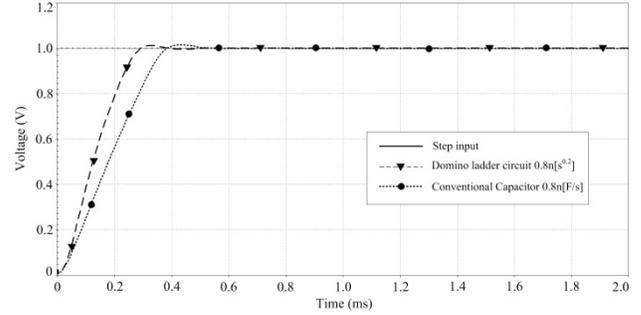


Fig. 17. Closed-loop response of the system.

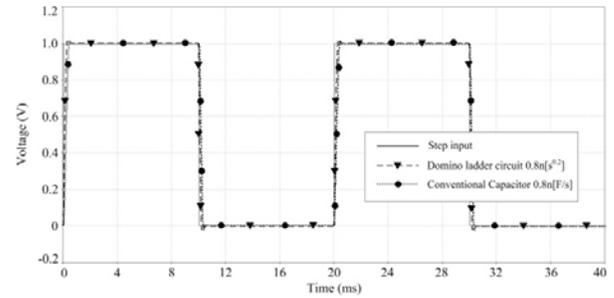


Fig. 18. Real-time closed-loop response of system.

For controlled system performance comparison, we have summarized some performance characteristics in Table 1 for the controlled system with both controllers. In any controlled system, the final step in the design process is the real-time controlled experiment. As can be observed from Fig. 18, the performances of the controllers are confirmation of the simulation results based on the identified model.

TABLE I  
SIMULATION RESULTS OF THE PSPICE-BASED CONTROL SALLEN-KEY ACTIVE LOW-PASS FILTER

Parameters	Fractional order $PI^{\delta}D^{\delta}$ controlled	Integer order PID controlled
Max. overshoot	2.24%	2.36%
Rise time	234.37 $\mu\text{s}$	324.72 $\mu\text{s}$
Settling time	269.06 $\mu\text{s}$	445.43 $\mu\text{s}$
Steady-state error	553.66 $\mu\text{s}$	654.49 $\mu\text{s}$

From Fig. 16 to Fig. 18, we can see that the overshoot, rise time, settling time and steady-state error of the unit step response using the designed fractional-order  $PI^{\delta}D^{\delta}$  controller is much shorter than that using the designed integer-order PID controller. Thus, following our proposed design algorithms, the fractional-order  $PI^{\delta}D^{\delta}$  controller outperforms the integer-

order PID for the fractional-order systems considered.

*B. Experimental Verification in real-time*

A closed-loop control system prototype is shown in Fig. 19. The passive and active elements are the same as appeared in the simulation program. The responses of fractional-order  $PI^{\delta}D^{\delta}$  and integer-order PID controllers are compared via the experimental setup of the control system. Fig. 20 and Fig. 21 show responses of fractional-order  $PI^{\delta}D^{\delta}$  and integer-order PID controllers per unit step. Fig. 22 and Fig. 23 show the real-time closed-loop response of the system for both the controllers.

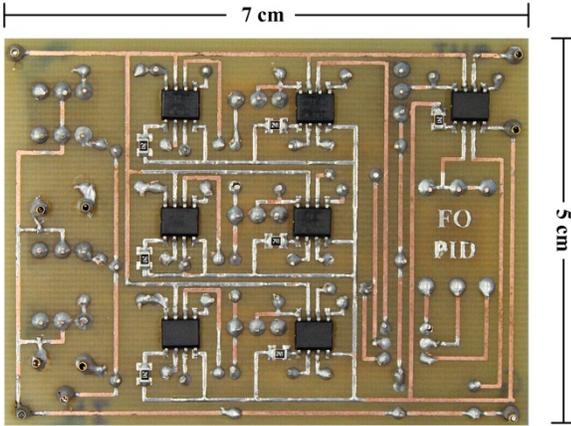


Fig. 19. The measured fractional order  $PI^{\delta}D^{\delta}$  prototype.

TABLE II  
EXPERIMENTAL RESULTS OF THE CIRCUIT PROTOTYPE-BASED CONTROL  
SALLEN-KEY ACTIVE LOW-PASS FILTER

Parameters	Fractional order $PI^{\delta}D^{\delta}$ controlled	Integer order PID controlled
Max. overshoot	8.11%	12.17%
Rise time	172.50 $\mu$ s	202.90 $\mu$ s
Settling time	490 $\mu$ s	950 $\mu$ s
Steady-state error	750 $\mu$ s	1.25 ms

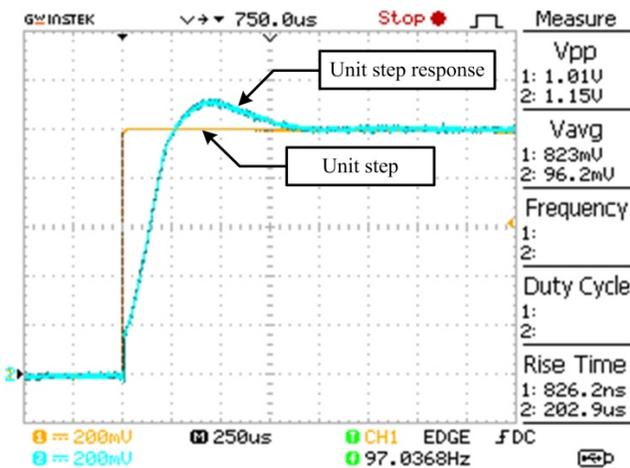


Fig. 20. Real-time closed-loop response of the Integer-order system.

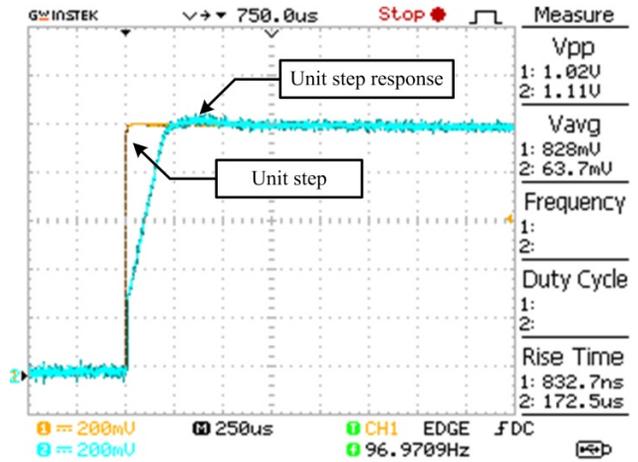


Fig. 21. Real-time closed-loop response of the Fractional-order system.

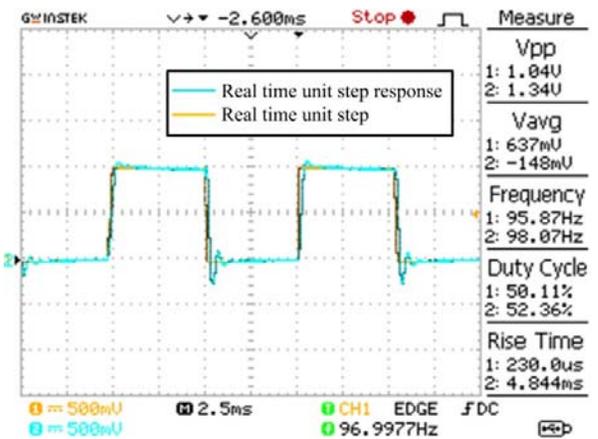


Fig. 22. Real-time closed-loop response of the Integer-order system.

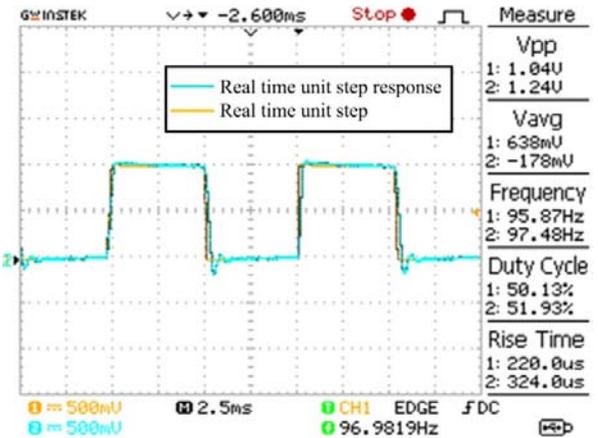


Fig. 23. Real-time closed-loop response of the Fractional-order system.

For control system performance enhancement comparison, we have summarized some performance characteristics in Table 2 for the feedback control system with both controllers. As seen, the experimental results confirm the theoretical results very well.

## V. CONCLUSION

The modified version of the building block, the so-called current feedback operational amplifier (CFOA), has been introduced in this paper. This modification can be achieved by using Diamond Transistor (DT) instead of going through complicated IC fabrication process. Modification of the CFOA has been applied as a fractional-order proportional integral derivative ( $PI^{\alpha}D^{\beta}$ ) controller. In this CFOA application, two fractional-order proportional integral controllers have been designed to improve the performance of fractional-order systems, which can model many real systems in control engineering. Comparisons have been made between the fractional-order proportional integral controllers and the traditional integer-order PID controller. From the simulation and experimental results, we can see that the overshoot, rise time, settling time and steady-state error of the unit step response using the designed fractional-order  $PI^{\alpha}D^{\beta}$  controller is much shorter than that using the designed integer-order PID controller.

## REFERENCES

- [1] N. Herencsar, K. Vrba, J. Koton, I. Lattenberg, "The conception of differential-input buffered and transconductance amplifier (DBTA) and its application", *IEICE Electronics Express*, vol. 6, no. 6, pp. 329-334, 2009.
- [2] A. U. Keskin, D. Biolk, E. Hancioglu, V. Biolkova, "Current-mode KHN filter employing current differencing transconductance amplifiers", *Int'l J. Electronics and Communications*, vol. 60, no. 6, pp. 443-446, 2006.
- [3] N. Herencsar, J. Koton, K. Vrba, J. Misurec, "A novel current mode SIMO type universal filter using CFTAs", *Contemporary Engineering Sciences*, vol. 2, no. 2, pp. 59-66, 2009.
- [4] Prokop, R., Musil, V., "New modular current devices for true current mode signal processing", *Electronics*, vol. 16, no. 4, pp. 36-42, 2007.
- [5] R. Prokop, V. Musil, "CCTA: a new modern circuit block and its internal realization", In *Electronic Devices and Systems IMAPS CZ International Conf. Brno (Czech Republic)*, pp. 89-93, 2005.
- [6] R. Prokop, V. Musil, "New modern circuit block CCTA and some its applications", In *The 14th Int. Scientific and Applied Science Conf. Electronics ET'2005, Book 5. Sofia (Bulgaria)*, pp. 93-98, 2005.
- [7] R. Prokop, V. Musil, "Modular approach to design of modern circuit blocks for current signal processing and new device CCTA", In *Proceedings of the 7th IASTED International Conference on Signal and Image Processing. Anaheim (USA)*, pp. 494-499, 2005.
- [8] R. Prokop, "Modular approach to design of modern analog devices in CMOS technology", *Doctoral (Ph.D.) Thesis*. Dept. of Microelectronics FEEC BUT Brno, Czech Republic, pp. 1-98, 2009.
- [9] Chiu W., Liu S.-I., Tsao H.-W., Chen J.-J., "CMOS differential difference current conveyors and their applications", *IEE Proceedings Circuits, Devices and Systems*, vol. 143, no. 2, pp. 91-96, 1996.
- [10] F. Khateb, D. Kubanek, G. Tsirimokou, and C. Psychalinos, "Fractional-order filters based on low-voltage DDCCs", *Microelectronics Journal*, vol. 50, pp. 50-59, 2016.
- [11] D. Biolk, R. Senani, V. Biolkova, Z. Kolka, "Active elements for analog signal processing: Classification, Review, and New Proposal", *Radioengineering*, vol. 17, no. 4, pp. 15-32, 2008.
- [12] Comlinear Corp., *Designer's guide for 200 series op amp*, Application note 200-1, 4800, wheaton Drive, Ft. Collins, CO 80525, Nov. 1984.
- [13] Analog Devices. *Linear products data book*. Norwood MA, 1990.
- [14] A. Fabre, "Insensitive voltage mode and current mode filters from commercially available transimpedance op-amps", In *IEE Proceedings - Circuits Devices and Systems*, vol. 142, pp. 140-144, 1995.
- [15] M. T. Abuelma'Atti, S. M. Al-Shahrani, "New universal filter using two current-feedback amplifiers", *International Journal of Electronics*, vol. 80, no. 6, pp. 753-756, 1996.
- [16] A. M. Soliman, "Generation of CCII and CFOA filters from passive RLC filters", *International journal of electronics*, vol. 85, no. 3, pp. 293-312, 1998.
- [17] S. Minaei, O. K. Sayin, H. Kuntman, "A new CMOS electronically tunable current conveyor and its application to current-mode filters", *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 53, no. 7, pp. 1448-1457, 2006.
- [18] E. Yuce, S. Minari, "A modified CFOA and its applications to simulated inductors, capacitance multipliers, and analog filters", *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 55, no. 1, pp. 254-263, 2008.
- [19] Texas Instruments. Wide bandwidth operational transconductance amplifier and buffer OPA 860. 2005, 29 p. [Online] Available at: [www.ti.com](http://www.ti.com)
- [20] C. Erdal, A. Toker, C. Acar, "A new proportional-integral-derivative (PID) controller realization by using current conveyors and calculating optimum parameter tolerances", *Journal of Electrical and Electronics*, vol. 1, no. 2, pp. 267-273, 2001.
- [21] V. V. Kulish, J. Luis Lage, "Application of Fractional Calculus to Fluid Mechanics", *Journal of fluids engineering*, vol. 124, pp. 803-806, 2002.
- [22] N. Engheia, "On the role of fractional calculus in electromagnetic theory", *IEEE Antennas and Propagation Magazine*, vol. 39, no. 4, pp. 35-46, 1997.
- [23] R.L. Bagley, R.A. Calico, "Fractional-order state equations for the control of viscoelastic damped structures", *Journal of Guidance, Control and Dynamics*, vol. 14, no. 2, pp. 304-311, 1991.
- [24] P.J. Torvik, R.L. Bagley, "On the appearance of the fractional derivative in the behavior of real materials", *Journal of Applied Mechanics-transactions of The Asme - J APPL MECH*, vol. 51, no. 2, pp. 294-298, 1984.
- [25] A. Makroglou, R.K. Miller, S. Skaar, "Computational results for a feedback control for a rotating viscoelastic beam", *Journal of Guidance, Control, and Dynamics*, vol. 17, no. 1, pp. 84-90, 1994.
- [26] K.B. Oldham, J. Spanier, "The fractional calculus. Theory and Application of differentiation and integration to arbitrary order", *New York (US): Academic Press*, 1974. ISBN 0125255500
- [27] B. Krisna, "Studies on fractional order differentiators and integrators: A survey", *Signal Processing*, vol. 91, no. 3, pp. 386-426, 2011.
- [28] G. Tsirimokou, C. Psychalinos, "Ultra-low voltage fractional-order differentiator and integrator topologies: an application for handy noisy ECGs", *Analog Integrated Circuits and Signal Processing*, vol. 81, no. 2, pp. 393-405, 2014.
- [29] I. Podlubny, "Fractional-order systems and  $PI^{\alpha}D^{\beta}$  controllers", *Automatic Control, IEEE Transactions on*, vol. 44, no. 1, pp. 208 - 213, 1999.
- [30] C. Jun-Yi, C. Bing-Gang, "Design of Fractional Order Controllers Based on Particle Swarm Optimization", *International Journal of Control, Automation, and Systems*, vol. 4, no. 6, pp. 775-781, 2006.
- [31] C. Yang-Quan, I. Petras, X. Dingyu, "Fractional order control - A tutorial", *American Control Conference, ACC '09*, pp. 1397-1411, 2009.

- [32] A. G. Radwan, A. M. Soliman, A. S. Ewakil, "First-order filters generalized to the fractional Domain", *Journal of Circuits, Systems and Computers*, vol. 17, no. 1, pp. 55-66, 2008.
- [33] A. G. Radwan, A. S. Elwakil, A. M. Soliman, "On the generalization of second-order filters to the fractional-order domain", *Journal of Circuits, Systems and Computers*, vol. 18, no. 2, pp. 361-386, 2009.
- [34] T. Freeborn, B. Maundy, A. S. Elwakil, "Field programmable analogue array implementation of fractional step filters", *IET Circuits, Devices and Systems*, vol. 4, no. 6, pp. 514-524, 2010.
- [35] B. Maundy, A. S. Elwakil, T. Freeborn, "On the practical realization of higher-order filters with fractional stepping", *Signal Processing*, vol. 91, no. 3, pp. 484-491, 2011.
- [36] P. Ahmadi, B. Maundy, A. S. Elwakil, "High quality factor asymmetric slope band-pass filters: fractional-order capacitor approach", *IET Circuits, Devices & Systems*, vol. 6, no. 3, pp. 1-11, 2011.
- [37] A. Ali, A. G. Radwan, A. M. Soliman, "Fractional order Butterworth filter: Active and passive realizations", *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no.3, pp. 346-354, 2013.
- [38] G. Tsirimokou, C. Laoudias, C. Psychalinos, "0.5-V fractional-order companding filters", *International Journal of Circuit Theory and Applications*, vol. 43, no. 9, pp. 1105-1126, 2015.
- [39] G. Tsirimokou, C. Psychalinos, A. S. Elwakil, "Digitally programmed fractional-order Chebyshev filters realizations using current-mirrors", In *Proceedings of IEEE Int. symposium on Circuits and Systems (ISCAS)*, Lisbon, Portugal, pp. 2337-2340, 2015.
- [40] M. Tripathy, D. Mondal, K. Biswas, and S. Sen, "Experimental studies on realization of fractional inductors and fractional-order bandpass filters", *International Journal of Circuit Theory and Applications*, vol. 43, pp. 1183-1196, 2015.
- [41] D. Mondal, K. Biswas, "Performance study of fractional order integrator using single-component fractional order element", *IET Circuits, Devices and Systems*, vol. 5, no. 4, 334-342, 2011.
- [42] A. G. Radwan, K. Salama, "Fractional-order RC and RL circuits", *Circuits Systems and Signal Processing*, vol. 31, no. 6, pp. 1901-1915, 2012.
- [43] R. Sotner, J. Jerabek, N. Herencsar, J. Petrzela, T. Dostal, K. Vrba, "First-order adjustable transfer sections for synthesis suitable for special purposes in constant phase block approximation", *AEU – International Journal of Electronics and Communications*, vol. 69, no. 9, pp. 1334-1345, 2015.
- [44] G. Tsirimokou and C. Psychalinos, "Ultra-Low Voltage Fractional-Order Circuits Using Current-Mirrors", *International Journal of Circuit Theory and Applications*, vol. 44, no.1 pp.109-126, 2016.
- [45] A. Adhikary, M. Khanra, S. Sen, K. Biswas, "Realization of carbon nanotube based electrochemical fractor", In *Proceedings of IEEE Int. symposium on Circuits and Systems (ISCAS)*, Lisbon, Portugal, pp. 2329-2332, 2015.
- [46] A. M. Elshurafa, M. N. Almadhoun, K. N. Salama, and H. N. Alshareef, "Microscale electrostatic fractional capacitors using reduced graphene oxide percolated polymer composites", *Applied Physics Letters*, vol. 102, no. 102, pp. 232901, 2013.
- [47] I. Dimeas, G. Tsirimokou, C. Psychalinos, A. Elwakil, "Realization of Fractional-Order Capacitor and Inductor Emulators Using Current Feedback Operational Amplifiers", In *Proceedings of Int. Symposium on Nonlinear Theory and its Applications (NOLTA2015)*, Hong Kong, China, pp. 237-240, 2015.
- [48] G. Tsirimokou, C. Psychalinos, and A. Elwakil, "Emulation of a Constant Phase Element Using Operational Transconductance Amplifiers", *Analog Integrated Circuits and Signal Processing*, vol. 85, no. 3, pp 413-423, 2015.
- [49] J. Petrzela, "Arbitrary phase shifters with decreasing phase", In *Proceedings of 37th International Conference on Telecommunications and Signal Processing (TSP 2014)*, Berlin, Germany, pp. 329-333, 2014.
- [50] A.M. Concepción, M. Licia Monje, "Design Methods of Fractional Order Controllers for Industrial Applications", *Doctoral (Ph.D.) Thesis*, University of Extremadura, Spain, 2006.
- [51] A.M. Concepción, M.V. Blas, F. Vicente, C. Yangquan, "Tuning and Auto-Tuning of Fractional Order Controllers for Industry Applications", *Control Engineering Practice*, vol. 16, pp. 798-812, 2008.
- [52] C. Yang-Quan, "Ubiquitous fractional order controls", In *Proceeding of the 2nd IFAC Workshop on Fractional Differentiation and its Applications*. Porto (Portugal), pp.19 -21, 2006.
- [53] J.A. Svoboda, L. Megory, S. Webb, "Applications of a commercially available current conveyor", *International Journal of Electronics*, vol. 70, pp. 159-164, 1991.
- [54] I. Podlubny, "Fractional Differential Equations. An Introduction to Fractional Derivatives, Fractional Differential Equations, Some Methods of Their Solution and Some of Their Applications", San Diego (USA): Academic Press, 1999. ISBN: 0125588402
- [55] G. Carlson, C. Halijak, "Approximation of fractional capacitors (1/s) 1/n by a regular Newton process", *Circuit Theory, IEEE Transactions on*, vol. 11, no. 2, pp. 210-213, 1964.
- [56] S. Westerlund, L. Ekstam, "Capacitor theory", *Dielectrics and Electrical Insulation, IEEE Transactions on*, vol. 1, no. 5, pp. 826- 839, 1994.
- [57] M. Moshrefi-Torbati, J.K. Hammond, "Physical and geometrical interpretation of fractional operator", *Journal of the Franklin Institute*, vol. 335, no. 6, pp. 1077-1086, 1998.
- [58] C.X. Liu, L. Liu, "Circuit implementation of a new hyper chaos in fractional-order system", *Chinese Physics B*, vol. 17, no. 8, pp. 2829 – 2836, 2008.
- [59] J. Valsa, P. Dvorak, M. Friedl, "Network Model of the CPE", *Radioengineering*, vol. 20, no. 3, pp. 619 – 626, 2011.
- [60] J. Slezak, T. Gotthans, J. Drinovsky, "Evolutionary Synthesis of Fractional Capacitor Using Simulated Annealing Method", *Radiengineering*, vol. 21, no. 4, pp. 1252 -1259, 2012.
- [61] A. Oustaloup, F. Levron, B. Mathieu, F. M. Nanot, "Frequency-band complex non integer differentiator: characterization and synthesis", *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 47 no. 1, pp. 25-39, 2000.
- [62] J.C. Wang, "Realizations of generalized warburg impedance with RC ladder networks and transmission lines", *Journal of the Electrochemical Society*, vol. 134, no. 8, pp. 1915-1920, 1987.

# Statistical Analysis of Multiple Access Interference in Chaotic Spreading Sequence Based DS-CDMA Systems

A. Litvinenko, E. Bekeris

**Abstract**— This paper presents a statistical analysis of multiple access interference (MAI) in Direct Sequence Code Division Multiple Access (DS-CDMA) communication systems based on different types of chaotic spreading sequences. The probability distribution of the interference in a system with  $K$  users causing the MAI is studied using MATLAB simulation. For chaotic spreading sequence generation six different 1-D chaotic maps are used: modified Bernoulli, modified Tent, Gauss, Sine-Circle, Cubic and Pinchers map. A brief statistical analysis of the cross-correlation properties of the chaotic sequences generated by the aforementioned maps is also presented.

**Index Terms**—1-D chaotic maps, chaotic spreading sequences, Direct Sequence Code Division Multiple Access (DS-CDMA) communication system, multiple-access interference (MAI).

*Original Research Paper*  
DOI: 10.7251/ELS1721034L

## I. INTRODUCTION

THE possibilities of exploiting chaos phenomena for the communication area have been intensively studied for the last 30 years. One promising research direction is the use of chaotic spreading sequence for Direct Sequence Code Division Multiple Access (DS-CDMA) systems [1]-[3] and one of approaches for the performance evaluation of these DS-CDMA multi-user systems is to apply the Gaussian approximation of the multiple-access interference (MAI) [2]-[4].

This research is an extension of the work devoted to the study of MAI distribution in DS-CDMA systems based on relatively short not-return-to-zero (NRZ) chaotic spreading sequences generated by a logistic map [5]. Publication [5] shows that MAI cannot be considered as Gaussian random

variable if the chaotic spreading sequence generated by Logistic map is less than 60 chips. In turn, the proposed research presents the analysis of MAI distribution in DS-CDMA systems based on chaotic spreading sequences generated by other 1-D maps: modified Bernoulli map, modified Tent map, Gauss map, Sine-Circle map, Cubic map, and Pinchers map.

According to [6], it is possible to generate a higher number of relatively short chaotic sequences with a low cross-correlation level compared with classical pseudo-noise (PN) sequences, which are widely used in DS-CDMA systems. This paper also analyzes periodic cross-correlation properties of chaotic spreading sequences generated by 1-D maps and classical pseudo-noise sequences.

Moreover, chaotic spreading sequences could be used to increase the resistance to fading in wireless sensor networks [7], to increase the security level of these networks [8], and for synchronization challenge [9]. Therefore, it is important to choose the correct method for the performance evaluation of chaotic spreading sequence based DS-CDMA multi-user system and this research verifies the correctness of Gaussian approximation for MAI simulation in the proposed systems.

The paper is organized in the following manner: the next section presents algorithms for generation of binary chaotic spreading sequences, whose periodic cross-correlation properties are analyzed in the third section; the fourth section describes the block diagram of the simulated DS-CDMA system and the results of simulation are shown in the fifth section; finally, an overview of the obtained results is presented in conclusions.

## II. GENERATION OF CHAOTIC SPREADING SEQUENCES

In this section algorithms of binary chaotic sequence generation are presented. The first step for obtaining the binary chaotic spreading sequence  $c(n)$  is generation of a non-binary chaotic sequence  $x(n)$ , and for this purpose six different 1-D chaotic maps have been chosen: Bernoulli (1), Tent (2), Gauss (3), Sine-Circle (4), Cubic (5), and Pinchers map (6). The parameters of these 1-D maps have been selected with the goal of obtaining a chaotic behavior for the sequences.

Manuscript received 3 May 2016. Received in revised form 31 October 2016. Accepted for publication December 20, 2016.

A. Litvinenko is with the Department of Fundamentals of Electronics, Riga Technical University, Azenes St. 12, Riga, Latvia (e-mail: anna.litvinenko@rtu.lv).

E. Bekeris is with the Department of Fundamentals of Electronics, Riga Technical University, Azenes St. 12, Riga, Latvia (e-mail: elmars.bekeris@rtu.lv).

TABLE I  
SPECIFIED THRESHOLD  $Th$  VALUE FOR CHAOTIC MAPS

Map type	Bernoulli	Tent	Gauss	Sine-Circle	Cubic	Pinchers
$Th$ value	0.5	0.5	0.01	3.25	0	0.38

Bernoulli map ( $p = 1.999$  and  $x \in (0, 1)$ ):

$$x(n+1) = \text{mod}(px(n), 1). \quad (1)$$

Tent map ( $\mu = 1.999$  and  $x \in (0, 1)$ ):

$$x(n+1) = \begin{cases} \mu(1-x(n)), & x(n) \geq 0.5 \\ \mu x(n), & x(n) < 0.5 \end{cases}. \quad (2)$$

Gauss map ( $\alpha = 9.75$ ,  $\beta = -0.53$ , and  $x \in (-0.5, 0.5)$ ):

$$x(n+1) = e^{-\alpha} x(n)^2 + \beta. \quad (3)$$

Sine-Circle map ( $H = 12$ ,  $\Omega = 0.2$ , and  $x \in (0, 6.5)$ ):

$$x(n+1) = \text{mod}\left(x(n) + \Omega - \frac{H}{2\pi} \sin(2\pi x(n)), 2\pi\right). \quad (4)$$

Cubic map ( $B = 3$  and  $x \in (-1.1547, 1.1547)$ ):

$$x(n+1) = Bx(n)(1-x(n)^2). \quad (5)$$

Pinchers map ( $s = 2$ ,  $C = 0.5$ , and  $x \in (0, 0.76)$ ):

$$x(n+1) = |\tanh s(x(n) - C)|. \quad (6)$$

The generated non-binary chaotic sequence  $x(n)$  is converted into the binary NRZ chaotic sequence  $c(n)$  using the comparison rule (7), where a defined threshold  $Th$  is equal to the mean value of  $x$  definition interval. The value of the threshold  $Th$  for each observed map is presented in Table I.

$$c(n) = \begin{cases} 1, & x(n) \geq Th \\ -1, & x(n) < Th \end{cases} \quad (7)$$

According to [4], by dividing the state space of 1-D maps into two parts and using different initial conditions, completely different binary sequences are generated.

### III. CROSS-CORRELATION PROPERTIES OF BINARY CHAOTIC SEQUENCES

In this section the distribution of maximum absolute values of periodic cross-correlation among binary chaotic sequences is studied.

To analyze cross-correlation properties, 100 binary chaotic sequences with the same length for four different length cases ( $N = 15, 31, 63, 127$ ) and for the six previously discussed maps (1) – (6) have been generated. The distribution of the maximum absolute values of the periodic cross-correlation  $|R_{ik}|$  is analysed for all possible pair combinations from 100 sequences of the same map and same length. Table II presents statistical parameters – minimal, maximal, mean values and standard deviation of the maximum absolute values of the periodic cross-correlation for chaotic sequences. These statistical parameters of chaotic sequences are compared with the corresponding maximum values of the periodic cross-correlation for pseudo-noise sequences. Obviously, it is possible to generate binary chaotic sequences with the same or better cross-correlation properties than in the case of classical pseudo-noise sequences for code lengths of 15–63 chips. In addition, the number of chaotic sequences with those lengths

could be increased, what is impossible in the case with

TABLE II  
STATISTICAL PARAMETERS OF CROSS-CORRELATION

N	15	31	63	127
Bernoulli map				
Min	0.2	0.23	0.21	0.16
Mean	0.52	0.41	0.32	0.25
Max	1	0.74	0.59	0.45
Standard deviation	0.12	0.08	0.05	0.03
Tent map				
Min	0.2	0.23	0.17	0.16
Mean	0.53	0.41	0.32	0.25
Max	1	0.87	0.62	0.39
Standard deviation	0.12	0.07	0.05	0.03
Gauss map				
Min	0.2	0.16	0.17	0.16
Mean	0.54	0.43	0.34	0.25
Max	1	0.87	0.59	0.45
Standard deviation	0.12	0.08	0.06	0.04
Sine-Circle map				
Min	0.07	0.23	0.2	0.16
Mean	0.56	0.45	0.36	0.28
Max	1	0.8	0.62	0.48
Standard deviation	0.14	0.09	0.06	0.04
Cubic map				
Min	0.07	0.16	0.17	0.16
Mean	0.54	0.44	0.35	0.27
Max	1	0.87	0.62	0.54
Standard deviation	0.15	0.09	0.06	0.04
Pinchers map				
Min	0.07	0.16	0.17	0.19
Mean	0.67	0.55	0.45	0.35
Max	1	0.94	0.8	0.46
Standard deviation	0.17	0.13	0.09	0.05
Maximum values of $ R_{ik} $ for PN sequences				
m-seq.	0.6	0.35	0.36	0.32
Gold	0.6	0.29	0.27	0.13
Kasami	0.33	0.215	0.14	0.09

classical pseudo-noise sequences.

### IV. MODEL OF CHAOTIC SPREADING SEQUENCE BASED DS-CDMA SYSTEM

In this section simulation parameters for a model of DS-CDMA communication system based on chaotic spreading sequences are discussed.

For the evaluation of the probability distribution of the MAI in a DS-CDMA communication system based on chaotic spreading sequences, a model with  $K$  transmitters and one receiver is used and a block diagram of this mode is represented in Fig. 1. The information bit NRZ sequence  $b(t)$  is sent at the input of each transmitter. Then information bits are spread by chaotic NRZ sequence  $c(t)$ , which is unique for

each transmitter. After this, the signal is transferred on carrier

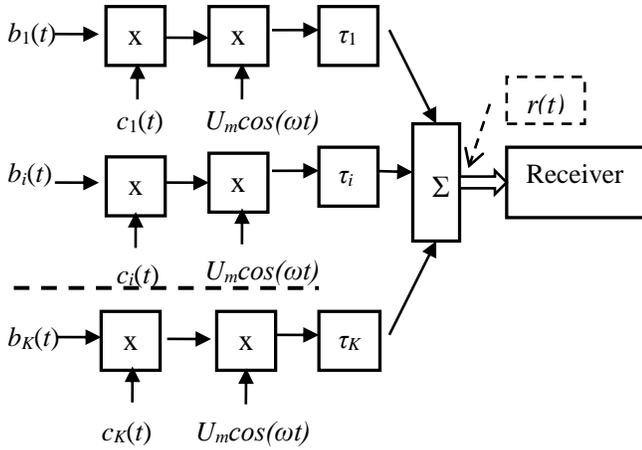


Fig. 1. Block diagram of DS-CDMA system [5].

frequency  $U_m \cos(\omega_0 t)$ . Each transmitter signal has a corresponding time delay  $\tau$  observed by the receiver. Since the subject of this research is the probability distribution of the multiple-access interference, the channel noise during simulation is neglected.

The correlation receiver is matched to the  $(K+1)$ -th transmitter spreading sequence  $c_{K+1}$ . The input signal of the receiver  $r(t)$  causes the multiple-access interference:

$$r(t) = \sum_{i=1}^K b_i(t) \cdot c_i(t) \cdot U_m \cos(\omega_0 t + \omega \tau_i). \quad (8)$$

The following parameters have been selected for simulation:

- Information bits are the random binary NRZ sequence  $b_i(t)$  of duration  $B_j T_b$ , where  $T_b$  is the duration of one bit and  $B_j$  is the number of transmitted bit, which is randomly selected from the set (0,300) with an equal probability.
- Spreading sequence is binary NRZ chaotic sequence  $c_i(t)$  generated on the basis of one of the 1-D maps (1) – (6) and comparison rule (7). The length of spreading sequence can have one of four different values:  $N = 15, 31, 63, 127$ .
- Initial values for chaotic spreading sequence generation are randomly selected from  $x$  definition interval with equal probability and correspondingly to the map type.
- Number of transmitters causing the multiple-access interference for the correlation receiver:  $K = 6, 18, 33, 65, 129, 259$ .

According to [5], at the time moments of  $jT_b$  the following values will be obtained at the output of the correlation receiver:

$$U_j = A \int_0^{jT_b} \left[ \sum_{i=1}^K b_i(t) \cdot c_i(t) \cdot \cos(\omega \tau_i) \right] c_{K+1}(t) dt, \quad (9)$$

where  $A$  is a coefficient which depends on the parameters of correlation receiver elements.

TABLE III  
RESULTS OF TESTS OF NORMALITY FOR PERFECT SYNCHRONIZATION CASE

K	Bernoulli map	Tent map	Gauss map	Sine-Circle map	Cubic map	Pincher's map	N
6	1	1	1	1	1	1	15
18	1	1	1	1	1	0	
33	1	1	1	0	1	1	
65	0	0	0	0	0	0	
129	0	0	0	1	0	0	
259	0	1	0	1	0	0	
6	1	1	1	1	1	1	31
18	1	0	1	0	1	1	
33	0	1	1	0	0	0	
65	0	1	0	0	0	0	
129	0	0	0	0	0	0	
259	0	0	0	0	0	0	
6	1	1	1	1	1	1	63
18	1	0	1	0	0	0	
33	0	1	0	1	1	1	
65	0	1	0	0	0	0	
129	1	0	1	0	0	0	
259	1	0	0	0	0	0	
6	1	1	1	1	1	1	127
18	1	0	0	0	1	0	
33	0	0	1	1	0	1	
65	0	0	0	0	0	1	
129	0	0	1	0	0	1	
259	0	0	0	0	0	0	

TABLE IV  
RESULTS OF TESTS OF NORMALITY FOR ASYNCHRONOUS TRANSMISSION CASE

K	Bernoulli map	Tent map	Gauss map	Sine-Circle map	Cubic map	Pincher's map	N
6	1	1	1	1	1	1	15
18	0	0	1	0	0	1	
33	1	1	1	1	0	0	
65	0	0	0	0	0	0	
129	0	0	0	0	0	0	
259	0	0	1	0	0	0	
6	1	1	1	1	1	1	31
18	0	0	0	0	0	0	
33	1	0	0	0	0	0	
65	1	0	0	0	0	0	
129	0	0	0	0	0	0	
259	1	0	0	0	0	0	
6	1	1	1	1	1	1	63
18	0	0	0	0	0	0	
33	1	0	0	0	0	0	
65	0	0	0	0	0	0	
129	0	0	0	1	0	0	
259	0	0	0	1	0	0	
6	1	1	1	1	1	1	127
18	0	0	1	0	0	0	
33	0	0	0	0	0	0	
65	0	0	0	1	0	0	
129	0	1	0	0	0	0	
259	0	0	0	0	1	0	

## V. THE RESULTS OF SIMULATION

This section describes the results of the MAI distribution analysis realized via MATLAB simulation of a DS-CDMA communication system based on chaotic spreading sequences

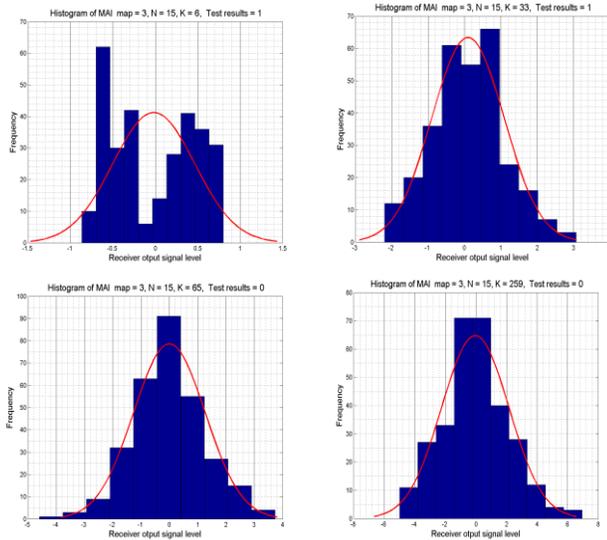


Fig. 2. Histograms of the MAI for a synchronous DS-CDMA system based on chaotic sequences with length 15 chips generated by Cubic map. The number of transmitters is respectively 6, 33, 65, 259 and the results of the test of normality for upper histograms are 1 (the number of transmitters 6, 33), and for bottom histograms are 0 (the number of transmitters 65, 259). A normal distribution curve for the corresponding mean value and variance is shown by the red line.

with  $K$  transmitters and a correlation receiver. The main purpose of the simulation is to test the hypothesis that the samples of the MAI calculated according to (9) come from a normally distributed population.

Two types of DS-CDMA system model have been examined during simulation: synchronous and asynchronous. In the case of the synchronous model, all time delays  $\tau_i$  are equal. In the case of the asynchronous system, the time delays observed by the receiver are different.

Considering that the expected value and variance of the distribution are not previously specified, the Lilliefors test has been used for MAI distribution analysis. The null hypothesis of the test is as follows: MAI samples come from a normally distributed population. The results of the test for the synchronous system model are presented in Table III, and the results for the asynchronous system model are presented in Table IV. For the asynchronous system simulation the delay impact  $\omega\tau_i$  is randomly selected from the set  $(0, 2\pi)$ .

If the result is 0, it means that the hypothesis of normality cannot be excluded. If the result is 1, it means that the hypothesis of normality can be rejected with the significance level 0.05.

Examples of histograms of the MAI and the expected normal distribution curves for 15-chips long chaotic spreading sequences generated by Cubic map and synchronous system case are presented in Fig. 2. The first upper histogram presents the MAI distribution for 6 users and the second upper histogram presents the MAI distribution for 33 users. As the Lilliefors test shows, the hypothesis of MAI normality for these cases can be rejected with the significance level 0.05. In turn, the lower histograms present the MAI distribution for 65 and 259 users. In these cases, according to the Lilliefors test,

the hypothesis of MAI normality cannot be rejected. The red curves show the normal distribution for simulated MAI mean and variance values.

## VI. CONCLUSION

The proposed research presents the multi-access interference (MAI) distribution analysis realized by MATLAB simulation of a synchronous and an asynchronous DS-CDMA communication system with  $K$  transmitters and a correlation receiver for relatively short chaotic spreading sequences. The MAI analysis has indicated overall similar results for the same sequences length, the number of transmitters and CDMA system type for the all observed 1-D maps.

Two main conclusions can be presented. For relatively short (6 - 127 chips long) binary chaotic spreading sequence, in more than half of the cases the MAI does not come from a normal distribution for the synchronous multi-user DS-CDMA communication system with the number of users less than 65. In the case of the asynchronous multi-user DS-CDMA communication system for all observed maps the MAI does not come from a normal distribution if the number of users is less than or equal to 6.

The approach for DS-CDMA multi-user system performance evaluation based on a Gaussian approximation of the MAI cannot be applied in the most of the observed cases.

## REFERENCES

- [1] C.H. Heidari-Batani, C.D. Mc.Gillem, "A Chaotic Direct-Sequence Spread-Spectrum Communication System", *IEEE Transactions on Communications*, vol.42, no. 2/3/4, pp. 1524-1527, 1994.
- [2] B.Jovic, C.P.Unsworth, G.S.Sandhu, S.M.Berber, "A Robust Sequence Synchronization Unit for Multi-user DS-CDMA Chaos-based Communication Systems", *Signal Processing* 87, Elsevier, pp. 1692-1708, 2007.
- [3] R.Rovatti, G.Setti, G.Mazzini, "Chaotic Complex Spreading Sequences for Asynchronous DS-CDMA – Part II: Some Theoretical Performance Bounds", *IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications*, vol. 45, no. 4, pp. 496-506, April, 1998.
- [4] V.Varadan, H.Leung, "Design of Piecewise Maps for Chaotic Spread-Spectrum Communications Using Genetic Programming", *IEEE Transactions on Circuits and Systems – I: Fundamental Theory and Applications*, vol.49, no.11, pp. 1543-1553, November, 2002.
- [5] E. Bekeris, A. Litvinenko, "Probability Distribution of Multiple-access Interference in Chaotic Spreading Codes Based on DS-CDMA Communication System", *Electronics and Electrical Engineering*, vol.123, no.7, pp.87-90, 2012.
- [6] E.Bekeris, A.Litvinenko, "Correlation Properties of Binary Spreading Sequences Generated by Chaotic Logistic Map", *Scientific Journal of RTU Series 7: Telecommunications and Electronics*, vol.9, RTU, pp. 42-47, 2009.
- [7] S. Berber, S. Feng, "Chaos-Based Physical Layer Design for WSN Applications", *17th WSEAS Int. Conf. on Communications*, Rhodes, Greece, pp. 157-162, July 16-19, 2013.
- [8] Kong De-peng, Zhu Jun-da, He Ting and Zhou Yi-fei. "The study of chaos encryption algorithm for wireless sensor networks based on the reconfigure technology of FPGA", *Journal of Chemical & Pharmaceutical Research*, vol. 6, issue 4, pp. 359-364, 2014.
- [9] G. Setti, R. Rovatti, G. Mazzini, "Synchronization Mechanism and Optimization of Spreading Sequences in Chaos-Based DS-CDMA Systems", *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E82-A (9), pp. 1737-1746, 1999.

# An improved implementation of hierarchy array multiplier using CslA adder and full swing GDI logic

Shoba Mohan and Nakkeeran Rangaswamy

**Abstract**—In this paper, an efficient implementation of a 16 bit array hierarchy multiplier using full swing Gate Diffusion Input (GDI) logic is discussed. Hierarchy multiplier is attractive because of its ability to carry the multiplication operation within one clock cycle. The existing hierarchical multipliers occupy more area and suffer from accumulation delay of base multiplier output bits. These issues can be addressed by incorporating carry select adder based addition and the multiplier implementation using full swing GDI logic. The basic computation blocks involved in the multiplier are AND gate and carry propagate adder. They are implemented with using full swing GDI logic. Due to their reduced transistor count and less power consumption, this multiplier implementation leads to significant improvement compared with the existing implementations. The designed and existing array multipliers are simulated at 45 nm technology model and their power consumption and delay are calculated from the simulation results. It is validated that the proposed hierarchy array multiplier based on full swing GDI logic has 27% less energy consumption than the existing design. The results confirmed that implemented multiplier has shown better performance and can be used for signal and image processing.

**Index Terms**— Full swing GDI logic, array multiplier, full adder, delay, digital circuit, hierarchy multiplier, carry select adder;

*Original Research Paper*  
DOI: 10.7251/ELS1721038M

## I. INTRODUCTION

While the growth of electronics market has driven the Very Large Scale Integration (VLSI) industry towards very high integration density and system on chip, critical concerns have been arising to the severe increase in power consumption and area. High power consumption raises temperature profile of the chip and affects overall performance of the system. Moreover, the explosive growth in laptops and portable personal communication systems demand long

battery life at the modest performance. This necessitates an intensive research in low power and low area IC design [1].

A Multiplier is a part of the processor that is widely used in digital devices such as computers, laptops, mobile phones and so forth. The various applications such as digital signal processing, image and video processing rely mainly in their multiplier performance. Also, multiplier is one of the major sources of power consumption in digital signal processor and microprocessor, etc. Therefore, the multipliers with high speed, lesser power consumption and low area are in great demand [2].

Hierarchical multipliers are considered as viable means for achieving orders of magnitude speed up in computer intensive applications through the use of fine grained parallelism. They are used in various fields of numerical and scientific computations, image processing, communication, cryptographic computation and so on [3-4].

In general, to design  $n$  bit hierarchical multiplier, four  $n/2$  base multipliers are necessary which generate  $2n$  bit output, where  $n$  represents hierarchical multiplier input width. It is noted that all the base multipliers are allowed to perform the task in parallel. Due to that, the performance of the hierarchy multiplier is determined from the accumulation delay of its base multipliers output bits. But this is a time consuming task as it requires more number of additions and considered a bottleneck for the hierarchy multiplier performance. In this work, an approach to perform this accumulation process is done by Carry Select Adder (CslA) to improve the performance. The following are the contributions discussed in the paper:

- (i) For the simple hierarchy multiplier implementation, array multiplication scheme is chosen for base multiplier realization
- (ii) To reduce the accumulation delay of base multiplier output bits, carry select adder is introduced
- (iii) To realize the hierarchy multiplier with small area, it is implemented using full swing Gate Diffusion Input (GDI) logic

The rest of the paper is organized as follows: An overview of the GDI logic is described in Section 2. In Section 3, the explanations of the parallel adders are given whereas in Section 4 the architecture of the proposed hierarchy multiplier

Manuscript received 12 June 2016. Received in revised form 27 May 2017. Accepted for publication 28 June 2017.

Shoba Mohan is with the Department of electronics Engineering, Pondicherry University, India (Phone: +91-7598308967; e-mail: [shobamalar@gmail.com](mailto:shobamalar@gmail.com)).

Nakkeeran Rangaswamy is with the Department of electronics Engineering, Pondicherry University, India.

is described. The simulation results and discussion are given in Section 5 and finally, the Section 6 concludes the paper.

## II. GDI LOGIC

The implementation of any digital circuits can be possible with Complementary Metal Oxide Semiconductor Logic (CMOS) [5]. It consists of both pull up (PMOS) and pull down (NMOS) transistor. Also, this logic needs an inverter to produce normal output from its complementary output. Though this logic has lesser power dissipation, it is suffer from more delay and area. To reduce the area i.e. (number of transistors required for the circuits implementation), Pass Transistor Logic (PTL) is invented [6]. But this logic has a drawback of threshold voltage problem which can be alleviated using a variant of pass transistor logic are complementary PTL called CPL, but the minimum requirement of generation of the complementary signal increases circuit overhead.

An alternative to CMOS logic, GDI logic is introduced, which is a low power design technique and offers the implementation of the various logic functions with fewer numbers of transistors [7]. The basic GDI cell is shown in Fig. 1. It looks like an inverter but it is not so. The source/drain both PMOS and NMOS transistor are tied at the diffusion input of P and N, respectively whereas in CMOS inverter, it is always tied at the  $V_{DD}$  and  $V_{SS}$ , respectively. The various logic functions realized using basic GDI cell is given in Table 1.

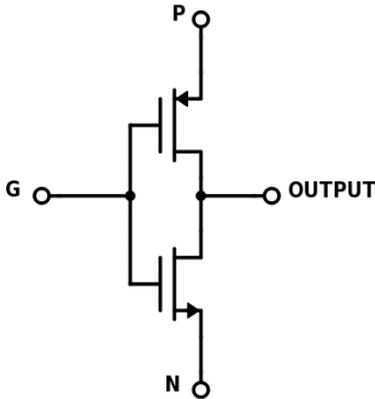


Fig. 1. Basic GDI cell [7]

Table 1. Various logic functions using GDI cell

TABLE 1  
VARIOUS LOGIC FUNCTIONS USING GDI CELL

N	P	G	Output	Function
0	B	A	$\bar{A}B$	F1
B	1	A	$\bar{A}+B$	F2
1	B	A	$A+B$	OR
B	0	A	$AB$	AND
C	B	A	$\bar{A}B+AC$	MUX
0	1	A	$\bar{A}$	NOT

From the operational characteristics of GDI gates, it is concluded that they produce reduced output voltage for certain

input combinations. This feature is beneficial for low power circuits. On the other hand, this may reduce noise margin and possible to increase the delay. Moreover, at low  $V_{DD}$  operation, the degraded output even may cause circuit malfunction. A simple technique for restoring output voltage level is the use of buffers. However, using buffers will cause propagation delay to increase proportionally to the number of cascaded stages [8]. Alternative to this, Ultra Low Power Diode (ULPD) is used, in which MOS transistor placed at the output terminal and can be realized as a diode [9]. An alternative to these techniques, full swing output voltage level can be retrieved by inserting proper transistor i.e. either NMOS/ PMOS depends on the voltage degradation. A set of full swing GDI gates and their operational characteristics are explained in [10]. As an example, the operation of XOR gate is explained.

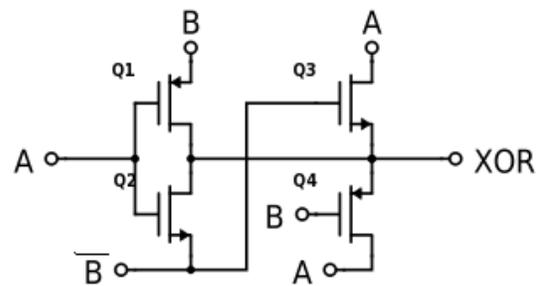


Fig. 2. Full swing GDI XOR gate [10]

The transistor level diagram of the XOR gate using full swing GDI logic is shown in Fig 2. The working mechanism of this gate is described here:

- Logic '0':

When  $AB = 00$ , Q1 and Q3 will be switched ON and other two transistors namely, Q4 and Q2 will be switched OFF. The output node is connected to  $GND$  potential through Q3 transistor. On the other hand, for the input combination of  $AB = 11$ , N1 transistor becomes switched ON and the remaining transistor are switched OFF. The output node is tied to  $GND$  potential.

- Logic '1':

When  $AB = 01$ , the transistors Q1 and Q4 will be switched ON whereas Q2 and Q3 will be switched OFF state. It is well known that PMOS transistor is good at delivering strong '1' potential ( $V_{DD}$ ). Likewise, for another input combination  $AB = 10$ , the transistor Q4 and Q2 will be switched ON and the delivering of  $V_{DD}$  potential is taken care by the PMOS transistor P2.

From this discussion, it is understood that these components exhibit better performance in terms of delay, power consumption and area. Therefore, they can be chosen while implementing the array multiplier to improve performance.

## III. PARALLEL ADDERS

Parallel adders are developed to minimize the delay involved in the binary addition task and are well suited for Very Large Scale Integration (VLSI) implementation. The

performance of these adders can be greatly influenced by the performance of their basic modules. The considered parallel adders are, ripple carry, carry select and carry look ahead adders. The basic modules of these parallel adders are Full Adder (FA) (for Ripple Carry Adder (RCA)), XOR and AND gate (for Carry Look Ahead (CLA)), FA and MUX (for Carry Select Adder (CsLA)). Therefore, these basic modules are realized using Full Swing Gate Diffusion Input (full swing GDI) logic, discussed by the authors in the earlier work [10]. For detailed understanding of their working mechanisms, readers are directed to refer [10]. The transistor level representation of FA is given in Fig. 3.

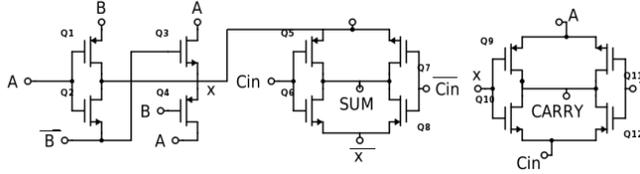


Fig. 3. Transistor level diagram of FA

#### A. Ripple Carry Adder

The RCA is  $O(n)$  time and  $O(n)$  area adders, where,  $n$  is the width of the operands. In the worst case, a carry can propagate from least significant bit position to the most significant bit position. Moreover, one stage of the RCA, the single full adder, determines the performance of RCA. Therefore, the delay of RCA can be decreased by implementing fast full adder. To achieve this, full adder based on full swing GDI logic is chosen as in [10]. Further, the carry propagation delay is reduced by minimizing carry propagation path or by performing pre-computation of carries.

#### B. Carry Look Ahead Adder

CLAs have become popular due to their high speed and modularity. They are  $O(\log n)$  time and  $O(n \log n)$  area adders. Consider the  $n$ -bit addition of two  $n$ -bit numbers  $A = a_n, a_{n-1}, \dots, a_0$  and  $B = b_n, b_{n-1}, \dots, b_0$  resulting in the output sum  $S = S_n, S_{n-1}, \dots, S_0$  and carry out  $C_{out}$ .

The first stage in CLA computes the bit generate ( $G_i$ ) and propagate ( $P_i$ ) as follows

$$G_i = A_i \cdot B_i \quad (1)$$

$$P_i = A_i \oplus B_i \quad (2)$$

These are then utilized to compute the final sum ( $S_i$ ) and carry ( $C_{i+1}$ ) bits,

$$S_i = P_i \oplus C_i \quad (3)$$

$$C_{i+1} = G_i \oplus P_i C_i \quad (4)$$

where  $0 \leq i \leq n$ .

The overall delay of carry look ahead adders is dominated by the delay of passing the carry in look ahead stages. The building blocks of CLA are XOR and AND gates. Moreover, the CLA performance is determined by the performance of them. Thus, the performance improvement in CLA is achieved by implementing those using full swing GDI logic in this paper.

#### C. Carry Select Adder

To minimize the delay due to carry propagation involved in RCA, CsLA is evolved, in which, two additions are performed in parallel, one assuming  $C_{in}$  as 0 and other one as 1. When the carry is known, finally the correct sum is selected. The pictorial representation of CsLA is shown in Fig. 4(a). They are  $O(2n)$  area and  $O(\sqrt{n})$  time adders. CsLA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a final carry to generate the sum. However, CsLA is not area efficient because it uses multiple pairs of RCA to generate intermediate sum and carry for  $C_{in}=0$  and  $C_{in}=1$ .

The different techniques for minimizing the use of dual RCA in CsLA is attempted in [13]-[15]. An interesting approach discussed in [13] is use of Binary to Excess 1 Converter (BEC) instead of RCA for  $C_{in}=1$  and its architecture is shown in Fig. 4(b). The inputs to BEC are as same as RCA, which is depicted in the Fig. 4b, as per ref. [13]. The BEC based CsLA involves less logic resources than the conventional CsLA. Further, the area reduction is possible in CsLA with the help of sharing common Boolean logic expression for  $C_{in}=0$  and 1 [14]. Though it requires less logic resources than the BEC based CsLA, the carry propagation delay is larger.

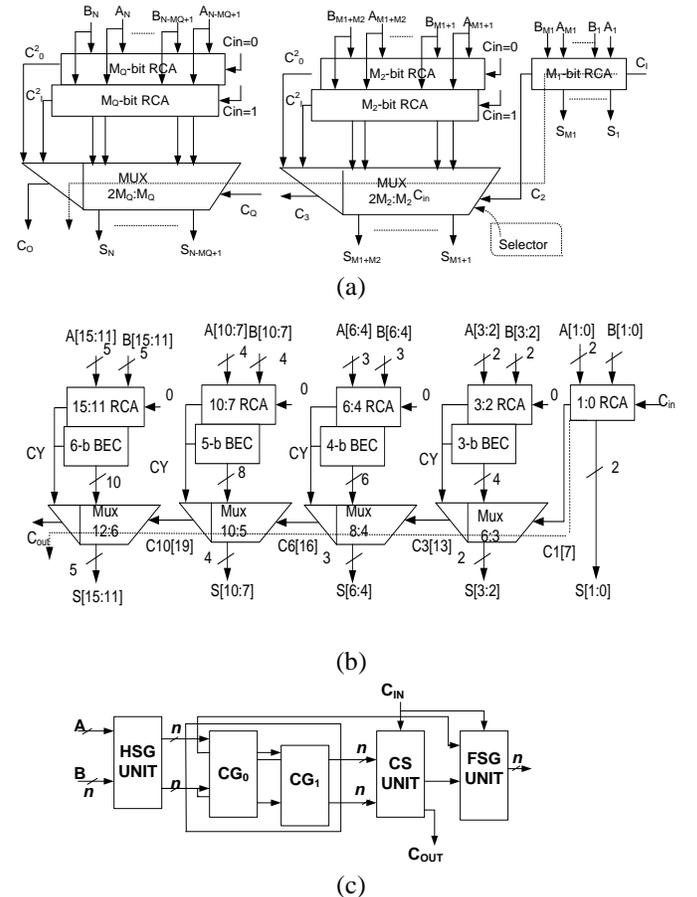


Fig. 4. CsLA Adders (a) N-bit Conventional CsLA in [11] (b) 16 bit BEC-CsLA in [13] and (c) N-bit CsLA in [15]

CsLA design is simplified based on logic reformulation and optimization of carry generator module [15]. This design possesses less area and delay than the conventional CsLA design. This adder schematic is given in Fig. 4(c). The performance of CsLA design can be improved by proper implementation of their basic modules such as MUX and FA. Therefore, these are designed using full swing GDI logic. In this paper, the conventional CsLA in [11], BEC CsLA discussed in [13] and CsLA given in [15] are implemented and their performance improvements are studied through simulation.

#### IV. HIERARCHY MULTIPLIER

Multipliers with large width are required for the implementation of cryptography and error correction circuits for more reliable transmission over highly insecure and/or noisy channels in networking and multimedia applications. The hierarchical principle helps to realize fast large bit multiplier, except that it requires a large width adder for performing the addition task, which poses limitation on the performance and increases area of the designed multiplier [16-18].

Over the last few decades, a lot of works have been dedicated, at the algorithmic and implementation level, to improve the performance of hierarchy multiplier. The conventional hierarchy multiplier architecture is shown in Fig. 5(a). The multiplier inputs are  $X$ ,  $Y$  of  $n$  bit width and produces the output  $P$  of  $2n$  bit. First, the inputs  $X$  and  $Y$  are divided into equal two halves namely,  $X_H$  and  $X_L$ ,  $Y_H$  and  $Y_L$  and they are multiplied by base multiplier. As seen in Fig. 5 (a), the base multipliers are used for the multiplication of  $(X_L$  and  $Y_L)$ ,  $(X_H$  and  $Y_L)$ ,  $(X_L$  and  $Y_H)$  and  $(X_H$  and  $Y_H)$ . Once these multiplication processes are over, their output bits will form a carry save array which is processed by Carry Save Adder (CSA) resulting with two rows of  $16$  bit output. These bits are further added with the help of Carry Propagate Adder (CPA) to produce the multiplier output bits.

The delay in the addition process of the hierarchy multiplier is reduced with the parallel execution of ripple carry adder [16]. However, this method requires twice the number of adders thus results in increased area. On the other hand, the delay is reduced with the deployment of carry look ahead adder for the addition process but this increases the interconnection complexity [17]. Not only delay and area, the power consumption of the hierarchy multiplier also has to be reduced because the existing designs appending more zeros to equalize the number of bits in order to make them suitable for parallel computation [18]. This might increase the spurious activities and thus increases the power consumption. The above mentioned issues in the existing hierarchy multiplier can be addressed by

- (i) Performing the final addition using proposed carry select adder,
- (ii) Implementing the proposed hierarchy multiplier using full swing GDI logic as carried out in this paper.

The architecture of the proposed hierarchy multiplier is given in Fig. 5(b). The carry propagate adder is replaced by

carry select adder to improve the computation time.

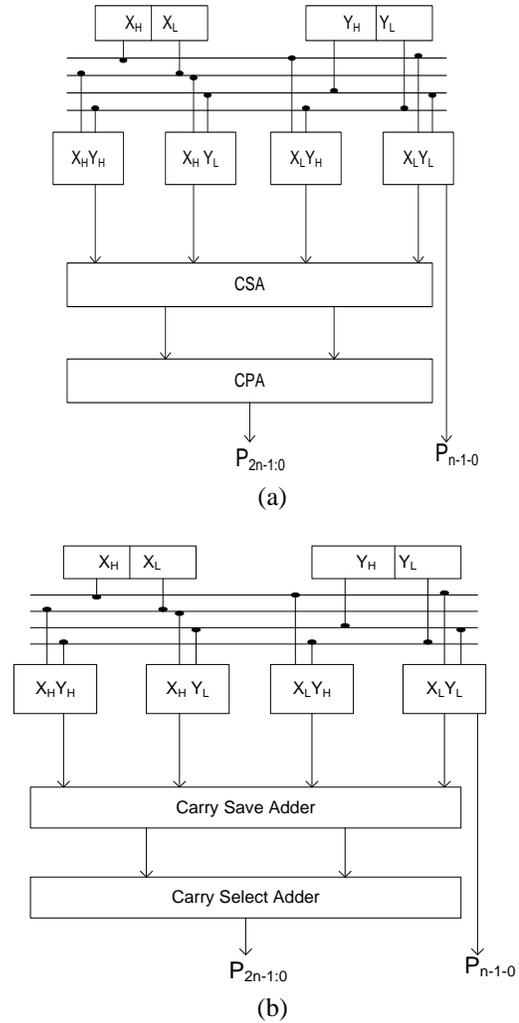


Fig. 5. Hierarchy Multiplier Architecture (a) Conventional and (b) Proposed

##### A. Base Multiplier

The array multiplier is chosen as base multiplier, which uses full adders for partial products reduction. Though it has a regular layout structure, the delay is increasing with increase in number of input bits. Since the delay of array multiplier is mainly determined from this adder delay, which might be decreased by incorporating hierarchy principle based multiplication. The architecture of a  $4 \times 4$  array multiplier for partial products reduction is given in Fig. 6. In the architecture,  $a$  and  $b$  are multiplier input operands each having 4 bits. The partial products are generated from  $AND$  gates, acted as inputs to these adders. Once the partial products accumulation is completed, the array multiplier output bits  $P_7, \dots, P_0$  become available. It is evident from the multiplier architecture, its power consumption and the area is determined by the number of transistor of all adders in the array. Since the basic components of the array multiplier are  $AND$  gate and full adder, this performance can be improved by utilizing the better circuits for them. Generally, the partial products of multiplication are generated by  $AND$  gates. For  $n$  bit

multiplier, it requires  $n^2$  *AND* gates. Because the *AND* gate based on CMOS and GDI uses 6 transistors, it needs  $6n^2$  transistors, to construct the *AND* gate array in the  $n$  bit array multiplier. But the full swing GDI based *AND* gate uses 5 transistors only. Thus, results in decreased number of transistors. Another component of multiplier i.e., full adder, required for  $n$  bit multiplier is  $(n-1)n$ , which is realized in CMOS, GDI and full swing GDI uses 28, 24 and 18 number of transistors, respectively. As a result of these modifications in the multiplier implementation, its performance is improved. The reduced number of devices not only offers the reduction in delay, the unnecessary switching activities are also minimized, thereby minimizing the power consumption as well.

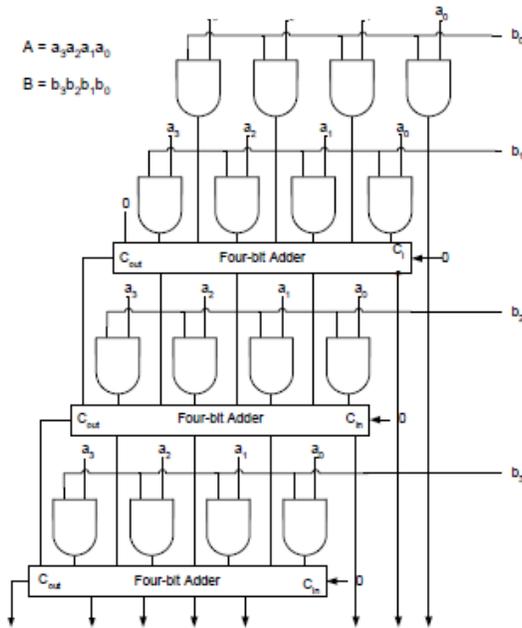


Fig. 6. 4x4 Array multiplier

## V. SIMULATION RESULTS AND DISCUSSION

### A. Parallel Adder

In this Section, the simulation results of the parallel adders based on CMOS, GDI and full swing GDI logic are presented and the performance of them is compared. During the evaluation of these adders, the performance metrics such as area, delay, power consumption and Power Delay Product (PDP) are taken into account. The simulations are performed at 45 nm freePDK technology with a supply voltage ( $V_{DD}$ ) of 1.1 V using Cadence Virtuoso tool. Typical transistor sizes, i.e.,  $(W/L)_p=120$  nm/45 nm and  $(W/L)_n=240$  nm/45 nm are used [19]. After the completion of simulation of parallel adders, the layout is generated for each of them and subjected to Design Rule Check (DRC) then Layout Versus Schematic (LVS) check before the extraction of parasitic. Subsequently, the extracted parasitic file is back annotated to perform the post layout simulation. In the simulation environment, each input is driven by buffered signal and each output is loaded

with buffer. Power and delay of the buffers are included in the power and delay calculations of the whole circuit.

**Delay:** The delay is measured by accounting the time from the 50% of the input voltage swing to 50% of the output voltage swing for each transition. The maximum delay is treated as worst case delay. The delay computed through simulation, for all the adder structures are plotted in Fig. 7(a). As it is expected, CLA structures have smaller delay compared to those other four adders due to the parallel computation of their carries. On the other hand, RCA has the highest delay due to its serial structure. However, RCA implemented based on full swing GDI adder, as reported in [10] has shown 12% and 6% speed improvement than CMOS and GDI adders, respectively.

The critical path delay of CslA is smaller than that of RCA due to the skipping of carry propagation. Further, the percentage of delay reduction in conventional CslA, BEC CslA in [13] and CslA in [15], which are implemented based on full swing GDI is 15, 27 and 20 than CMOS based implementation of those adders. The implementation of basic modules of CslA such as XOR, MUX and FA in full swing GDI logic can able to provide significant speed improvement than attained in other parallel adders like RCA and CLA.

**Power Consumption:** The power consumption of any circuit mainly depends on the switching activities of node and wire capacitances. The power consumed by the parallel adders are computed through simulation and also presented in Fig. 7(b). The results indicate that the CLA and CslA have more power consumption than that of RCA. The minimum power consumption is witnessed in RCA owing to its simple and regular structure while CLA consumes more power due to its dense wiring tracks. However, the power consumption of the CLA based on full swing GDI reduced by 30% comparing to CMOS based design. As shown in Fig. 7(b), the power consumption of adders based on full swing GDI gates is decreased almost same for all the adders and the results reveal, on average, 35% improvement is achieved over CMOS logic. Comparing with conventional GDI, full swing GDI minimizes the power consumption in parallel adders, on average, 30% by maintaining full swing voltage at intermediate nodes, which reduces the spurious transitions of the adder.

**Area:** The layout is drawn for all these implemented adders. The area is evaluated from their layout and it is plotted in Fig. 7(c). From the obtained results, it is witnessed that full swing GDI based RCA has less area whereas more area belongs to CMOS based CLA adder. Since the single FA realized by full swing GDI has less area than either CMOS or GDI logic, which might be a reason that overall area of RCA becomes lesser. Likewise, in the CslA designs, 26% area savings is achieved in full swing GDI based designs than those are implemented in CMOS logic. Similarly, the percentage of area reduction using full swing GDI based CLA adder is 17 and 13, respectively more than CMOS and GDI logic.

**PDP:** The power delay product of the parallel adders for CMOS, GDI and full swing GDI logic are plotted in Fig. 7(d). Among the adders discussed, the worst and the best PDP belongs to full swing GDI based CslA in [15] and conventional CslA based on CMOS, respectively.

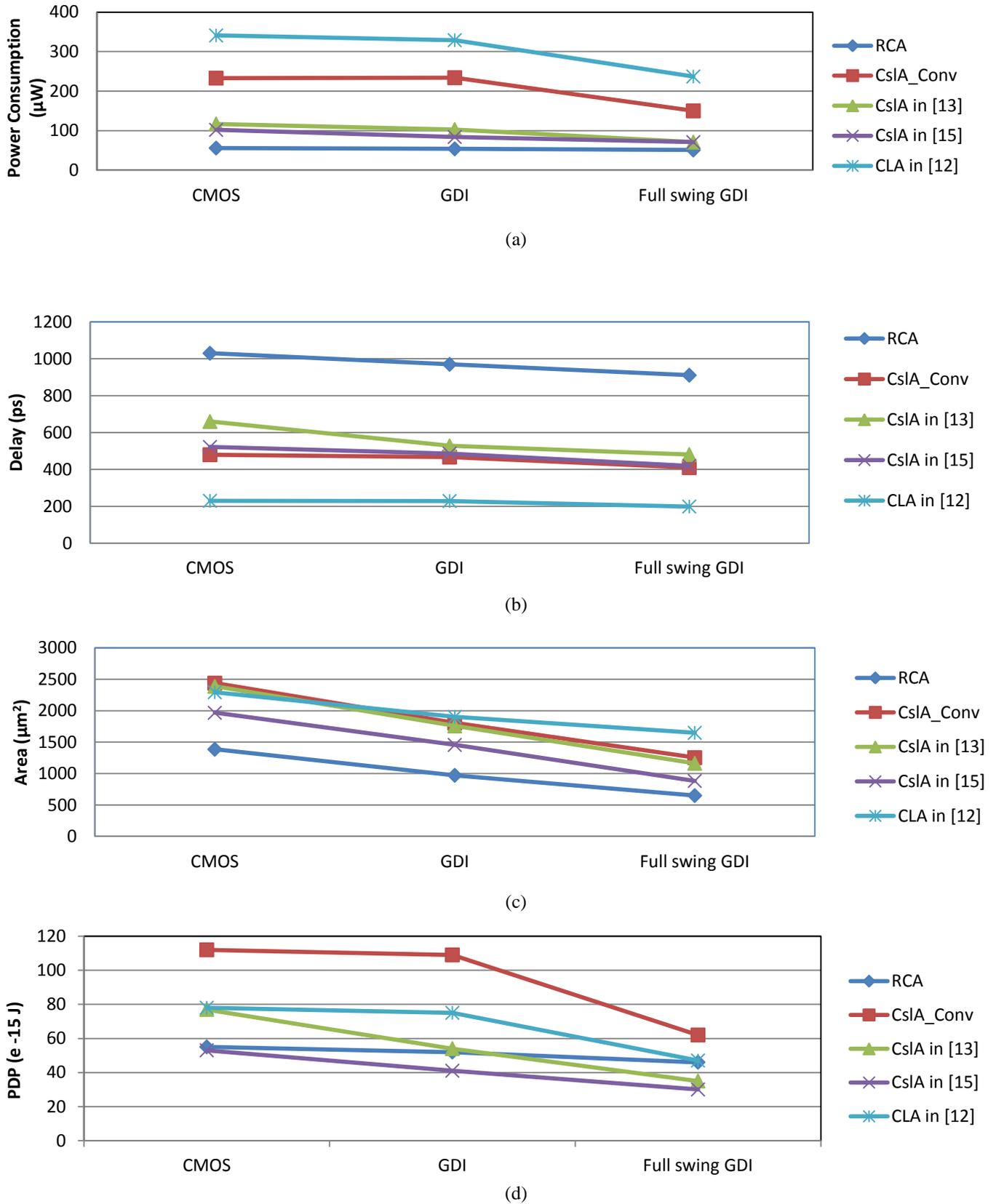


Fig. 7 Simulation results of parallel adders (a) Delay (b) Power Consumption (c) Area and (d) PDP

However, the PDP of conventional CslA is reduced in full swing GDI by 45% and 43% than CMOS and GDI, respectively. Similarly, in the CLA and RCA operated with lesser PDP in full swing GDI by 40% and 16%, respectively than CMOS. Also, it is examined from the obtained results of PDP of parallel adders, CslA implemented with full swing GDI logic has small PDP with acceptable speed and hence, they can be a proper choice while designing high performance and low power applications.

*Sensitive to Process Variation:* In order to evaluate the sensitivity of the designs to local and global process variations Monte Carlo simulations have been carried out for parallel adders. The simulations have carried out for 1000 runs. The

variations in power consumption, delay and PDP with respect to the process variations are depicted in Fig. 8. As expected, the full swing GDI based parallel adders have better immunity to process variation compared with others. The parallel adders, based on CMOS have variation in delay, power consumption and PDP as, on average, 3%, where as in full swing GDI based, the variations are approximately about 1.

*Adder Efficiency:* An approach of comparing different adders by defining Merit Factor (MF) based on the performance and the reliability parameters is discussed in [1]. The speed and energy consumption are the key parameters for determining the efficiency of the adders design, so the product of delay and power are considered as the merit factor. The expression merit

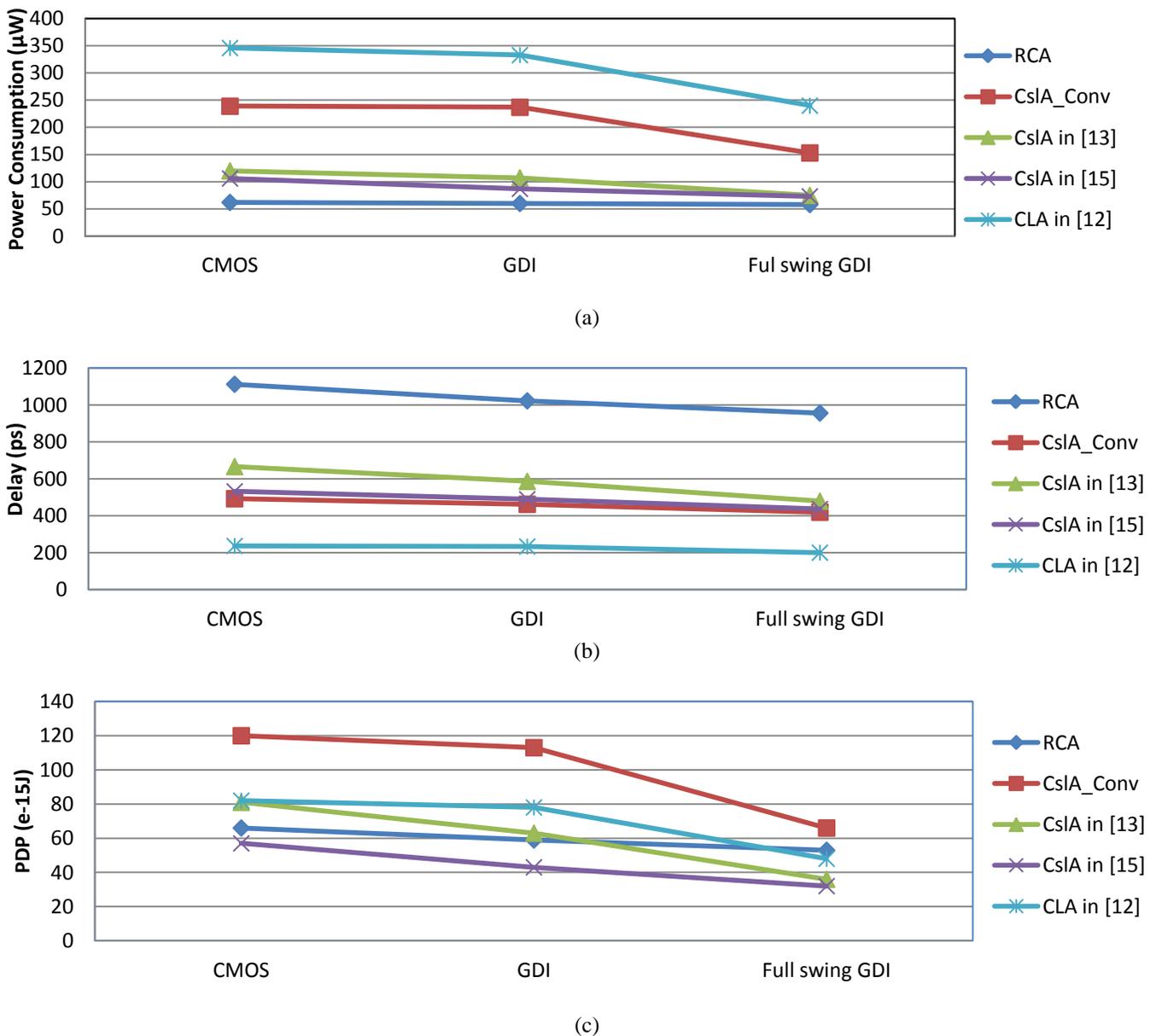


Fig. 8 Monte Carlo Simulation Results of Parallel Adders (a) Power Consumption (b) Delay and (c) PDP

factor is given in Eq. (5).

$$MF = \frac{1}{D_{norm} \cdot P_{norm} \cdot \Delta D_{norm,pv} \cdot \Delta P_{norm,pv}} \quad (5)$$

The delay and power consumption of the adders may be normalised by dividing them to the maximum value of power consumption and delay of particular adder among all the logic styles. The normalised delay and power consumption, represented as  $D_{norm}$  and  $P_{norm}$ , respectively. In addition to that, normalized delay and power consumption changes (variations) due to process variations, represented as  $(\Delta D_{norm, pv})$  and  $(\Delta P_{norm, pv})$ , respectively, are included in MF to evaluate the reliability of the designed adders. The amount of  $(\Delta D_{norm, pv})$  for each logic style is calculated by dividing the delay variations due to process variation with the maximum value of delay variation among all the logic styles for a particular adder. The same procedure is applied for  $(\Delta P_{norm, pv})$  computation. For merit function defined above, higher MF corresponds to better adder design.

The merit factor for all the adders versus different logic styles namely, CMOS, GDI and full swing GDI is shown in Fig. 9. It is observed that, the low and high value of MF corresponds to full swing GDI based CslA in [15] adder and CMOS based CLA adder. Moreover, full swing GDI logic based parallel adders namely, CslA in [15] and RCA has shown MF improvement of 80% and 36%, respectively than the realization in CMOS logic. From the obtained results, it is concluded that the adder efficiency can be improved significantly with the help of full swing GDI logic implementation.

The performance improvement of parallel adder structures such as RCA, CslA and CLA with the help of full swing GDI logic is attempted. The basic modules of these adder such as XOR, AND, MUX and full adder are realised using full swing GDI logic. It is observed that, the full swing GDI based RCA, CslA and CLA have shown speed improvement in terms of 12%, 27% and 14%, respectively than CMOS logic. Likewise,

the amount of area reduction achieved in RCA, CslA and CLA is 53%, 28% and 17%, respectively than CMOS logic. Among the parallel adders, the CslA adders based on full swing GDI logic have 45% PDP improvement than their existing implementation.

Finally, the adder efficiency is measured for all the simulated adders based on the merit factor, which depends on delay and power delay product. From the efficiency results, it is concluded that the full swing GDI based parallel adders namely; CslA adder discussed in [15] has shown merit factor improvement by 80% compared to that CMOS based implementation of the same adder. From the discussion of the performance improvement in parallel adders based on full swing GDI logic, CslA adders have 45% improvement in PDP than that of RCA and CLA adders. Therefore, they can be used in the place of adders while implementing multipliers for improving their performance.

### B. Hierarchy Multiplier

In this Section, the simulation results of the conventional and the proposed hierarchy array multiplier are presented.

**Delay:** The delay is measured from the 50% of the input voltage swing to 50% of the output voltage swing for each transition. The maximum delay is treated as worst case delay. The delay results of the simulated multipliers are given in Table 2. The hierarchy multiplier based on full swing GDI gates has 7% lesser delay than conventional design. This is attained due to the deployment of CslA in the hierarchy multiplier architecture thus reduces the base multiplier accumulation delay.

**Power Consumption:** The power consumed by the multipliers are computed through simulation and also presented in Table 2. From the results, it is observed that the multiplier based on full swing GDI logic consumes less power than the conventional solution discussed in [3]. This is achieved due to the elimination of spurious transitions involved in the multiplier. Also, the full swing output at intermediate nodes minimizes the unnecessary switching of the transistor which

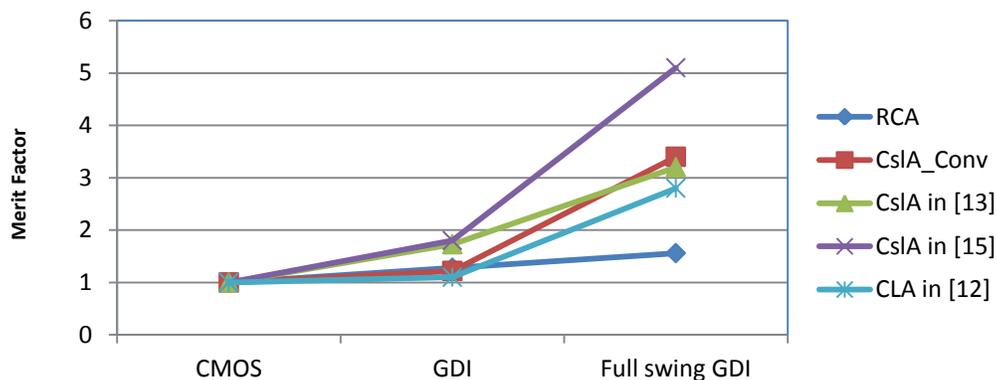


Fig. 9. Merit factor of parallel adder

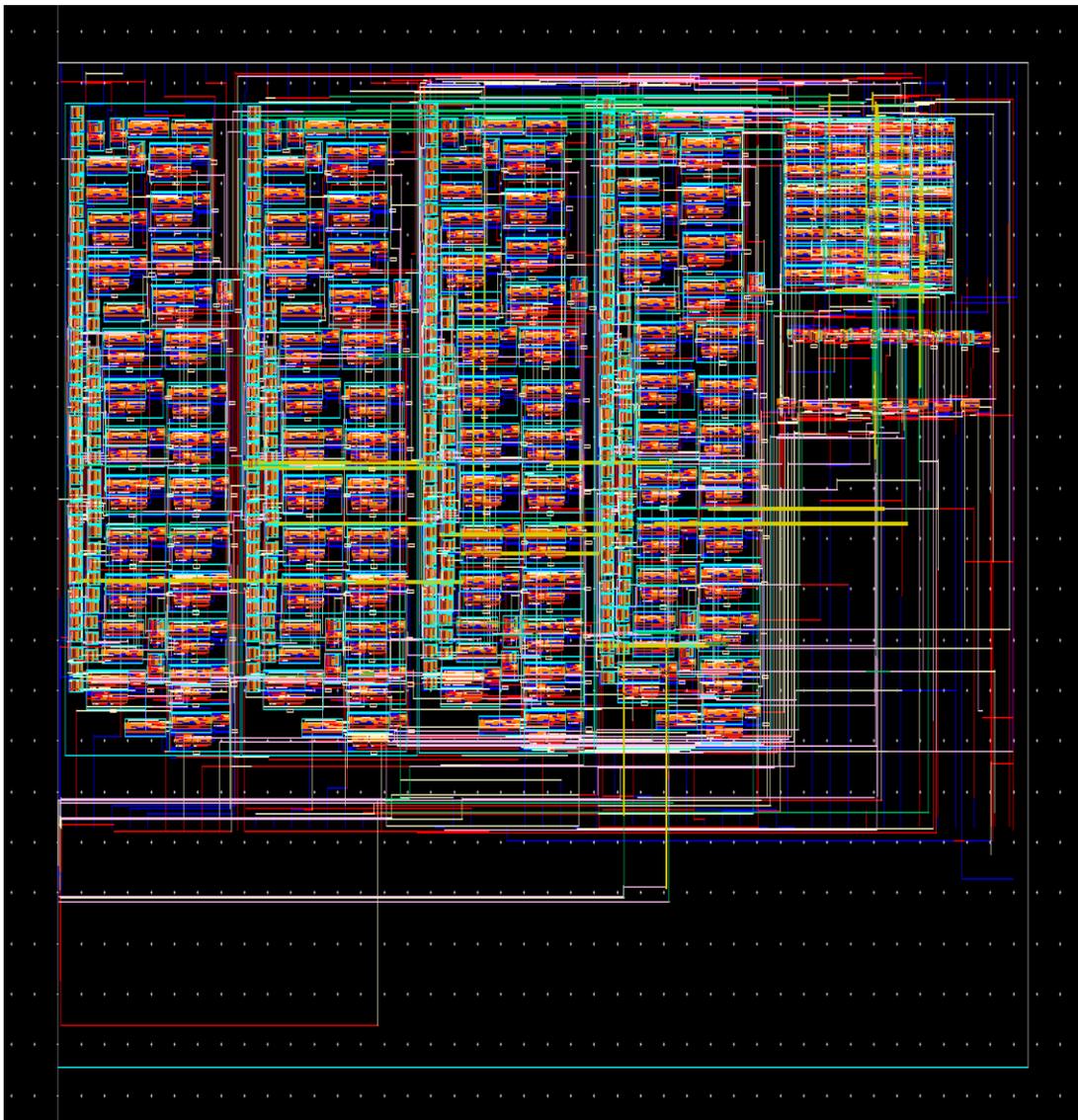


Fig. 10. Layout of the hierarchy multiplier architecture

would result in the power consumption minimization. The power saving is possible with full swing GDI based hierarchy multiplier design is 21% than existing multiplier.

TABLE 2  
SIMULATION RESULTS OF 16 BIT ARRAY MULTIPLIER

Multiplier	Delay (ps)	Power Consumption ( $\mu\text{W}$ )	PDP ( $e-15 \text{ J}$ )	Area ( $\mu\text{m}^2$ )
Conventional [3]	882	70	62	17857
Proposed	824	55	45	16681

*Area:* The layouts are drawn for all the simulated multiplier and the area is calculated from them and listed in Table 2. From the obtained results, it is observed that the proposed multiplier requires 7% less area compared with the existing hierarchy multiplier. This is attained due to the implementation using full swing GDI logic, which in turn

minimises the transistor count thus results in the small area. The layout of the proposed multiplier is given in Fig. 10.

*PDP:* From the simulated results, it is observed that, proposed multiplier has smaller PDP or energy consumption. It is observed from the results that conventional design operates with more energy consumption than the proposed one.

*Sensitive to Process Variation:* A study of circuit performance under the local and global process variations is carried through Monte Carlo simulations with thousand runs ( $N=1000$ ) and observed that the proposed multiplier is able to sustain the same performance with or without process variations.

## VI. CONCLUSION

The existing implementation of array multiplier based hierarchy multiplication lacks in terms of area and delay, which is due to the requirement of more transistor count for its base components such as AND gate and adder. To overcome these drawbacks, hierarchy multiplier is designed with CslA

which is further implemented using full swing GDI logic. An investigation of various parallel adder architectures performance is carried out and it is concluded that Cs1A possess better performance than conventional one. The introduction of Cs1A in hierarchy multiplier reduced its delay significantly. The performance of the multipliers is analyzed using SPICE simulation at 45 nm technology models. The performance parameters like delay and power consumption of the multipliers are measured from simulation results. From the simulation results, it is understood that the implemented multiplier design gives smaller power delay product and number of transistor comparing to the design found in the literature.

#### ACKNOWLEDGMENT

The authors would like to thank the VIT University, Vellore, India for providing support to carry out some of the simulation works at Integrated Circuit Design Laboratory.

#### REFERENCES

- [1] M. Bahadori, M. Kamal, A. Afsail-kusha and M. Pedram, A Comparative study on performance and reliability of 32 bit binary adders, *Integration, the VLSI Journal*, Vol. 53, No. 1, 2016, pp. 54-67.
- [2] M. Valinataj, Fault-tolerant carry look ahead adder architectures robust to multiple simultaneous errors, *Microelectronics Reliability*, Vol. 55, No. 12, 2015, pp. 2845-2857.
- [3] G. Quan, J. P. Davis, S. Devarkal and D.A. Buell, High level synthesis for large bit width multipliers on FPGAs: A case study. *In proceedings of the international conference on hardware/software codesign and system synthesis*, pp. 213-218, 2005.
- [4] Z. Zakaria and S. A. Abbasi, Optimized multiplier based upon 6 input LUTs and Vedic mathematics, *World Academic of Science, Engineering and Technology*, Vol. 7, 2013, pp. 26-30.
- [5] S. Goel, A. Kumar and M.A. Bayoumi, Design of robust,energy-efficient full adders for deep-submicrometer design using hybrid-CMOS logic style, *IEEE Trans. on VLSI Syst.*, Vol. 14 No.12, 2006, pp. 82-94.
- [6] S. Purohit and M. Margala, Investigating the impact of logic and circuit implementation for full adder performance, *IEEE Trans. on VLSI Syst.*, Vol. 20, No. 7, 2012, pp. 1327-1331.
- [7] A. Morgenshtein, A. Fish and I. A. Wagner, Gate-Diffusion Input (GDI) – A power-efficient method for digital combinatorial circuits, *IEEE Trans. on VLSI Syst.*, Vol. 10, No. 5, 2002, pp. 566-581.
- [8] A. Morgenshtein, I. Shwartz and A. Fish, Gate Diffusion Input (GDI) Logic in standard CMOS nanoscale process, *Proc. of IEEE Convention of Electrical and Electronics Engineers in Israel*, pp.776-780, 2010.
- [9] V. Foroutan, M. Teheri, K. Navi and A. Mazreah, Design of two low power full adder using GDI structure and hybrid CMOS logic style, *Integration, the VLSI Journal*, Vol. 47, No.1, 2014, pp. 48-61.
- [10] M. Shoba and R. Nakkeeran, GDI based full adders for energy efficient arithmetic applications, *Engineering Science and Technology, an International Journal*, Vol. 19, No.1, 2016, pp. 485-496.
- [11] N. H. E. Weste and D. Harris, *CMOS VLSI Design*, 2<sup>nd</sup> ed, Pearson Education, 2005.
- [12] A. Morgenshtein, I. Shwartz and A. Fish, Full swing Gate Diffusion Input (GDI) logic –case study for low power CLA adder design, *Integration, the VLSI Journal*, Vol. 47, No.1, 2014, pp. 62-70.
- [13] B. Ramkumar and H. M. Kittur, Low-power and area-efficient carry select adder, *IEEE Trans. VLSI Syst.*, Vol. 20, No. 2, 2012, pp. 371-375.
- [14] S. Y. Sun, Z. X. Zhang, and X. J. Jin, High-performance carry select adder using fast all-one finding logic, *Second Asia International Conference on Modelling Simulation*, pp. 1–5, 2008.
- [15] B. K. Mohanty and S. K. Patel, Area-delay-power efficient carry-select adder, *IEEE Trans. Circuits Syst. I Regul. Pap.*, Vol. 61, No.6, 2014, pp. 418-422.
- [16] J. Shi, G. Jing, Z. Di and S. Yang, The design and implementation of reconfigurable multiplier with high flexibility, *In proceedings of the international conference on electronics, communications and control*, pp. 1095-1098, 2011.
- [17] S. A. Abbasi, A. R. M. Zulhelmi and Alamoud, FPGA design, simulation and prototyping of 32 bit pipeline multiplier based on Vedic mathematics, *IEICE Electronics Express*, Vol. 12, 2015, pp. 1-12.
- [18] S. Quan, Q. Qiang, C. L. Wey, (2005). A novel reconfigurable architecture of low power unsigned multiplier for digital signal processing, *In proceedings of the international symposium on circuits and systems*, pp.3327-3330, 2005.
- [19] <https://www.eda.ncsu.edu/wiki/FreePDK45>



# Instructions for Authors

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.

## 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 delivered to the editor by e-mail: [els-journal@etf.unibl.org](mailto:els-journal@etf.unibl.org). 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 quality on your printer or monitor, please keep in mind that cannot improve the quality after submission.

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@etfbl.net](mailto:author@etfbl.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](mailto: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](mailto:author@elfak.ni.ac.rs)).

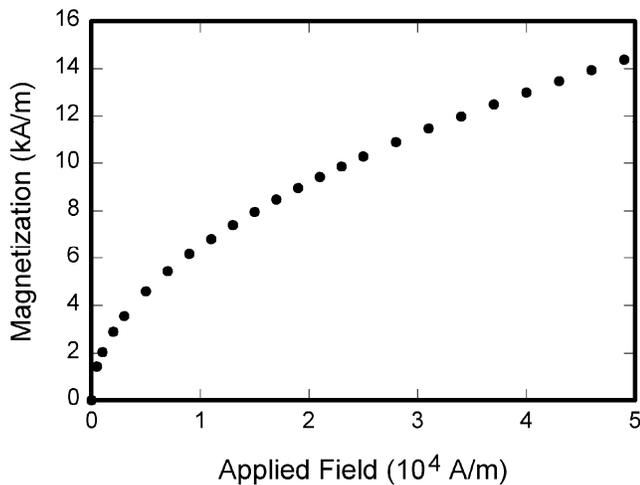


Fig. 1. Magnetization as a function of applied field. Note that “Fig.” is abbreviated. There is a period after the figure number, followed by two spaces. It is good practice to explain the significance of the figure in the caption.

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<sup>2</sup> (100 Gb/in<sup>2</sup>).” An exception is when English units are used as identifiers in trade, such as “3½-in disk drive.” Avoid combining SI and

TABLE I  
UNITS FOR MAGNETIC PROPERTIES

Symbol	Quantity	Conversion from Gaussian and CGS EMU to SI <sup>a</sup>
$\Phi$	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 <sup>2</sup>
$H$	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 <sup>2</sup> = $10^{-3}$ J/T
$M$	magnetization	1 erg/(G·cm <sup>3</sup> ) = 1 emu/cm <sup>3</sup> $\rightarrow$ $10^3$ A/m
$4\pi M$	magnetization	1 G $\rightarrow$ $10^3/(4\pi)$ A/m
$\sigma$	specific magnetization	1 erg/(G·g) = 1 emu/g $\rightarrow$ 1 A·m <sup>2</sup> /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 <sup>3</sup> ) = 1 emu/cm <sup>3</sup> $\rightarrow$ $4\pi \times 10^{-4}$ T
$\chi, \kappa$	susceptibility	1 $\rightarrow$ $4\pi$
$\chi_\rho$	mass susceptibility	1 cm <sup>3</sup> /g $\rightarrow$ $4\pi \times 10^{-3}$ m <sup>3</sup> /kg
$\mu$	permeability	1 $\rightarrow$ $4\pi \times 10^{-7}$ H/m = $4\pi \times 10^{-7}$ Wb/(A·m)
$\mu_r$	relative permeability	$\mu \rightarrow \mu_r$
$w, W$	energy density	1 erg/cm <sup>3</sup> $\rightarrow$ $10^{-1}$ J/m <sup>3</sup>
$N, D$	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.

<sup>a</sup>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.

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.

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<sup>2</sup>.”

## 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)  $\times$  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).<sup>1</sup> 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

<sup>1</sup>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.

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

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

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  $\times$  0.2 cm,” not “0.1  $\times$  0.2  $cm^2$ .” The abbreviation for “seconds” is “s,” not “sec.” Do not mix complete spellings and abbreviations of units: use “Wb/m<sup>2</sup>” or “webers per square meter,” not “webers/m<sup>2</sup>.” 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.

## VI. SOME COMMON MISTAKES

The word “data” is plural, not singular. The subscript for the permeability of vacuum  $\mu_0$  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  $\text{Ni}_{0.5}\text{Mn}_{0.5}$  whereas “Ni–Mn” indicates an alloy of some composition  $\text{Ni}_x\text{Mn}_{1-x}$ .

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

Appendices, 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. 9<sup>th</sup> Annu. Conf. Magnetism Japan, 1982, p. 301].
- [9] M. Young, *The Technical 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 in 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.\(URL\)](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.\(URL\)](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>



# Instruction 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 delivered to the editor by e-mail: [els-journal@etf.unibl.org](mailto:els-journal@etf.unibl.org).

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.electronics.etfbl.net](http://www.electronics.etfbl.net)).

**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.

# ELECTRONICS, VOL. 21, NO. 1, JUNE 2017

---

## PAPERS

LUT BASED GENERALIZED PARALLEL COUNTERS FOR STATE-OF-ART FPGAS .....	3
Burhan Khurshid	
EFFICIENT COMPUTERIZED-TOMOGRAPHY RECONSTRUCTION USING LOW-COST FPGA-DSP CHIP .....	12
Bassam A. Abo-Elftooh, Mohamed H. El-Mahlawy and Mahmoud E. A. Gadallah	
CFOA-BASED FRACTIONAL ORDER $PI^2D^0$ CONTROLLER.....	25
Tada Comedang and Pattana Intani	
STATISTICAL ANALYSIS OF MULTIPLE ACCESS INTERFERENCE IN CHAOTIC SPREADING SEQUENCE BASED DS-CDMA SYSTEMS (MAY 2016) .....	34
A. Litvinenko, E. Bekeris	
AN IMPROVED IMPLEMENTATION OF HIERARCHY ARRAY MULTIPLIER USING CSLA ADDER AND FULL SWING GDI LOGIC.....	38
Shoba Mohan and Nakkeeran Rangaswamy	

---

ISSN 1450-5843

UDK 621.38